



Super-Resolution Reconstruction of Porous Media Using Concurrent Generative Adversarial Networks and Residual Blocks

Ting Zhang¹ · Qingyang Liu¹ · Yi Du²

Received: 6 December 2021 / Accepted: 25 November 2022 / Published online: 25 December 2022
© The Author(s), under exclusive licence to Springer Nature B.V. 2022

Abstract

Accurate porous media reconstruction has always been one of the significant research hot-spots in the numerical simulation of reservoirs. The traditional methods such as multi-point statistics perform porous media reconstruction based on the statistical features of training images, but the process is possibly cumbersome and the result is less effective. Porous media reconstruction has been greatly developed and benefited by applying current flourishing deep learning to its simulation process thanks to the strong capability of extracting features by deep learning. As a typical branch of deep learning methods, generative adversarial network (GAN) can simulate a two-person zero-sum game through confrontation between a generator and a discriminator. However, in real experiments, constrained by the resolution of physical equipment and the size of samples, it is difficult to physically obtain a large-scale image of porous media with high-resolution (HR) since HR and large field of view are usually contradictory for physical equipment. In this paper, a method is proposed based on multistage concurrent GAN to learn the structural features of porous media from one low-resolution 3D image and then stochastically reconstruct larger-sized porous media images. Experimental comparison with some typical methods proves that this method can reconstruct HR images with favorable quality.

Article Highlights

- Our method is quite fast in multiple reconstructions by reusing model parameters.
- Our method realizes the super-resolution reconstruction of porous media from one low-resolution image.
- Our method outperforms SNESIM and some GAN's variants in the accuracy of reconstruction.

✉ Yi Du
duyi0701@126.com

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

² School of Computer and Information, Shanghai Polytechnic University, Shanghai 201209, China

Keywords Deep learning · Generative adversarial network · Super-resolution · Porous media · Reconstruction

Abbreviations

CGAN	Conditional generative adversarial network
CNN	Convolutional neural network
ConSinGAN	Concurrent single-image GAN
CT	Computed tomography
D	Discriminator
DDs	Difference degrees
E	Mathematical expectation
EDSR	Enhanced deep residual network
FIB	Focus ion beam
FOV	Field of view
G	Generator
GP	Gradient penalty
GAN	Generative adversarial network
HR	High-resolution
HR_TI_SA	High-resolution training image of sandstone
HR_TI_SH	High-resolution training image of shale
I_n	The input real image at stage n
\tilde{I}_n	The fake image generated at stage n
L_{adv}	The adversarial loss
LR	Low-resolution
LR_TI_SA	Low-resolution training image of sandstone
LR_TI_SH	Low-resolution training image of shale
Max_Con	A maximum concurrent stage number
MPC	Multi-point connectivity
MPS	Multi-point statistics
MSPGAN	Multi-scale pattern generative adversarial network
P_r	The real data distribution
P_g	The generator distribution
ResNets	Residual networks
REV	Representative elementary volume
SDs	Similarity degrees
SinGAN	Single-image generative adversarial network
SNESIM	Single normal equation simulation
SR	Super-resolution
SRCNN	Super-resolution CNN
SRCSGAN	Super-resolution concurrent single-image GAN
SRGAN	Super-resolution generative adversarial network
TIs	Training images
VDSR	Very deep convolutional networks
XOR	Exclusive OR
r	The scaling factor of images at each stage
x	Real image
z	Noise

1 Introduction

Porous media are widely found in soil, rocks, plants, animals and other substances. Seepage is a very common natural phenomenon existing in porous media. Seepage mechanics studies the motion pattern and law of fluids flow in porous media (Okabe and Blunt 2005; Singh and Mohanty 2000), which is not only related to the properties of the fluids themselves, but also influenced by the internal structures of porous media since the topological structure and geometric characteristics of pore space directly affect the flow of fluids in porous media. The study about the structures of porous media is quite meaningful for the research of fluids flow, but the extremely complex structures of pore space have caused a great challenge for the accurate acquisition or reconstruction of such structures. To address the above issue, physical experimental methods and numerical reconstruction methods are developed to obtain or reconstruct the structures of porous media (Zhang et al. 2016).

The principle of physical experimental methods is to scan the sample of porous media with a high-precision scanning instrument to obtain 2D or 3D data. Typical physical experimental methods include sequence slice imaging (Lymberopoulos and Payatakes 1992), focus ion beam (FIB) (Fredrich and Lindquist 1997), computed tomography (CT) (Hou et al. 2007), etc. The sequence slice imaging method can reasonably divide different segments of pore space into pores and throats and obtain the distribution of pore diameters, throat sizes and lengths through the study of pore space. However, the applicability of this method is limited because it is easy to destroy the pore structure (Vogel and Roth 2001). Fredrich and Lindquist (1997) used FIB to reconstruct 3D digital cores and preliminarily studied the relationship between porosity and permeability of sandstone with different volumes. Digital core reconstruction based on micro- and nano-CT can obtain high-resolution (HR) accurate images, but the experimental equipment is expensive and the preparation of experimental samples is complicated (Hou et al. 2007).

Physical experimental methods can accurately describe the real pore structures of porous media, but they cannot achieve a proper balance between HR and the large field of view (FOV) due to the limited size of experimental samples and equipment. Therefore, as complementary tools for physical experimental methods, numerical reconstruction methods have been extensively studied, which analyze and obtain large-scale digital pore data in a more cost-efficient way. Hazlett (1997) used the simulated annealing method to make the reconstructed model closer to the real results by obtaining more statistical information. Helene and Didier (2015) proposed the truncated Gaussian simulation method to improve reconstruction quality. Okabe and Blunt (2004, 2005) used multi-point statistics (MPS) for digital core reconstruction. Zhang et al. (2016) used isometric mapping to achieve non-linear dimensionality reduction of training images (TIs) to decrease the redundant data of porous media, accelerating the simulation process.

However, the above numerical reconstruction methods, such as MPS, tend to rely on the statistical information extracted from each simulation process, which is only stored in memory instead of files in hard disk. Hence, when the statistics in memory are cleared (e.g., when the computer is turned off), these data will be lost and cannot be reused. At present, the rapid development of deep learning is supported by the improvements in GPU computing and optimization of neural networks (He et al. 2016), which has been widely used in the field of prediction with the potential for the reconstruction of porous media. These deep learning techniques have been proved successful in enhancing and accelerating modeling beyond the limitations of previous physical modeling, providing a new idea and broad prospect for porous media reconstruction. Besides, deep learning

methods can reuse the parameters and models for subsequent simulations or reconstructions, showing an evident advantage over traditional numerical reconstruction methods.

Generative adversarial network (GAN) (Goodfellow et al. 2014) is regarded as an important generative model or an unsupervised generation method in deep learning, whose basic idea is derived from the two-person zero-sum game theory. The purpose of GAN is to estimate the distribution of real data samples and generate new data samples similar to the real ones. GAN was first applied to the reconstruction of porous media by Mosser et al. (2017). Based on the powerful feature extraction capability for TIs and the parallel GPU framework, GAN is able to speed up the reconstruction process and obtain favorable reconstructions. With the development of GAN and its variants, multiple GAN-based methods have been proposed for the reconstruction of porous media. Feng et al. (2019) reconstructed porous media from small sub-areas to a complete image using CGAN (conditional generative adversarial network). Valsecchi et al. (2020) presented a reconstruction method using GAN to reconstruct 3D porous media from 2D images.

One of GAN's disadvantages is its requirement for a lot of training data, but in practice sometimes it is difficult to meet this condition. To address this issue, a GAN's variant single-image GAN (SinGAN) (Shaham et al. 2019) used only a single TI and a multistage model to simultaneously learn the detailed information and general information of the TI in different training stages. Then, a concurrent SinGAN (ConSinGAN) (Tobias et al. 2020) was proposed to accelerate the training process by including multiple neighboring stages to perform concurrent training.

Super-resolution (SR) reconstruction can reproduce the statistics of high-frequency components from low-resolution (LR) images and obtain HR images from LR images by learning the implicit redundancy in LR images. SR algorithms based on deep learning establish the nonlinear end-to-end mapping relationship between input and output through multi-layer convolutional neural networks (CNNs). Dong et al. (2014) first proposed a super-resolution CNN (SRCNN) to construct a simple shallow CNN, and the image reconstruction quality was significantly improved compared to other SR reconstruction algorithms. Kim et al. (2016) proposed very deep convolutional networks (VDSR) introducing residual networks (ResNets) into SR. Ledig et al. (2017) proposed a super-resolution generative adversarial network (SRGAN) using perceptual loss and adversarial loss to enhance the reality sense of images in SR. By improving SRGAN through an enhanced residual network with simpler structures, Lim et al. (2017) proposed the enhanced deep residual network (EDSR). However, the reconstruction results of these methods are relatively fixed since they cannot reconstruct a variety of stochastically reconstructed results.

SR reconstruction essentially is a type of numerical analysis problems called the ill-posed problem, in which the output is very sensitive to the input. For an ill-posed problem, if the input has a small error, then the relative error of the output possibly will be very large. For the SR reconstruction problem, the input LR images are often affected by noise and other interference. In this case, multiple stochastically reconstructed SR images with the features similar to the TI can be viewed as equal-probability stochastic models, whose differences reflect the uncertainty in practical application, providing the trend or tendency of generated information with multiple possible results and predicting the models based on incomplete data.

To obtain 3D SR stochastic reconstruction of porous media, this paper proposes a SR concurrent single-image GAN (SRCSGAN), whose stochastic reconstruction can be larger than the original TI. At this point, the reconstructed scale will not be restricted by the size of the TI. In addition, only a single TI is needed in SRCSGAN and the network is

combined with ConSinGAN and residual blocks. Compared to some other typical reconstruction methods, SRCSGAN shows certain advantages in terms of reconstruction quality.

2 Related Work

2.1 GAN

The basic structure of GAN consists of two networks: a generator G and a discriminator D . The noise z obeying a certain distribution (such as the uniform distribution, Gaussian distribution, etc.) is input into G , which tries to generate a fake image $G(z)$ similar to the TI. D is a binary classifier to estimate the probability that an image input to D is from a real image x or a fake image $G(z)$. In the end, the fake image cannot be identified from the input data, meaning the probability of the discriminator to correctly identify the fake image is about 50%. The specific structure of GAN is shown in Fig. 1.

The optimization process of GAN can be regarded as a minimum and maximum game problem, whose ultimate goal is to make the discriminator D unable to distinguish the true or false image. The optimization function is as follows:

$$\min_G \max_D V(G, D) = E_{x \sim p_{\text{data}(x)}} [\log D(x)] + E_{z \sim p_z(z)} [\log(1 - D(G(z)))], \tag{1}$$

where $p_{\text{data}(x)}$ is the distribution of the real image x ; $p_z(z)$ is the probability distribution of the noise z ; $D(x)$ represents the probability that D determines that x comes from the real image; $D(G(z))$ represents the probability that D judges $G(z)$ comes from the noise z ; E is the mathematical expectation.

2.2 Residual Blocks

Generally, a deep network can improve its performance by learning more features of TIs through its multiple layers, thus generally better than a shallower one, but it is difficult to train such a deep network. With the deepening of the network layer, the accuracy of the model first continuously improves and reaches the maximum (saturation of accuracy). Then, with the continuous increase of the network depth, the accuracy of the model decreases significantly, which is called the “degradation problem.”

Residual networks (He et al. 2016) were proposed to solve the degradation problem of deep networks. Its main idea is using identity mapping to directly connect the different

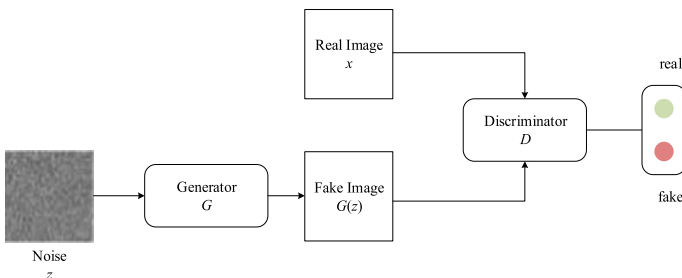


Fig. 1 The structure of GAN

layers of the network through “shortcut” connections, by which the residual networks are considered as the integration of multiple shallow networks. Then, the training can be carried out more easily compared to traditional deep networks.

A residual network is composed of a certain amount of residual blocks. The way to make the output of a residual block equal to its input is called identity mapping. Assume a traditional learning block and a residual block have, respectively, achieved identity mapping $H(x)=x$, as shown in Fig. 2, in which x is the input; $F(x)$ is the residual term; $H(x)$ is the expected output; relu is an activation function; “ \oplus ” is an adding operation. The main difference of such a traditional learning block and a residual one lies in the “shortcut” connection. In Fig. 2a, there is a relation $H(x)=x$, while in Fig. 2b this relation has been changed by “shortcut” and can be expressed as:

$$H(x) = F(x) + x. \tag{2}$$

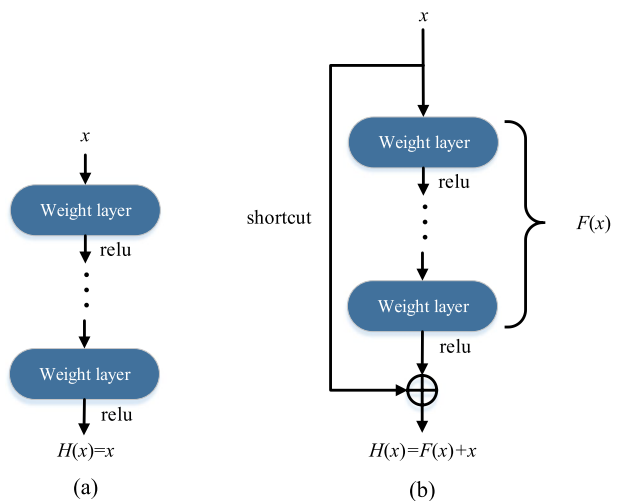
As shown in Fig. 2a, the input x passes through each weight layer and the activation function relu until the expected output $H(x)=x$ is obtained. At the end of the residual block in Fig. 2b, the input x is directly transferred to the output by a “shortcut” connection, and then the result $H(x)=F(x)+x$ is obtained. Then, the identity mapping $H(x)=x$ can be transformed into learning a residual term $F(x)=H(x)-x$, so the identity mapping $H(x)=x$ can be achieved in Fig. 2b as long as $F(x)=0$.

The “shortcut” connections will not add additional parameters or computational complexity in the training process, which can be trained by stochastic gradient descent (SGD) method in backpropagation and can be easily realized by using known common software libraries [e.g., Caffe (Jia et al. 2014)]. Therefore, compared to the traditional learning block, the residual block simplifies the learning process and learns the identity mapping easier and faster, greatly alleviating the risk of the degradation problem.

2.3 ConSinGAN

For GAN, it is almost impossible to directly capture all the features (e.g., global structures and local details) of TIs through only a single image due to its limited receptive field. Therefore,

Fig. 2 A traditional learning block and a residual block both achieving identity mapping. **a** A traditional learning block, **b** a residual block



the training of GAN is usually based on a large number of training samples. However, if the available training data are insufficient, the model trained on the basis of a small number of images or even a single image is often not quite convincing. SinGAN (Shaham et al. 2019) solves this problem by using the convolutional pyramid structure to capture the statistics of a single TI from global structures to local details. The original single TI is downsampled to different stages to form a multistage training sequence. The generator and discriminator of SinGAN are trained in multiple stages at different resolutions. The image generated by the previous stage is input to the current generator, but the generator at each stage is trained separately, which means that the trained weights of the previous generator cannot be reused by the current one.

In SinGAN, each stage is relatively isolated and therefore the corresponding training process at each stage is not related, which has greatly limited the inherent connection of different stages. Instead, ConSinGAN (Tobias et al. 2020) designs a concurrent training process for the generator, in which multiple stages in the generator can be trained together simultaneously. For example, the training process of ConSinGAN with a total of six stages is shown in Fig. 3. The whole training process starts from stage 0 and ends at stage 5. When the network at stage 0 completes its iteration, stage 1 joins the generator G_0 for concurrent training, but the discriminator D remains unchanged. The generator at stage 1 now includes G_0 and G_1 , in which G_0 "inherits" from stage 0 and initially reuses the trained parameters of G_0 . For the new added generator G_1 , its parameters need to be initialized. The discriminator D at stage 1 can reuse its parameters at stage 0 instead of reinitialization. The following stages repeat the above operation until stage 5 is reached. These reused parameters are relatively closer to the final determined values than completely random initialized parameters at each stage. Hence, the adjustment and training of parameters are relatively less, thus accelerating the convergence process.

In addition, to avoid overfitting, only part of generators are trained simultaneously. A maximum concurrent stage number (for convenience called Max_Con) is used to control the number of concurrent training stages. In Fig. 3, at most three generators are trained together in one stage, which means Max_Con is set to 3.

2.4 MSPGAN

Another SR method for porous media reconstruction, called multi-scale pattern generative adversarial network (MSPGAN) (Zhang et al. 2022), will be compared with SRCSGAN in the following experiments, so it is briefly introduced in this section. MSPGAN only needs a single 3D TI to reconstruct high-quality images by capturing and visualizing the distribution information of patches from the TI. The scaling transformation added to the generator enables MSPGAN to reconstruct images of various scales, especially to perform the SR reconstruction. MSPGAN's multi-scale discriminator can effectively optimize the reconstructed images and guarantee reconstruction quality. In the lower scales, the discriminator mainly discriminates the reconstruction of local detailed information, while in the higher scales, the discriminator pays more attention to the global structural information.

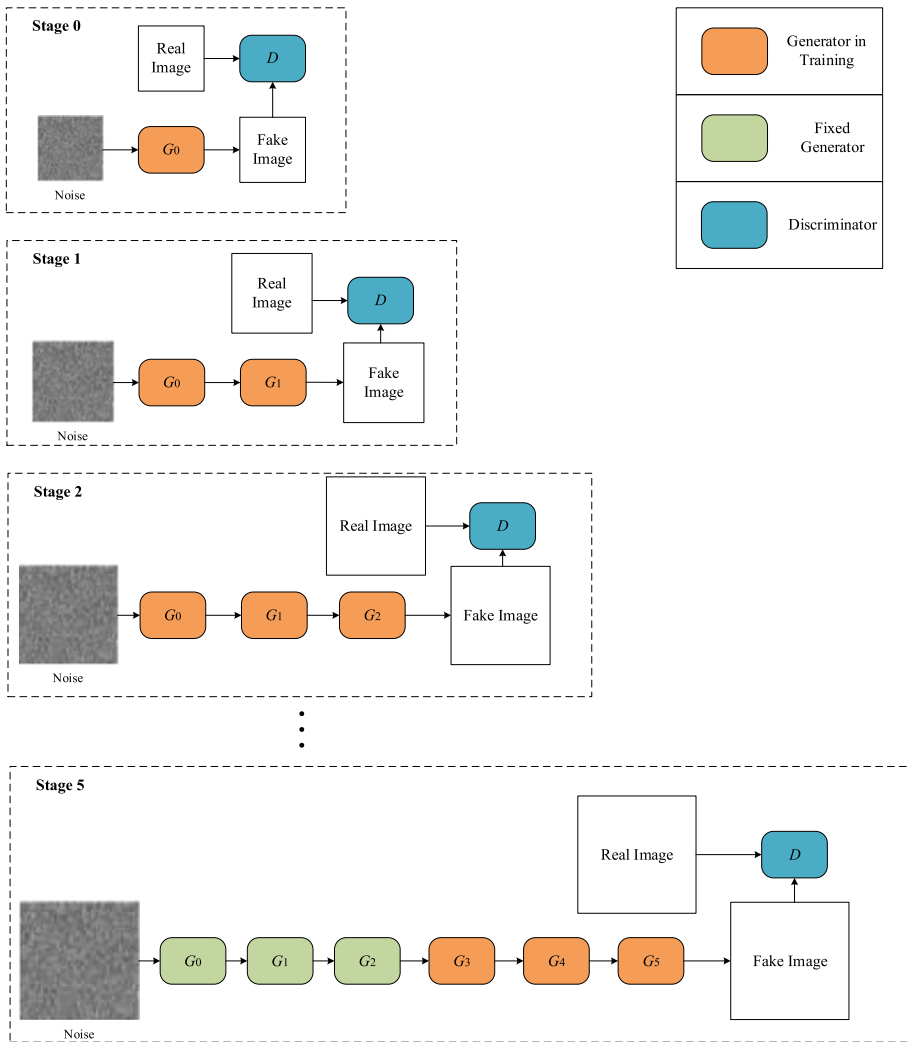


Fig. 3 An example of training process in ConSinGAN (the total stage number is 6; Max_Con=3)

3 Methodology

3.1 The Architecture of SRCSGAN

As mentioned above, ConSinGAN uses concurrent training to obtain global image information and local details at the same time to address the issue of the limited receptive field of GAN. But if the internal features of TIs are difficult to catch, then a proper way of extracting more details is to increase the layer number of convolutional layers in each stage of ConSinGAN. A deeper network understands and characterizes the features of TIs better, nevertheless surely increasing the risk of degradation problem in

deep networks. To overcome this problem, SRCSGAN integrates residual blocks with the idea of concurrent training together.

In addition, to accelerate the training process, the original features of the previous stage are upsampled and then input to the current stage. Upsampling is the process of enlarging the original image so that it can be displayed on a higher-resolution scale or display area. To reuse details generated in the previous stage, they are combined with the new learned information of the current stage by residual connections, which are defined as follows:

$$\tilde{x}_n = \tilde{x}_{n-1} \uparrow + \varphi_n(z_n + \tilde{x}_{n-1} \uparrow), \quad n = 1, \dots, N, \quad (3)$$

where \tilde{x}_n represents the result generated at stage n ; $\tilde{x}_{n-1} \uparrow$ represents the upsampling result from stage $n-1$; z_n represents the added noise in stage n for increasing the diversity of results; $\varphi_n(z_n + \tilde{x}_{n-1} \uparrow)$ represents the new learned information of the convolutional network at stage n .

The structure of SRCSGAN is shown in Fig. 4. For stage 0, multiple residual blocks are used to alleviate the risk of degradation. When the training of stage 0 reaches the convergence, stage 1 is added to the network to start the training. Residual connections are used to unify stage 0 and stage 1. The noises before each stage are used to increase the diversity of results. At this point, stage 1 only needs to slightly adjust its parameters obtained by stage 0 instead of reinitialization. Similarly, in stage 2 and subsequent stages, the network repeats this process until the final stage N .

Since the parameters of a stage are shared by its next stage, the whole adjustments of parameters are relatively small, thus greatly reducing the training time. In addition to the initial stage 0, the input of each stage includes random noises and the upsampled image features obtained from previous stages. Both the generator G and the discriminator D are trained separately. When the loss of one stage reaches the termination condition (a fixed epoch number for each stage), the generator of its next stage will be added to the current stage to form a training sequence for concurrent training, while the discriminator in each stage keeps unchanged. However, too many concurrent stages easily cause overfitting, so a maximum concurrent stage number (called Max_Con, set to 3 in this paper) is used in SRCSGAN, which means at most three stages simultaneously participate in the concurrent training process. If the current concurrent training number is bigger than Max_Con, the first stage in the training sequence will be dropped; otherwise, the next stage is added to the training sequence. This process will be repeatedly performed for N times until the desired stage N is reached, by which the final reconstructed result (i.e., reconstruction N in Fig. 4) is obtained. At each stage, the discriminator tries to estimate the probability that a sample is from the real image or a fake image. $L_{\text{all}}(G_n, D_n)$ is the loss function of SRCSGAN, which will be discussed in Sect. 3.4.

3.2 The Parameter Settings of SRCSGAN

In real experiments, the main parameters of the generator and discriminator are shown in Tables 1 and 2, in which some functions are described below.

1. Conv3d: a 3D convolution operation for extracting local features of 3D data.
2. BatchNorm3d: a regularization function that is used to normalize data, alleviating the problem of scattered distribution of data features and making the network model more stable.
3. LeakyReLU: an activation function that alleviates the vanishing gradient problem.

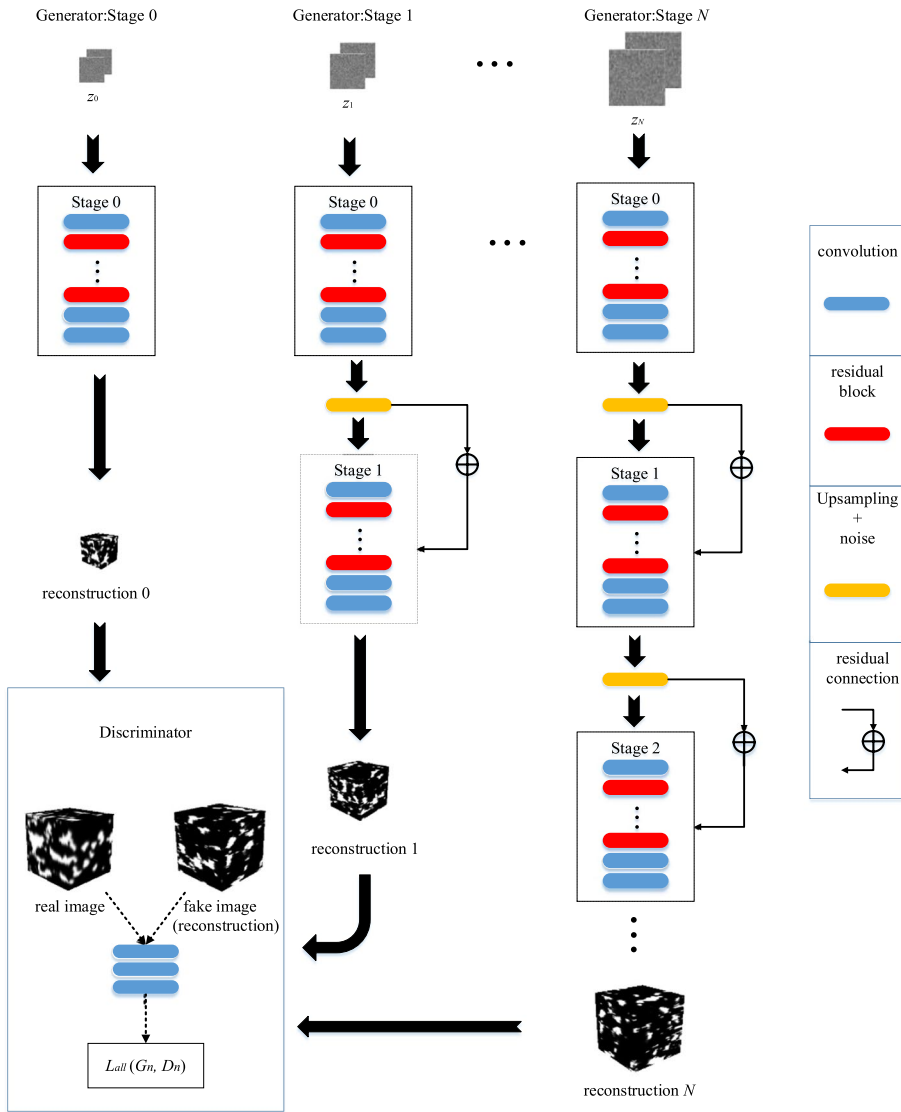


Fig. 4 The architecture of SRCSGAN

4. Tanh: an activation function that normalizes the generator output value to $[-1, 1]$.
5. Upsample: a 3D deconvolution operation that performs upsampling.
6. residual_block: the component of residual network.
7. residual_block \times 16: a total of 16 residual blocks.

Table 1 Main parameters used in the generator G

Layer	Function	Parameters
Layer 1	Conv3d BatchNorm3d LeakyReLU	In_channel=3, out_channel=64 kernel=(3, 3, 3) stride=(1, 1, 1)
residual_block $\times 16$	Conv3d	In_channel=64, out_channel=64 kernel=(3, 3, 3) stride=(1, 1, 1)
Layer 2	Conv3d BatchNorm3d LeakyReLU	In_channel=64, out_channel=64 kernel=(3, 3, 3) stride=(1, 1, 1)
Layer 3	Conv3d BatchNorm3d LeakyReLU	In_channel=64, out_channel=64 kernel=(3, 3, 3) stride=(1, 1, 1)
Layer 4	Conv3d BatchNorm3d LeakyReLU	In_channel=64, out_channel=64 kernel=(3, 3, 3) stride=(1, 1, 1)
Layer 5	Conv3d Upsample	In_channel=64, out_channel=256 kernel=(3, 3, 3) stride=(1, 1, 1) mode=trilinear
Layer 6	Conv3d Tanh	In_channel=256, out_channel=3 kernel=(9, 9, 9) stride=(1, 1, 1)

Table 2 Main parameters used in the discriminator D

Layer	Function	Parameters
Layer 1	Conv3d BatchNorm3d LeakyReLU	In_channel=3, out_channel=64 kernel=(3, 3, 3) stride=(1, 1, 1)
Layer 2	Conv3d BatchNorm3d LeakyReLU	In_channel=64, out_channel=64 kernel=(3, 3, 3) stride=(1, 1, 1)
Layer 3	Conv3d BatchNorm3d LeakyReLU	In_channel=64, out_channel=64 kernel=(3, 3, 3) stride=(1, 1, 1)
Layer 4	Conv3d BatchNorm3d LeakyReLU	In_channel=64, out_channel=64 kernel=(3, 3, 3) stride=(1, 1, 1)
Layer 5	Conv3d	In_channel=64, out_channel=1 kernel=(3, 3, 3) stride=(1, 1, 1)

3.3 The Size of Input Real Image at Each Stage

In SRCSGAN, the original TI is downsampled to smaller images or upsampled to bigger ones, all used as input real images for each stage. SRCSGAN needs small input images at lower stages to reflect the global information and larger input images at higher stages to represent detailed information using a fixed receptive field. These input images are used as TIs at each

stage. Therefore, the different sizes of input images at each stage need to be designed before downsampling or upsampling.

Suppose $y_n = [L_n, W_n, H_n]$ representing the size of a 3D input image at stage n , where L_n, W_n and H_n are, respectively, the length, width and height of the input image at stage n . The sizes at stage 0 and stage N , i.e., $y_n = [L_0, W_0, H_0]$ and $y_n = [L_N, W_N, H_N]$, should be, respectively, specified in advance. The size of the input image at each stage is defined as:

$$y_n = y_N \times r^{((N-1)/\log(N)) \times \log(N-n)+1}, n = 1, \dots, N - 1, \tag{4}$$

where the scaling factor r is:

$$r = (\min(L_0, W_0, H_0) / \min(L_N, W_N, H_N))^{\frac{1}{N}}, \tag{5}$$

Note that the generated image has the same size as the input image at each stage.

3.4 The Loss Functions of SRCSGAN

In the SRCSGAN model, each generator G_n is coupled with the discriminator D_n . For any stage $n \in \{0, 1, \dots, N\}$, the overall loss function $L_{\text{all}}(G_n, D_n)$ at stage n is consisted of the adversarial loss and reconstruction loss, which is defined as:

$$L_{\text{all}}(G_n, D_n) = \min_{G_n} \max_{D_n} \{L_{\text{adv}}(G_n, D_n) + \phi L_{\text{rec}}(G_n)\}, \tag{6}$$

where $L_{\text{adv}}(G_n, D_n)$ represents the adversarial loss; $L_{\text{rec}}(G_n)$ is the reconstruction loss; ϕ is the weight of $L_{\text{rec}}(G_n)$ and set as 10 by default.

In GAN and its variants, the discriminator D tries to distinguish between the input images and the generated images, and the generator G tries to deceive the discriminator by generating images similar to the input real images. The initial GAN uses the Sigmoid cross-entropy loss function, as shown in Eq. (1), which easily causes the problem of vanishing gradient, making the training of G insufficient. To make the generated samples fit the distribution of the real samples as much as possible, the WGAN-GP (Gulrajani et al. 2017) loss function is used in this paper, which is:

$$L_{\text{adv}}(G_n, D_n) = E_{\tilde{I}_n \sim P_g} [D(\tilde{I}_n)] - E_{I_n \sim P_r} [D(I_n)] + \varphi GP, \tag{7}$$

$$GP = E_{\hat{I}_n \sim \hat{I}_n} \left[\left(\left\| \nabla_{\hat{I}_n} D(\hat{I}_n) \right\|_2 - 1 \right)^2 \right], \tag{8}$$

$$\tilde{I}_n = \begin{cases} G(z_n), & n = 0 \\ G(z_n, \tilde{I}_{n-1}), & n = 1, \dots, N \end{cases}, \tag{9}$$

where \tilde{I}_n represents the fake image generated at stage n and I_n represents the input real image at stage n ; P_r is the real data distribution and P_g is the generator distribution; $I_n \sim \hat{I}_n$ means sampling uniformly along straight lines between pairs of points sampled from P_r and P_g (Gulrajani et al. 2017); $\nabla_{\hat{I}_n} D(\hat{I}_n)$ is the gradient with respect to \hat{I}_n ; \tilde{I}_{n-1} represents the upsampled result from stage $n-1$ to stage n ; \tilde{I}_n (except stage 0) is calculated from the random noise z_n at stage n and \tilde{I}_{n-1} ; $E_{\tilde{I}_n \sim P_g} [D(\tilde{I}_n)]$ is the mathematical expectation when the generated sample is used as the input of the discriminator; $E_{I_n \sim P_r} [D(I_n)]$ is the

mathematical expectation when the input of the discriminator is a real sample; φ is the weight of GP (gradient penalty) and is set as 0.1 in our experiments.

The main innovation of WGAN-GP lies in the GP term (i.e., Equation (8)) in the loss function. WGAN-GP uses Wasserstein distance to measure the distance between the distribution of generated data and the distribution of real data, which theoretically solves the problems of mode collapse and unstable training (e.g., vanishing gradient and exploding gradient). More detailed explanations can be found in Gulrajani et al. (2017).

The reconstruction loss is added as a constraint in Eq. (10), which is:

$$L_{\text{rec}}(G_n) = \left\| G_n(\tilde{I}_n) - I_n \right\|_2^2, \quad (10)$$

where $G_n(\tilde{I}_n)$ represents the reconstructed image at stage n .

3.5 Implementation

The procedure of SRCSGAN is displayed in Fig. 5.

Step 1 Initialize the training model, including setting parameters, the maximum stage number and the maximum concurrent stage number (i.e., Max_Con).

Step 2 Input the image of porous media as the original TI.

Step 3 Calculate the sizes of the input images at different stages according to Eq. (4); then upsample or downsample the TI to the above sizes.

Step 4 If the current stage number has not exceeded the maximum stage number, train the model and update parameters according to the loss function Eq. (6) until the termination condition (a fixed epoch number for each stage, e.g., 3000) is reached.

Step 5 For each stage, when the number of current concurrent stages is smaller than Max_Con, the next stage starts training; otherwise, the first stage in the training sequence is dropped and then add the next stage into the training sequence for concurrent training.

Step 6 If all the stages finish training, go to step 7; otherwise, go to step 4.

Step 7 Save the training model and the parameters; output the reconstructed results of porous media.

4 Experimental Results and Analyses

4.1 Experimental Data

4.1.1 Sandstone

As a kind of typical porous media, sandstone is widely distributed in the world, suitable for evaluating reconstruction quality of our method. The real sandstone data used in the experiments were extracted from a cylindrical sandstone sample (porosity 19% around) with a diameter of 3 mm, whose grayscale images of slices were obtained by the scanning of micro-CT (resolution is 10.9 μm) from the Beijing Synchrotron Microtomography Morphology Station, China. After denoising by mean filtering (Anindita et al. 2020) and binary segmentation by the Otsu method (Otsu 1979), 500 binary images (500^2 pixels) were achieved. By continuously stacking these 500 binary 500^2 2D images together,

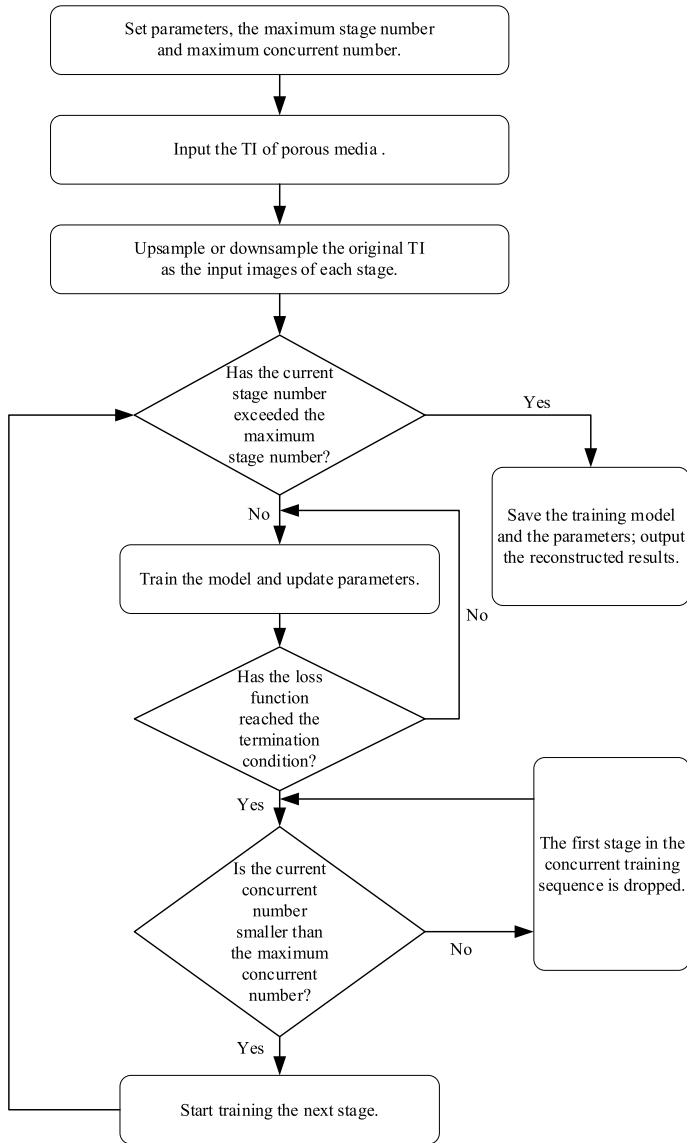


Fig. 5 The implementation of SRCSGAN

a 3D 500-cubed sandstone sample can be obtained. Then, a 3D cube (80^3 voxels, porosity=0.180) called HR_TI_SA is obtained from the 500-cubed sample for the following experiments.

4.1.2 Shale

Different porous media exhibits different pore structure features. To make our method more convincing, shale is also used to evaluate the reconstruction performance. A shale

sample (porosity 10% around) is scanned by nano-CT (resolution is 64 nm) from Langfang Petroleum Research Institute, Hebei Province, China. Similar to the sandstone sample, a 3D 500-cubed shale sample is obtained after denoising and binary segmentation. Then, a 3D cubic shale (80^3 voxels, porosity=0.097) called HR_TI_SH is obtained from the 500-cubed shale sample.

4.1.3 Preparation of Multiple-Stage Experimental Data

The trilinear interpolation (Kim et al. 2005) is used to, respectively, lower the resolution of HR_TI_SA and HR_TI_SH, obtaining the lower-resolution version of HR_TI_SA and HR_TI_SH, which is called LR_TI_SA and LR_TI_SH both with 40^3 voxels. The porosities of LR_TI_SA and LR_TI_SH are, respectively, 0.178 and 0.096, quite close to those of HR_TI_SA and HR_TI_SH. In the following experiments, all the reconstructed HR images of sandstone are reconstructed based on LR_TI_SA and evaluated by comparing with the HR ground truth image HR_TI_SA in reconstruction quality. Similarly, all the reconstructed HR images of shale are reconstructed based on LR_TI_SH and evaluated by comparing with the HR ground truth image HR_TI_SH. Therefore, when evaluating the reconstruction quality, the ground truth HR images, downsampled LR images and reconstructed HR images are all necessary.

Specifically, in the following experiments, sandstone and shale samples are, respectively, used to evaluate SRCSGAN. The deep learning platform is PyTorch 1.7.0. The learning rate is 0.0001 and the epoch number of each stage is set to 3000. The maximum concurrent stage number (i.e., Max_Con) is set to 3. For the reconstruction of both sandstone and shale, six training stages (i.e., stages 0–5) are designed in SRCSGAN, whose real input data for the discriminator are, respectively, LR_TI_SA (sandstone) and LR_TI_SH (shale). Then, LR_TI_SA and LR_TI_SH are, respectively, upsampled or downsampled to the different-sized images at different stages according to Eq. (4) by trilinear interpolation (Kim et al. 2005). Since the total stage number is six, there are six real input images correspondingly for each stage, i.e., stage 0 (16^3 voxels), stage 1 (20^3 voxels), stage 2 (25^3 voxels), stage 3 (34^3 voxels), stage 4 (58^3 voxels) and stage 5 (80^3 voxels). As shown in Fig. 6, a fixed receptive field (11^3 voxels) is used to scan these real input images at each stage to capture different features. At lower stages, the model mainly learns the overall information, while the focus is on learning the details of the image at higher stages.

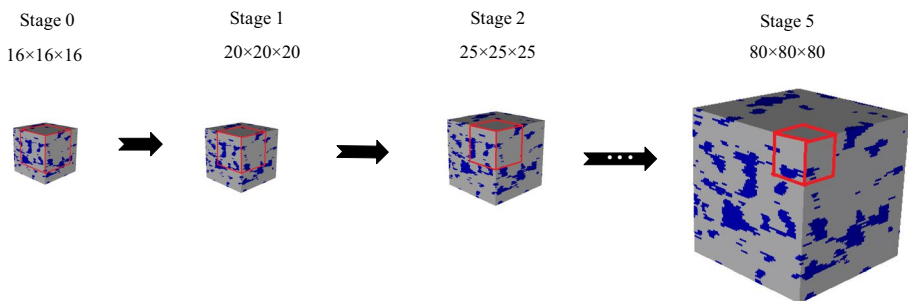


Fig. 6 A fixed receptive field (11^3 voxels) is used to scan the real input images at each stage to capture different features

4.2 The Reconstruction of Sandstone

4.2.1 The Determination of an REV

Since only the limited size of sandstone is used as training data in our experiments, the representative elementary volume (REV) (Costanza et al. 2011) of sandstone should be determined first. REV is the minimum volume that can reflect the statistical average features of materials. When the size or volume of a sample is less than an REV, its properties are easy to change with the different sizes of the sample itself, thus showing obvious fluctuations in features. Therefore, only when the sample is larger than an REV, the study of the sample is meaningful because its parameters are stable enough to reflect the average features of materials.

The REV in our experiments is determined by the multi-point connectivity (MPC) (Krishnan and Journel 2003; Strebelle 2002), which is used to measure the connectivity between multiple points and defined as follow:

$$\text{MPC}(n) = E\{I(u) \cdot I(u+h) \cdot \dots \cdot I(u+(n-1)h)\} = E\left\{\prod_{j=0}^{n-1} I(u+jh)\right\}, \quad (11)$$

where n is the number of pixels in one direction; u is the position of a certain pixel (in 2D cases) or voxel (in 3D cases) in space; h is the “lag distance” between neighboring pixels or voxels. The index variable $I(u)$ has two attribute values for porous media: when a specific position in the image is pore space, $I(u)=1$; otherwise, $I(u)=0$.

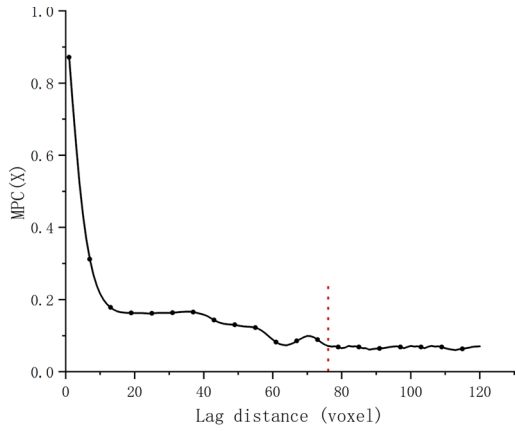
The MPC curves of the 500-cubed sandstone sample (described in Sect. 4.1.1) are computed and displayed in Fig. 7a–c, in which the MPC curves tend to be stable when $h=76, 70, 72$ (the positions represented by red dotted lines in Fig. 7) in the X, Y and Z directions, respectively, meaning the REV for our experiments should at least be $76 \times 70 \times 72$ voxels. Since our training sample, i.e., the HR ground truth image HR_TI_SA (80^3 voxels), is larger than the REV, the validity of our experiments can be assured.

4.2.2 Reconstructed Images Using SNESIM, MSPGAN, ConSinGAN and SRCSGAN

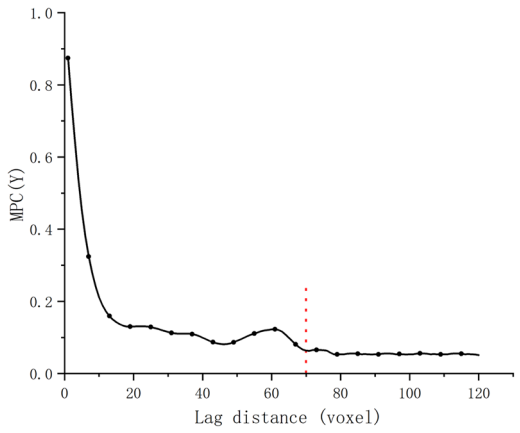
Some methods were specifically chosen for the comparison of reconstruction quality with SRCSGAN. Single normal equation simulation (SNESIM) (Strebelle 2002) is a typical MPS method, which has been widely used for the reconstruction of porous media (Okabe and Blunt 2004, 2005). SNESIM can reconstruct images that are much larger than the original TI and only needs one TI. As mentioned in Sect. 2.4, MSPGAN can obtain 3D SR reconstruction of porous media using only a single TI. Hence, MSPGAN and SNESIM are used as comparison methods. Besides, since our method (i.e., SRCSGAN) is derived from ConSinGAN, it is also included in the comparison.

To be specific, all the four methods use LR_TI_SA (40^3 voxels) as the only single TI and stochastically reconstruct 3D sandstone images with the size of 80^3 voxels, which will be compared with the HR ground truth image HR_TI_SA (80^3 voxels) by the metrics of MPC, porosity, absolute permeability, autocorrelation, etc. HR_TI_SA and the reconstructed images (80^3 voxels) using SNESIM, MSPGAN, ConSinGAN and SRCSGAN are shown in Fig. 8.

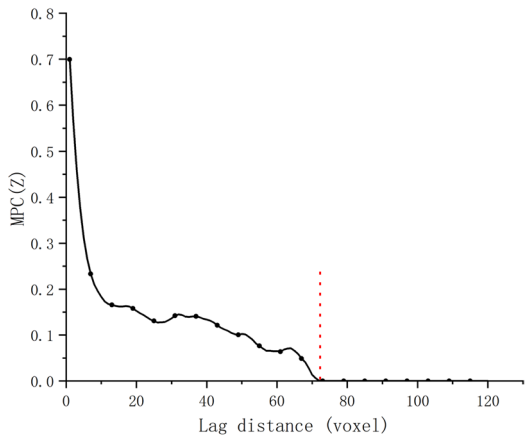
Fig. 7 The MPC curves of the 500-cubed sandstone sample (500^3 voxels)



(a) X direction

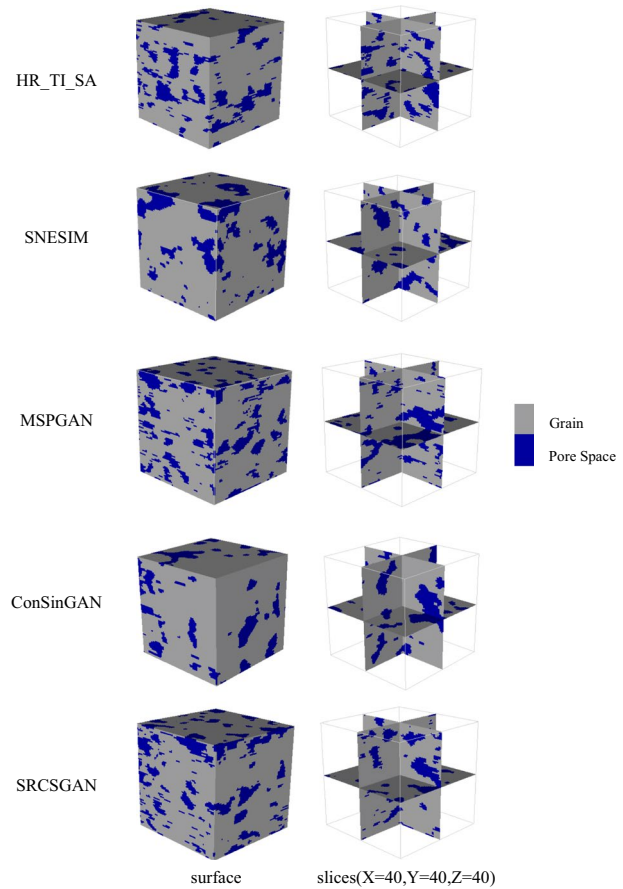


(b) Y direction



(c) Z direction

Fig. 8 HR_TI_SA and the reconstructed images (80^3 voxels) using SNESIM, MSPGAN, ConSinGAN and SRCSGAN



4.2.3 Comparison of Pore Space

The pore space in HR_TI_SA and reconstructions can be divided into connected and isolated pore space, but only the connected pore space is considered as valid contribution to permeability. The total pore space, connected pore space and isolated pore space of HR_TI_SA and reconstructions using SNESIM, MSPGAN, ConSinGAN and SRCSGAN in Fig. 8 are extracted using the software Avizo (Avizo 2015), which are displayed in Fig. 9. To obtain average statistics, another ten reconstructed results, respectively, by SNESIM, MSPGAN, ConSinGAN and SRCSGAN were obtained. The average proportions of the total pore space, connected pore space and isolated pore space, respectively, calculated from these results and the proportion of HR_TI_SA are shown in Table 3. Bold emphasis means the bold numbers are best compared to other statistics in the same metrics. It is seen that SRCSGAN has better performance than other methods due to its higher similarity with HR_TI_SA.

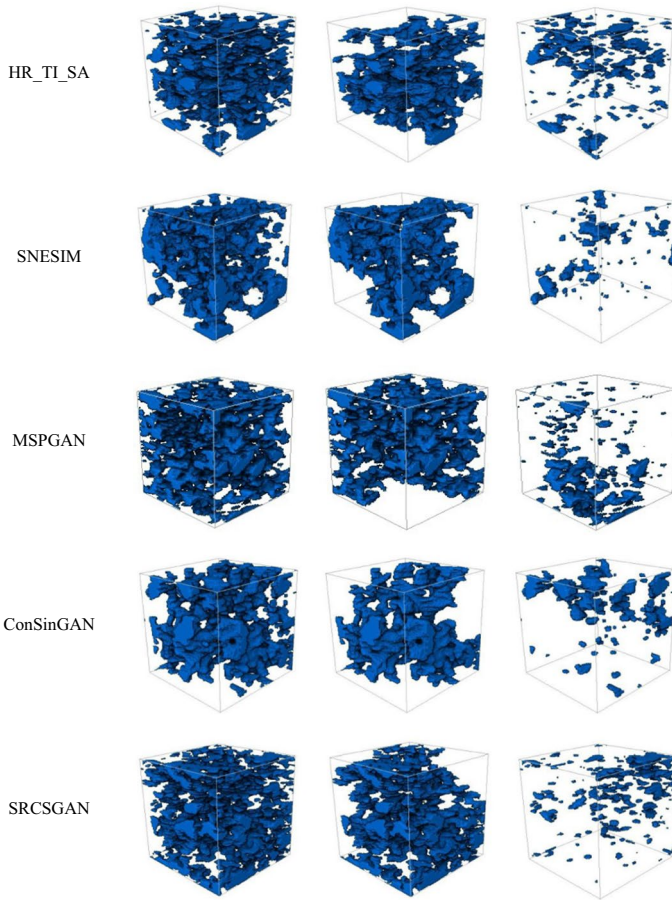


Fig. 9 The total pore space, connected pore space and isolated pore space of HR_TI_SA and reconstructions using SNESIM, MSPGAN, ConSinGAN and SRCSGAN (in Fig. 8)

Table 3 The average proportions of the total pore space, connected pore space and isolated pore space calculated from another ten reconstructed results, respectively, by SNESIM, MSPGAN, ConSinGAN and SRCSGAN and the proportion of the HR_TI_SA

Type	HR_TI_SA	SNESIM	MSPGAN	ConSinGAN	SRCSGAN
Total pore space (%)	18.0	20.9	19.8	20.0	18.3
Connected pore space (%)	15.1	19.7	17.5	17.0	15.7
Isolated pore space (%)	2.9	1.2	2.3	3.0	2.6

4.2.4 Comparison of Autocorrelation Functions

The autocorrelation function of 3D microstructure is widely used in statistical reconstruction as an evaluation tool. Suppose \vec{r} represents any point in the microstructure. When the microstructure is defined by the following binary phase function $Z(\vec{r})$:

$$Z(\vec{r}) = \begin{cases} 1, & \vec{r} \in \text{pore space} \\ 0, & \vec{r} \notin \text{pore space} \end{cases}, \tag{12}$$

the autocorrelation function $R_z(\vec{u})$ is defined as:

$$\varphi = \overline{Z(\vec{r})}, \tag{13}$$

$$R_z(\vec{u}) = \frac{\overline{[Z(\vec{r}) - \varphi][Z(\vec{r} + \vec{u}) - \varphi]}}{[\varphi - \varphi^2]}, \tag{14}$$

where overbars mean statistical averages; φ is porosity; \vec{u} is a lag vector between two vectors.

Figure 10 shows the autocorrelation functions of HR_TI_SA and reconstructions by SNESIM, MSPGAN, ConSinGAN and SRCSGAN, averaged over three orthogonal directions. It seems that the autocorrelation function of SRCSGAN is more similar to that of HR_TI_SA.

4.2.5 Comparison of Absolute Permeability

The absolute permeability is used to indicate the magnitude of single-phase fluid flow (Zhang et al. 2016). HR_TI_SA and the reconstructions are, respectively, used as the input files to calculate the absolute permeabilities of those models using the software Avizo (Avizo 2015). The absolute permeabilities of HR_TI_SA and the average absolute permeabilities of ten reconstructed images using SNESIM, MSPGAN, ConSinGAN and SRCSGAN in three directions are shown in Table 4, from which it is seen that the absolute permeabilities of SRCSGAN are closest to those of HR_TI_SA in the Y and Z directions, only a little worse than MSPGAN in the X direction.

Fig. 10 The autocorrelation functions of HR_TI_SA and reconstructions by SNESIM, MSPGAN, ConSinGAN and SRCSGAN, averaged over three orthogonal directions

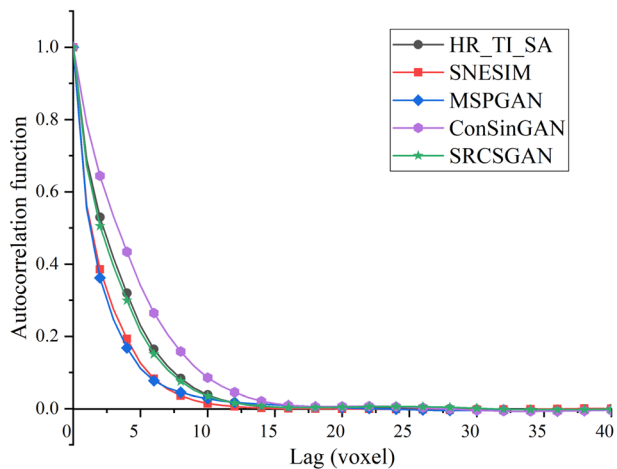


Table 4 The absolute permeabilities of HR_TI_SA and the average absolute permeabilities of ten reconstructed images using SNESIM, MSPGAN, ConSinGAN and SRCSGAN in three directions

Direction	Absolute permeability ($10^{-3} \mu\text{m}^2$)				
	HR_TI_SA	SNESIM	MSPGAN	ConSinGAN	SRCSGAN
X	0.346	0.258	0.353	0.385	0.369
Y	0.258	0.343	0.233	0.293	0.268
Z	0.333	0.269	0.372	0.316	0.313

Bold emphasis means the bold numbers are best compared to other statistics in the same metrics

4.2.6 Comparison of MPC Curves

Figure 11 shows the MPC curves of HR_TI_SA and average MPC curves of ten reconstructed images by SNESIM, MSPGAN, ConSinGAN and SRCSGAN in the X, Y and Z directions, but the differences of MPC between HR_TI_SA and ConSinGAN as well as the differences between HR_TI_SA and SRCSGAN in the X and Y directions are quite close. To quantitatively measure the differences of MPC between HR_TI_SA and reconstructions, a difference degree (DD) is defined as:

$$\text{DD}(\text{HR_TI, re}) = \sum_{h=1}^n (X_h - x_h)^2, \quad (15)$$

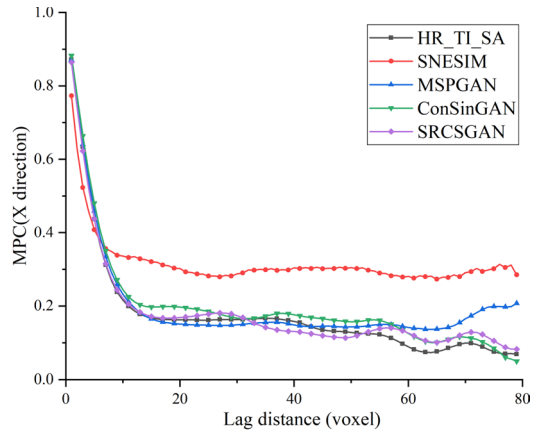
where HR_TI is the HR ground truth image; re is the reconstruction result; X_h and x_h are the MPC values of HR_TI and the reconstructed image with a distance of h , respectively. The smaller the DD is, the higher similarity between HR_TI and the reconstruction. Note that in this paper HR_TI, respectively, means HR_TI_SA or HR_TI_SH. DDs between HR_TI_SA and the reconstructions by these four methods, measured by MPC (as shown in Fig. 11) in the X, Y and Z directions, are displayed in Table 5. It is seen that reconstructions by SRCSGAN are closest to HR_TI_SA since the DDs are smallest in three directions.

4.2.7 Comparison of Pore Network Models

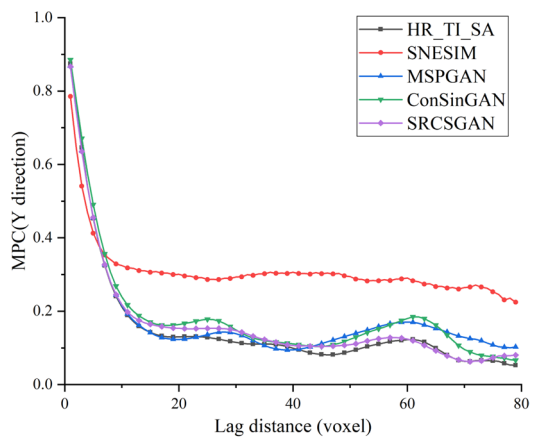
The pore network model (PNM) has the equivalent topological structure with the pore space of the digitalized porous media, which reflects the topological structure of real pore space. A maximal ball method (Dong and Blunt 2009) was used to extract PNM, and the whole pore space is divided into throats and pores, in which a pore is replaced by a sphere to represent a relatively wide pore space, while a throat is replaced by a cylinder to represent a narrow and long pore space.

The comparisons of PNM were performed using Avizo (Avizo 2015) to analyze the pore distribution of HR_TI_SA and the reconstructions, including pore numbers, throat numbers, pore diameters, throat diameters and coordination numbers. The pore number and throat number of HR_TI_SA and the average of ten reconstructions by SNESIM, MSPGAN, ConSinGAN and SRCSGAN are shown in Table 6. It can be seen that the reconstruction by SRCSGAN is closest to HR_TI_SA in both pore number and throat number.

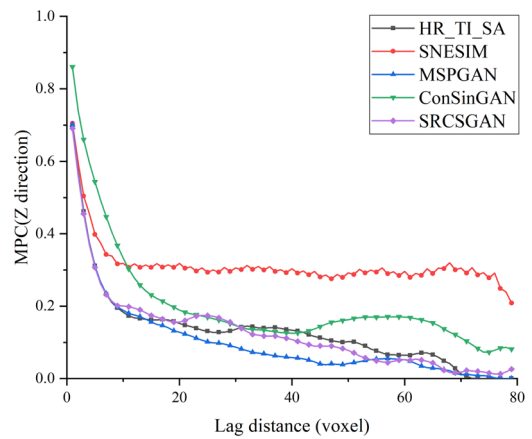
Fig. 11 The MPC curves of HR_TI_SA and average MPC curves of ten reconstructed images by SNESIM, MSPGAN, ConSinGAN and SRCSGAN in the X, Y and Z directions



(a) X direction



(b) Y direction



(c) Z direction

Table 5 Difference degrees (DDs) between HR_TI_SA and the reconstructions by SNESIM, MSPGAN, ConSinGAN and SRCSGAN measured by MPC (as shown in Fig. 11) in the X, Y and Z directions

Direction	Difference degrees (DDs)			
	DD(HR_TI_SA, SNESIM)	DD(HR_TI_SA, MSPGAN)	DD(HR_TI_SA, ConSinGAN)	DD(HR_TI_SA, SRCSGAN)
X	2.041	0.102	0.087	0.031
Y	2.375	0.118	0.117	0.091
Z	2.909	0.317	0.704	0.038

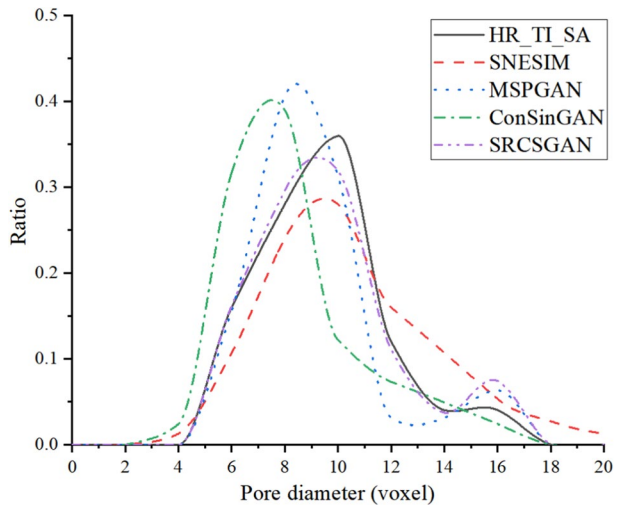
Bold emphasis means the bold numbers are best compared to other statistics in the same metrics

Table 6 The pore number and throat number of HR_TI_SA and the average pore numbers and throat numbers of ten reconstructions, respectively, using SNESIM, MSPGAN, ConSinGAN and SRCSGAN

	HR_TI_SA	SNESIM	MSPGAN	ConSinGAN	SRCSGAN
Pore number	25	75	32	41	27
Throat number	33	312	45	52	38

Bold emphasis means the bold numbers are best compared to other statistics in the same metrics

Fig. 12 Average pore diameter ratios of HR_TI_SA and reconstructed images by SNESIM, MSPGAN, ConSinGAN and SRCSGAN



The ratios of pore diameters, throat diameters and coordination numbers are shown in Figs. 12, 13 and 14. DDs are calculated to quantitatively show the differences between HR_TI_SA and the reconstructions, as shown in Table 7. It can be seen that SRCSGAN performs best.

Fig. 13 Average throat diameter ratios of HR_TI_SA and reconstructed images by SNESIM, MSPGAN, ConSinGAN and SRCSGAN

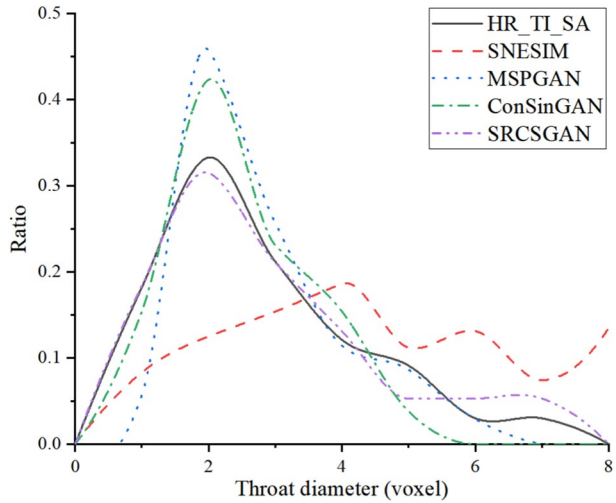


Fig. 14 Average coordination number ratios of HR_TI_SA and reconstructed images by SNESIM, MSPGAN, ConSinGAN and SRCSGAN

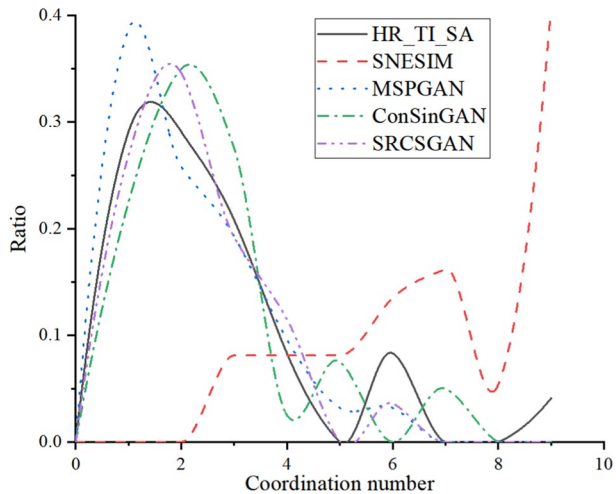


Table 7 Difference degrees (DDs) between HR_TI_SA and the reconstructions by SNESIM, MSPGAN, ConSinGAN and SRCSGAN measured by the ratio curves of pore diameters, throat diameters and coordination numbers

Type	Difference degrees (DDs)			
	DD (HR_TI_SA, SNESIM)	DD(HR_TI_SA, MSPGAN)	DD(HR_TI_SA, ConSinGAN)	DD(HR_TI_SA, SRCSGAN)
Pore diameter	0.018	0.027	0.097	0.007
Throat diameter	0.091	0.034	0.015	0.013
Coordination number	0.357	0.016	0.032	0.009

Bold emphasis means the bold numbers are best compared to other statistics in the same metrics

4.2.8 The Diversity of Reconstruction Results

As mentioned above, the loss function of WGAN-GP can solve mode collapse (Gulrajani et al. 2017). Since SRCSGAN also uses the same loss function as WGAN-GP, the mode collapse problem can be avoided and enough diversity will be generated in reconstructions. To prove this point, an exclusive OR (XOR) operation is defined to quantitatively measure the similarity between the reconstruction results and HR_TI_SA. For two integer numbers a and b , if the two numbers are same, then $XOR(a, b) = 0$; otherwise, $XOR(a, b) = 1$. The percentage of 0 in the XOR operation result is used as the similarity degree, named "SD." A SD close to 0 means that the similarity of two reconstructions is very high, while a SD close to 1 means they are totally different. Table 8 shows the SDs between HR_TI_SA and each reconstruction (respectively, called R_n , $n = 1, \dots, 10$) by SRCSGAN. Note that Table 8 is symmetric along the diagonal and all SDs are equal to 0 in the diagonal direction. All SDs fluctuate between 0.45 and 0.62, which display enough diversity between HR_TI_SA and each reconstruction, meaning that SRCSGAN generates reasonable diversity in reconstruction results.

4.2.9 The Settings of the Maximum Concurrent Stage Number

In the actual experiment, it is found that using a large maximum concurrent stage number (i.e., Max_Con) can shorten the training time, but it also leads to the problem of overfitting. Therefore, it is particularly important to reasonably select Max_Con.

Max_Con is set to 1, 3 and 5, respectively, to evaluate the overfitting problem. Note that when the concurrent number is 1, it means that the training process actually is not concurrent. For each situation of Max_Con (= 1, 3, 5, respectively), ten reconstructions were generated to obtain the average statistics. Through the comparison of MPC curves in Fig. 15 and DDs in Table 9, it is found that when Max_Con=5, the reconstructions are highly similar to HR_TI_SA.

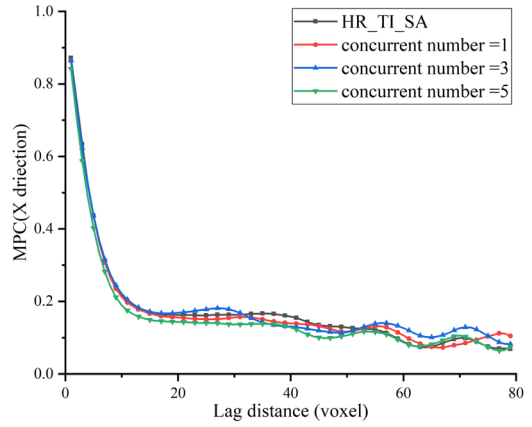
The XOR operation is also performed between HR_TI_SA and the reconstructions (Max_Con=1, 3, 5, respectively). When Max_Con=1, the XOR operation between

Table 8 Similarity degrees (SDs) between HR_TI_SA and each reconstruction by SRCSGAN

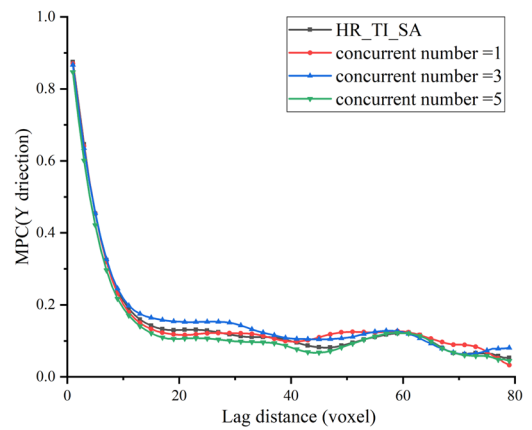
	HR_TI_SA	R ₁	R ₂	R ₃	R ₄	R ₅	R ₆	R ₇	R ₈	R ₉	R ₁₀
HR_TI_SA	0	0.55	0.54	0.48	0.49	0.50	0.56	0.59	0.53	0.52	0.51
R ₁	0.55	0	0.57	0.53	0.52	0.54	0.53	0.49	0.51	0.57	0.61
R ₂	0.54	0.57	0	0.53	0.49	0.54	0.56	0.55	0.59	0.58	<u>0.45</u>
R ₃	0.48	0.53	0.53	0	0.56	0.57	0.55	0.58	0.56	0.61	0.57
R ₄	0.49	0.52	0.49	0.56	0	0.48	0.57	<u>0.62</u>	0.61	0.55	0.54
R ₅	0.50	0.54	0.54	0.57	0.48	0	0.55	0.59	0.51	0.55	0.60
R ₆	0.56	0.53	0.56	0.55	0.57	0.55	0	0.56	0.53	0.54	0.53
R ₇	0.59	0.49	0.55	0.58	<u>0.62</u>	0.59	0.56	0	0.51	0.55	0.55
R ₈	0.53	0.51	0.59	0.56	0.61	0.51	0.53	0.51	0	0.53	0.49
R ₉	0.52	0.57	0.58	0.61	0.55	0.55	0.54	0.55	0.53	0	0.57
R ₁₀	0.51	0.61	<u>0.45</u>	0.57	0.54	0.60	0.53	0.55	0.49	0.57	0

Note this table is symmetric along the diagonal and all SDs are equal to 0 in the diagonal direction

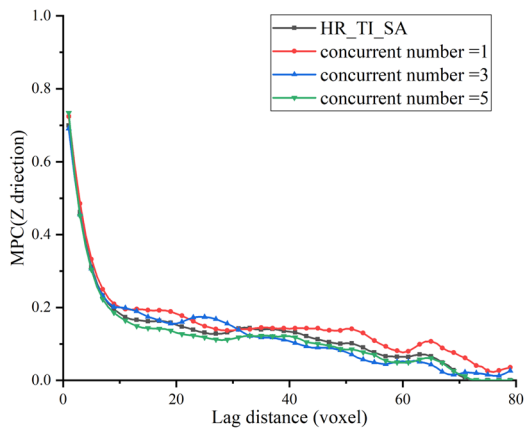
Fig. 15 MPC curves of HR_TI_SA and reconstructions with different maximum concurrent numbers (Max_Con = 1, 3, 5) by SRCSGAN in the X, Y and Z directions



(a) X direction



(b) Y direction



(c) Z direction

Table 9 Difference degrees (DDs) between HR_TI_SA and the SRCSGAN reconstructions with different maximum concurrent number (Max_Con = 1, 3 and 5) measured by MPC curves in the X, Y and Z directions

Direction	Difference degrees (DDs)		
	Concurrent number = 1	Concurrent number = 3	Concurrent number = 5
X	0.041	0.031	0.016
Y	0.124	0.091	0.018
Z	0.116	0.038	0.029

Bold emphasis means the bold numbers are best compared to other statistics in the same metrics

HR_TI_SA and ten reconstructions R_1, R_2, \dots, R_{10} (expressed as $XOR(HR_TI_SA, R_i)$, $i = 1, 2, \dots, 10$) is calculated to obtain an average SD, which is 0.683. Similarly, the average SDs in the situations of Max_Con=3 and 5 are, respectively, calculated, which are 0.544 and 0.335, respectively. It is concluded that although the reconstruction quality is best when Max_Con=5, its corresponding SD (i.e., 0.335) displays a relatively high similarity between the reconstructions and HR_TI_SA, showing a high degree of overfitting. Meanwhile, when Max_Con=1, the reconstruction quality is unsatisfactory. Therefore, it seems that setting Max_Con to 3 can balance the overfitting risk and reconstruction quality.

4.2.10 The Settings of the Total Stage Number

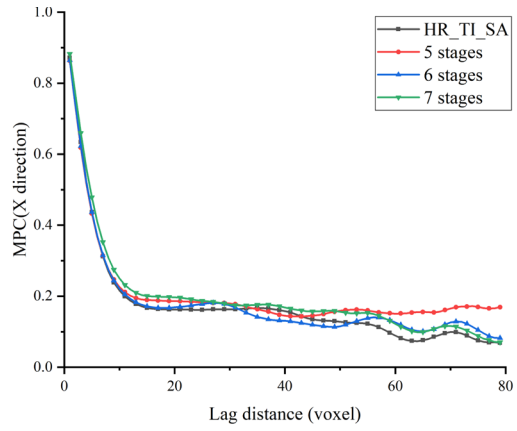
Since the total stage number is variable, the influence of the total stage number on the reconstructed results is discussed in this section. To facilitate the discussion, other parameters keep unchanged in the experiment. Table 10, respectively, shows the reconstruction times for 5, 6 and 7 stages of SRCSGAN. It can be seen that with the increase in training stages, the reconstruction times also increase.

Figure 16 shows the MPC curves of HR_TI_SA and the reconstructed images with different training stages by SRCSGAN in the X, Y and Z directions. Normally increasing layers can improve reconstruction quality first, but when the reconstruction quality reaches a plateau, more layers will not lead to evident improvement in reconstruction quality. On the contrary, too many layers possibly cause the fluctuation in reconstruction quality. Meanwhile, the corresponding reconstruction times surely will increase with the increase in layer numbers, so the reconstruction time and quality should be balanced in real simulation. Table 11 is the DDs between HR_TI_SA and the SRCSGAN reconstructions of 5, 6 and 7 stages measured by MPC curves in the X, Y and Z directions. Judging from Fig. 16 and Table 11, it seems that a 6-stage SRCSGAN is a proper balance for the reconstruction quality and time in our experiments.

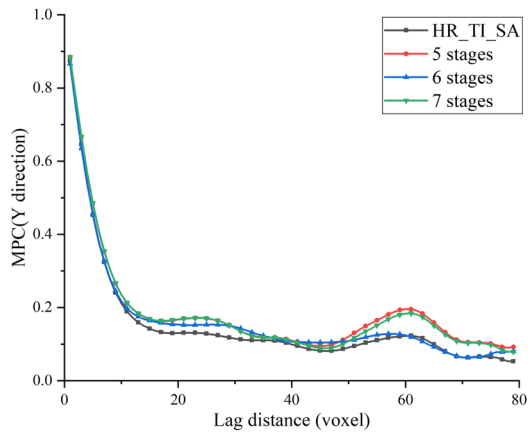
Table 10 The reconstruction times using different training stages of SRCSGAN

Total stage number	Reconstruction time (s)
5	19,981
6	20,293
7	25,852

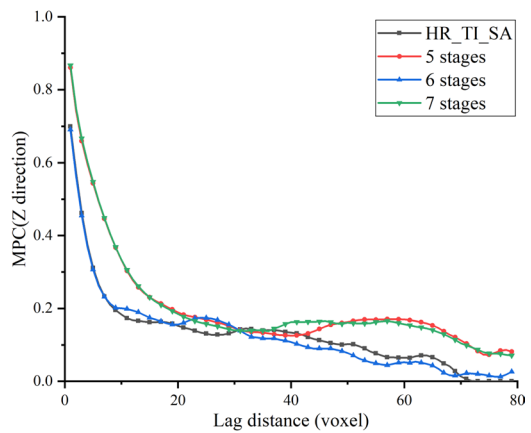
Fig. 16 MPC curves of HR_TL_SA and reconstructions with different training stages (5, 6 and 7 stages) by SRCSGAN in the X, Y and Z directions



(a) X direction



(b) Y direction



(c) Z direction

Table 11 Difference degrees (DDs) between HR_TI_SA and the SRCSGAN reconstructions (5, 6 and 7 stages) measured by MPC curves in the X, Y and Z directions

Direction	Difference degrees (DDs)		
	5 stages	6 stages	7 stages
X	0.157	0.031	0.052
Y	0.115	0.091	0.089
Z	0.704	0.038	0.685

Bold emphasis means the bold numbers are best compared to other statistics in the same metrics

Table 12 The reconstruction times and usage of CPU, GPU and memory by SNESIM, MSPGAN, ConSinGAN and SRCSGAN for ten reconstructions

	SNESIM	MSPGAN	ConSinGAN	SRCSGAN
Average CPU usage	56%	40%	35%	35%
Average GPU usage	0	95%	90%	93%
Average memory usage	49%	30%	40%	41%
Time for the first reconstruction (s)	1860	8538	18,935	20,292
Time for the other 9 reconstructions (s)	16,200	1	1	1

Bold emphasis means the bold numbers are best compared to other statistics in the same metrics

Table 13 The reconstruction times using different-sized LR_TI_SA to reconstruct different-sized HR_TI_SA by SRCSGAN

Size of LR_TI_SA (voxel)	Size of HR_TI_SA (voxel)	Reconstruction time (s)
32 ³	64 ³	10,849
40 ³	80 ³	20,293

4.2.11 The Comparison of Reconstruction Time and CPU/GPU/Memory Usage

Since the reconstruction is quite CPU-intensive and normally costs a large amount of memory, the average reconstruction time and CPU, GPU and memory usage of SNESIM, MSPGAN, ConSinGAN and SRCSGAN are evaluated in this section. The experiments in the whole manuscript were performed using the AMD R9 3900X 3.8 GHz CPU, 16 GB memory and GeForce RTX3080 GPU with 10 GB video memory. In our experiments, SNESIM are based on CPU, while MSPGAN, ConSinGAN and SRCSGAN use both CPU and GPU. Table 12 shows the reconstruction times and average usage of CPU, GPU and memory by SNESIM, MSPGAN, ConSinGAN and SRCSGAN for ten sandstone reconstructions (80³ voxels). Note that the "reconstruction times" include both the training times and generation times.

The whole reconstruction time consists of "the first reconstruction time" and "the other nine reconstruction times". As can be seen from Table 12, although MSPGAN, ConSinGAN and SRCSGAN cost much longer time in the first reconstruction, the subsequent nine reconstructions can be carried out extremely fast by reusing existing models and

parameters, which is also the advantage of deep learning methods over traditional methods like SNESIM.

4.2.12 The Influence of the Size of the LR Training Image

The size of the LR image LR_TI_SA has an evident impact on the reconstruction time, especially in 3D reconstruction. Suppose the sizes of LR_TI_SA are 32^3 and 40^3 , respectively; correspondingly, the sizes of HR_TI_SA are set to 64^3 and 80^3 voxels, respectively. The reconstruction times using different-sized LR_TI_SA to reconstruct different-sized HR_TI_SA by SRCSGAN are shown in Table 13. It is seen that the reconstruction time of 40^3 is nearly doubled compared to that of 32^3 . It is concluded that a small increase in the size of the LR image may lead to a sharp increase in reconstruction time.

4.3 The Reconstruction of Shale

4.3.1 The Determination of an REV

Similar to the sandstone sample, the MPC curves of the 500-cubed shale sample (500^3 voxels) is computed and displayed in Fig. 17, in which the MPC curves tend to be stable when $h = 60, 30$ 50 (the positions represented by red dotted lines in Fig. 17) in the X, Y and Z directions, respectively, meaning the REV for our experiments should at least be $60 \times 30 \times 50$ voxels. Since the shale sample HR_TI_SH (80^3 voxels) is larger than the REV, the validity of our experiments can be assured.

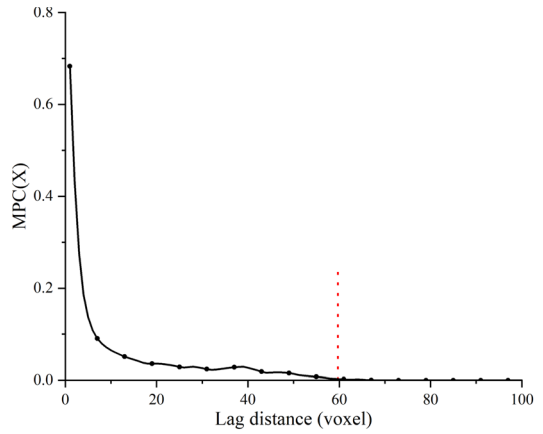
4.3.2 Reconstructed Images Using SNESIM, MSPGAN, ConSinGAN and SRCSGAN

Similar to the sandstone reconstruction, the reconstructions by SNESIM, MSPGAN, ConSinGAN and SRCSGAN are compared. To be specific, all the four methods use LR_TI_SH (40^3 voxels) as the only single TI and stochastically reconstruct 3D shale images with the size of 80^3 voxels, which will be compared with the ground truth 3D shale image HR_TI_SH (80^3 voxels) by the metrics of MPC, porosity, absolute permeability, etc. HR_TI_SH and the reconstructed images (80^3 voxels) using SNESIM, MSPGAN, ConSinGAN and SRCSGAN are shown in Fig. 18.

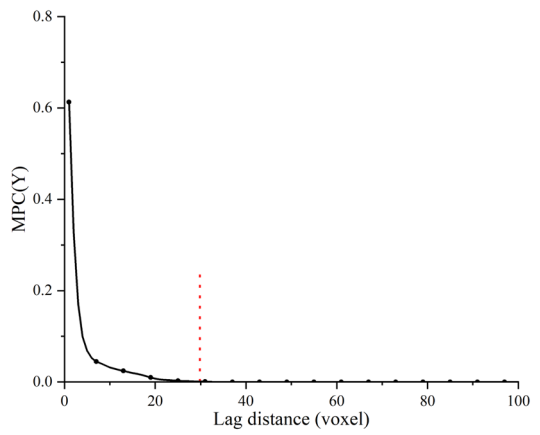
4.3.3 Comparison of Pore Space

The total pore space, connected pore space and isolated pore space of HR_TI_SH and reconstructions using SNESIM, MSPGAN, ConSinGAN and SRCSGAN in Fig. 18 are extracted using the software Avizo (Avizo 2015), which are displayed in Fig. 19. To obtain average statistics, another ten reconstructed results, respectively, by SNESIM, MSPGAN, ConSinGAN and SRCSGAN were realized. The average proportions of the total pore space, connected pore space and isolated pore space, respectively, calculated from these ten results and the proportion of HR_TI_SH are shown in Table 14. It is seen that SRCSGAN has better overall performance than other methods due to its higher similarity with HR_TI_SH in the proportions of the connected pore space and isolated pore space.

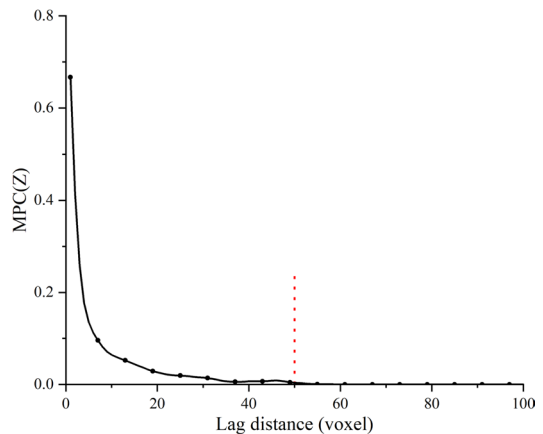
Fig. 17 The MPC curves of the 500-cubed shale sample (500^3 voxels)



(a) X direction



(b) Y direction



(c) Z direction

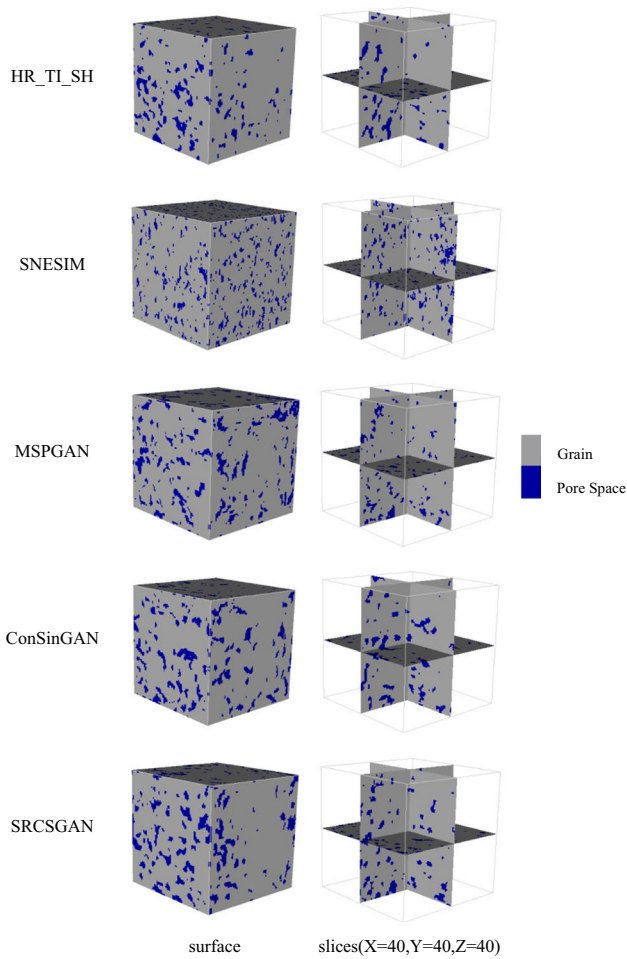


Fig. 18 HR_TL_SH and the reconstructed images (80^3 voxels) using SNESIM, MSPGAN, ConSinGAN and SRCSGAN

4.3.4 Comparison of Autocorrelation Functions

Figure 20 shows the autocorrelation functions of HR_TL_SH and reconstructions by SNESIM, MSPGAN, ConSinGAN and SRCSGAN, averaged over three orthogonal directions. It seems that the autocorrelation curves of SRCSGAN and HR_TL_SH almost overlap, meaning the autocorrelation curve by SRCSGAN is highly similar to that of HR_TL_SH.

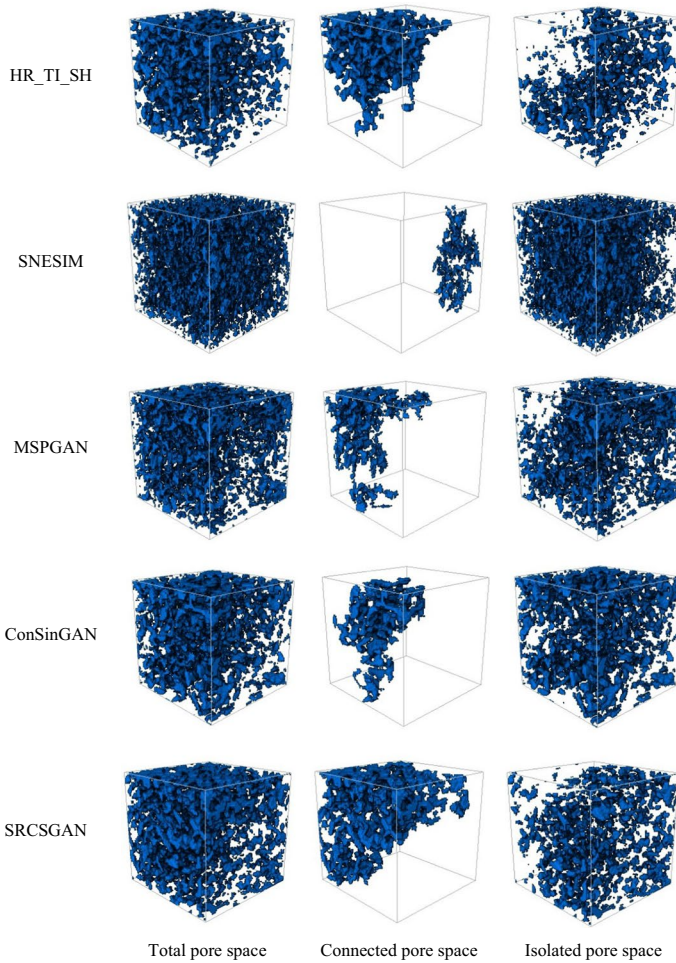


Fig. 19 The total pore space, connected pore space and isolated pore space of HR_TI_SH and reconstructions using SNESIM, MSPGAN, ConSinGAN and SRCSGAN (in Fig. 18)

Table 14 The average proportions of the total pore space, connected pore space and isolated pore space calculated from another ten reconstructed results, respectively, by SNESIM, MSPGAN, ConSinGAN and SRCSGAN and the proportion of the HR_TI_SH

Type	HR_TI_SH	SNESIM	MSPGAN	ConSinGAN	SRCSGAN
Total pore space (%)	9.69	9.33	8.81	10.27	10.93
Connected pore space (%)	5.45	0.56	1.93	2.13	5.65
Isolated pore space (%)	4.24	8.77	6.88	8.14	5.28

Bold emphasis means the bold numbers are best compared to other statistics in the same metrics

Fig. 20 The autocorrelation functions of HR_TI_SH and reconstructions by SNESIM, MSPGAN, ConSinGAN and SRCSGAN, averaged over three orthogonal directions

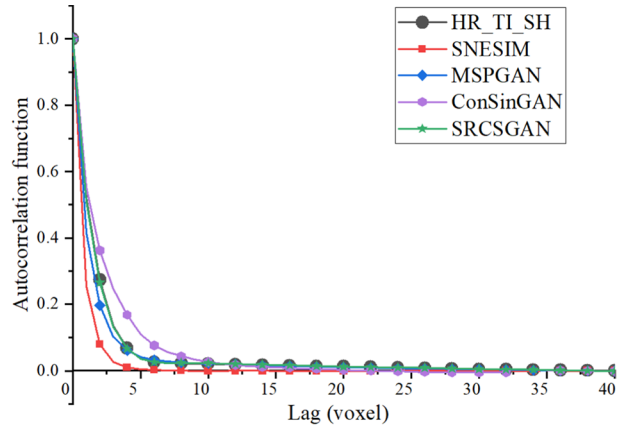


Table 15 The absolute permeabilities of HR_TI_SH and the average absolute permeabilities of ten reconstructed images using SNESIM, MSPGAN, ConSinGAN and SRCSGAN in three directions

Direction	Absolute permeability ($10^{-3} \mu m^2$)				
	HR_TI_SH	SNESIM	MSPGAN	ConSinGAN	SRCSGAN
X	0.332	0.516	0.431	0.381	0.364
Y	0.367	0.499	0.454	0.360	0.367
Z	0.352	0.529	0.399	0.373	0.359

Bold emphasis means the bold numbers are best compared to other statistics in the same metrics

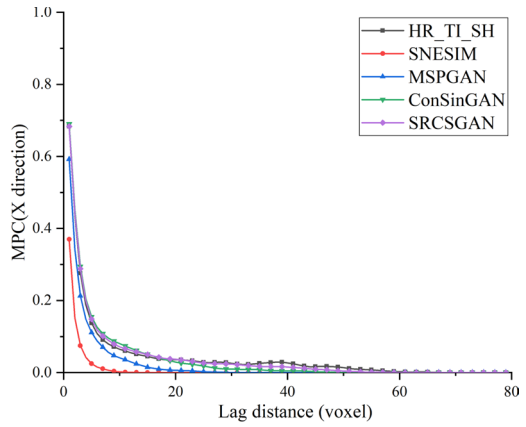
4.3.5 Comparison of Absolute Permeability

The absolute permeabilities of HR_TI_SH and the average absolute permeabilities of ten reconstructed images using SNESIM, MSPGAN, ConSinGAN and SRCSGAN in three directions are shown in Table 15, from which it is seen that the absolute permeabilities of SRCSGAN are closest to those of HR_TI_SH in all three directions, displaying favorable reconstruction quality.

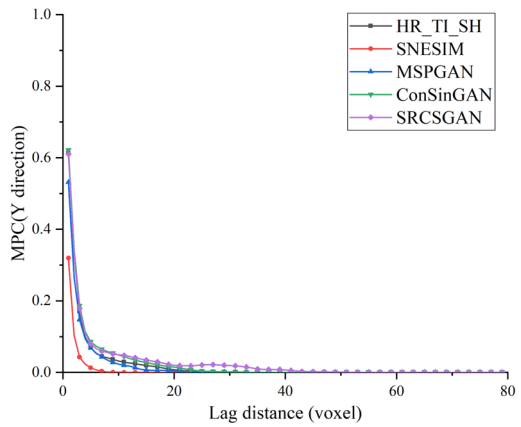
4.3.6 Comparison of MPC Curves

Figure 21 shows the MPC curves of HR_TI_SH and the average curves of ten reconstructed images by SNESIM, MSPGAN, ConSinGAN and SRCSGAN in the X, Y and Z directions. DDs between HR_TI_SH and the reconstructions are displayed in Table 16, measured by MPC in three directions. It is seen that SRCSGAN performs best.

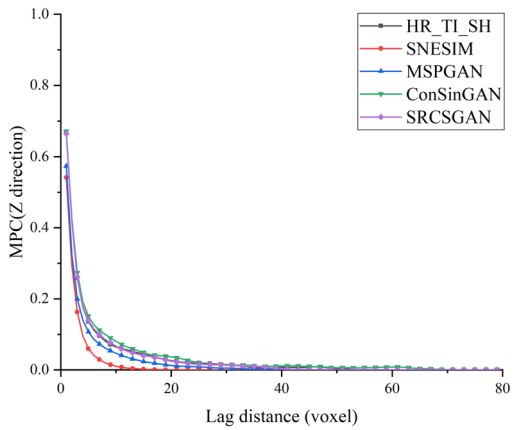
Fig. 21 The MPC curves of HR_TL_SH and average MPC curves of ten reconstructed images by SNESIM, MSPGAN, ConSinGAN and SRCSGAN in the X, Y and Z directions



(a) X direction



(b) Y direction



(c) Z direction

Table 16 Difference degrees (DDs) between HR_TI_SH and the reconstructions by SNESIM, MSPGAN, ConSinGAN and SRCSGAN measured by MPC (as shown in Fig. 21) in the X, Y and Z directions

Direction	Difference degrees (DDs)			
	DD(HR_TI_SH, SNESIM)	DD(HR_TI_SH, MSPGAN)	DD(HR_TI_SH, ConSinGAN)	DD(HR_TI_SH, SRCSGAN)
X	0.322	0.052	0.017	0.013
Y	0.174	0.012	0.014	0.010
Z	0.093	0.031	0.014	0.011

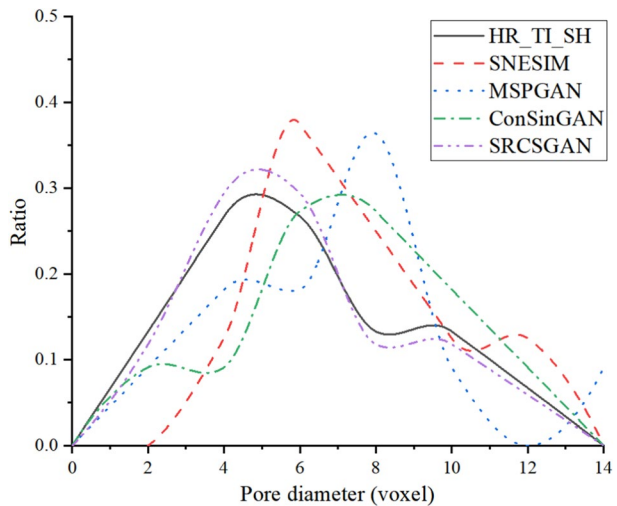
Bold emphasis means the bold numbers are best compared to other statistics in the same metrics

Table 17 The pore number and throat number of HR_TI_SH and the average pore numbers and throat numbers of ten reconstructions, respectively, using SNESIM, MSPGAN, ConSinGAN and SRCSGAN

	HR_TI_SH	SNESIM	MSPGAN	ConSinGAN	SRCSGAN
Pore number	15	8	11	11	17
Throat number	21	10	16	14	23

Bold emphasis means the bold numbers are best compared to other statistics in the same metrics

Fig. 22 Average pore diameter ratios of HR_TI_SH and reconstructed images by SNESIM, MSPGAN, ConSinGAN and SRCSGAN



4.3.7 Comparison of Pore Network Models

The pore number and throat number of HR_TI_SH and the average of ten reconstructions by SNESIM, MSPGAN, ConSinGAN and SRCSGAN are shown in Table 17. It can be seen that the reconstruction by SRCSGAN is closest to HR_TI_SH in the pore number and throat number.

Fig. 23 Average throat diameter ratios of HR_TI_SH and reconstructed images by SNESIM, MSPGAN, ConSinGAN and SRCSGAN

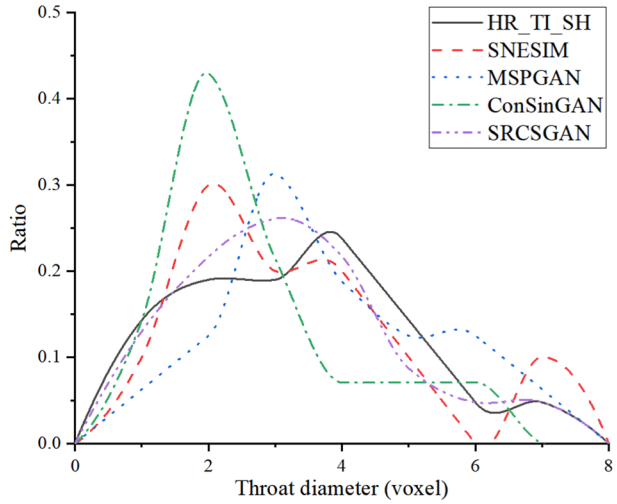


Fig. 24 Average coordination number ratios of HR_TI_SH and reconstructed images by SNESIM, MSPGAN, ConSinGAN and SRCSGAN

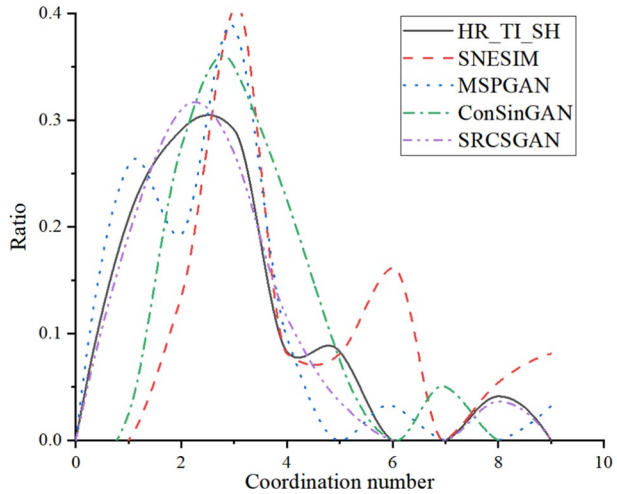


Table 18 Difference degrees (DDs) between HR_TI_SH and the reconstructions by SNESIM, MSPGAN, ConSinGAN and SRCSGAN measured by the ratio curves (Figs. 22, 23 and 24) of pore diameters, throat diameters and coordination numbers

Type	Difference degrees (DDs)			
	DD (HR_TI, SNESIM)	DD (HR_TI, MSPGAN)	DD (HR_TI, ConSinGAN)	DD (HR_TI, SRCSGAN)
Pore diameter	0.067	0.084	0.055	0.003
Throat diameter	0.042	0.035	0.094	0.010
Coordination number	0.114	0.032	0.062	0.005

Bold emphasis means the bold numbers are best compared to other statistics in the same metrics

Table 19 Similarity degrees (SDs) between HR_TI_SH and ten reconstructions by SRCSGAN

	HR_TI_SH	R ₁	R ₂	R ₃	R ₄	R ₅	R ₆	R ₇	R ₈	R ₉	R ₁₀
HR_TI_SH	0	<u>0.43</u>	0.47	0.51	0.49	0.55	0.52	0.49	0.47	0.53	0.58
R ₁	<u>0.43</u>	0	0.55	0.54	0.52	0.61	0.53	0.59	0.51	0.57	0.50
R ₂	0.47	0.55	0	0.59	0.63	0.54	0.61	0.55	0.49	0.52	0.45
R ₃	0.51	0.54	0.59	0	0.54	0.51	0.52	0.50	0.48	0.51	0.49
R ₄	0.49	0.52	0.63	0.54	0	0.48	0.57	0.62	0.64	0.55	0.44
R ₅	0.55	0.61	0.54	0.51	0.48	0	0.54	0.49	0.51	0.60	0.55
R ₆	0.52	0.53	0.61	0.52	0.57	0.54	0	0.59	0.53	0.66	0.47
R ₇	0.49	0.59	0.55	0.50	0.62	0.49	0.59	0	0.68	0.55	0.51
R ₈	0.47	0.51	0.49	0.48	0.64	0.51	0.53	0.68	0	0.47	0.49
R ₉	0.53	0.57	0.52	0.51	0.55	0.60	0.66	0.55	0.47	0	<u>0.70</u>
R ₁₀	0.58	0.50	0.45	0.49	0.44	0.55	0.47	0.51	0.49	<u>0.70</u>	0

Note this table is symmetric along the diagonal and all SDs are equal to 0 in the diagonal direction. The underlined numbers respectively mean the largest (i.e. 0.70) and smallest numbers (i.e. 0.43).

The ratios of pore diameters, throat diameters and coordination numbers are shown in Figs. 22, 23 and 24. DDs are calculated to quantitatively show the differences between HR_TI_SH and the average of reconstructions measured by the ratio curves (Figs. 22, 23, 24) of pore diameters, throat diameters and coordination numbers, as shown in Table 18. It can be seen that SRCSGAN performs best.

4.3.8 The Diversity of Reconstruction Results

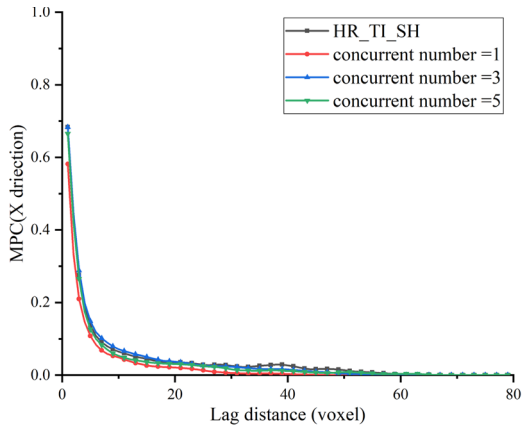
Table 19 shows the similarity degrees (SDs) between HR_TI_SH and ten reconstructions (called R_{*n*}, *n* = 1, ..., 10) by SRCSGAN. Note that Table 19 is symmetric along the diagonal and all SDs are equal to 0 in the diagonal direction. The SDs fluctuate between 0.43 and 0.70, displaying enough diversity between HR_TI_SH and each reconstruction.

4.3.9 The Settings of the Maximum Concurrent Stage Number

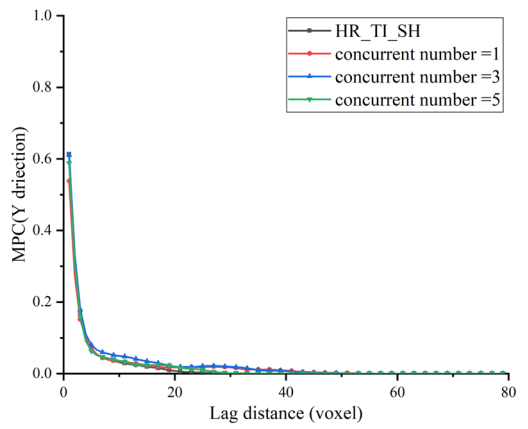
Similar to the sandstone reconstruction, Max_Con is, respectively, set to 1, 3 and 5 to evaluate the overfitting problem. For each situation of Max_Con (= 1, 3, 5, respectively), there are ten reconstructions to obtain the average statistics. Through the comparison of MPC curves in Fig. 25 and DDs in Table 20, it is found that when Max_Con = 5, the reconstructions are highly similar to HR_TI_SH.

The XOR operation is also performed between HR_TI_SH and the reconstructions (Max_Con = 1, 3, 5, respectively). In the situation of Max_Con = 1, the XOR (HR_TI_SH, R_{*i*}), *i* = 1, 2, ..., 10, can be calculated to obtain its average SD, which is 0.728. Similarly, the average SDs in the situations of Max_Con = 3 and 5 are, respectively, 0.537 and 0.265. It is concluded that when Max_Con = 5, a high degree of overfitting occurs, similar to the situation of sandstone reconstruction in Sect. 4.2.9. Meanwhile, when Max_Con = 1, the reconstruction quality is worse than that of Max_Con = 3.

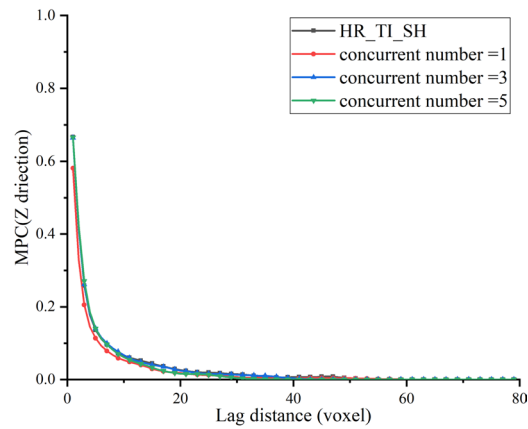
Fig. 25 MPC curves of HR_TI_SH and reconstructions with different maximum concurrent numbers (Max_Con = 1, 3, 5) by SRCSGAN in the X, Y and Z directions



(a) X direction



(b) Y direction



(c) Z direction

Table 20 Difference degrees (DDs) between HR_TI_SH and the SRCSGAN reconstructions with different maximum concurrent numbers (Max_Con = 1, 3, 5) measured by MPC curves in the X, Y and Z directions

Direction	Difference degrees (DDs)		
	Concurrent number = 1	Concurrent number = 3	Concurrent number = 5
X	0.022	0.013	0.006
Y	0.018	0.010	0.001
Z	0.029	0.011	0.002

Bold emphasis means the bold numbers are best compared to other statistics in the same metrics

Table 21 The reconstruction times using different training stages of SRCSGAN

Total stage number	Reconstruction time (s)
5	17,837
6	19,031
7	24,141

Therefore, setting Max_Con to 3 is a proper choice to balance overfitting and reconstruction quality.

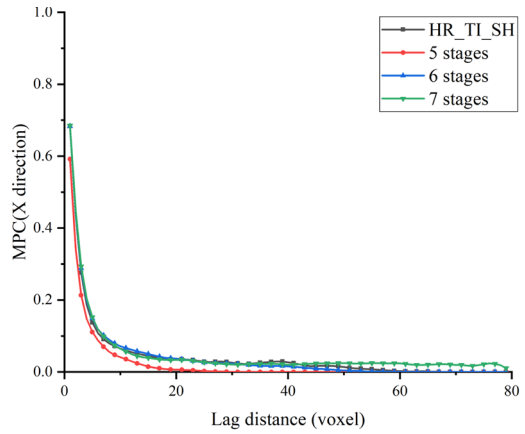
4.3.10 The Settings of the Total Stage Number

Similar to the sandstone reconstruction, Table 21, respectively, shows the reconstruction times for 5, 6 and 7 stages in SRCSGAN. It can be seen that with the increase in training stages, the reconstruction times also increase. Figure 26 shows the MPC curves of HR_TI_SH and the reconstructed images with different training stages by SRCSGAN in the X, Y and Z directions. Table 22 displays the DDs between HR_TI_SH and the SRCSGAN reconstructions of 5, 6 and 7 stages measured by MPC curves in the X, Y and Z directions. Although the reconstruction of 7 stages has the best quality, it also needs the longest reconstruction time. Based on the reconstruction times in Table 21, MPC curves shown in Fig. 26 and DDs in Table 22, it is concluded that a 6-stage SRCSGAN is a proper balance for the reconstruction quality and time in our experiments, similar to the situation of sandstone (discussed in Sect. 4.2.10).

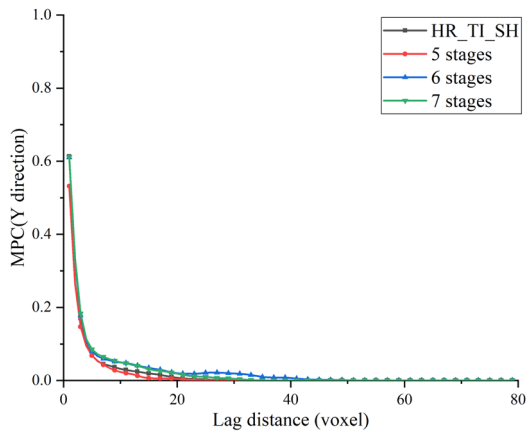
4.3.11 The Comparison of Reconstruction Time and CPU/GPU/Memory Usage

Table 23 shows the reconstruction times and average usage of CPU, GPU and memory by SNESIM, MSPGAN, ConSinGAN and SRCSGAN for ten shale reconstructions. Similar to the sandstone reconstruction, MSPGAN, ConSinGAN and SRCSGAN are much slower than SNESIM in the first reconstruction, but extremely fast in the remaining nine reconstructions by reusing the established parameters and models in the first reconstruction.

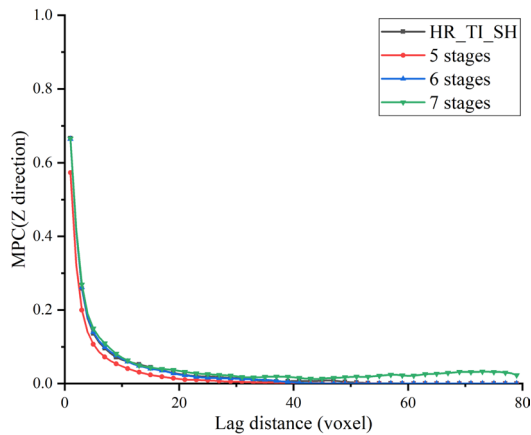
Fig. 26 MPC curves of HR_TL_SH and reconstructions with different training stages (5, 6 and 7 stages) by SRCSGAN in the X, Y and Z directions



(a) X direction



(b) Y direction



(c) Z direction

Table 22 Difference degrees (DDs) between HR_TI_SH and the SRCSGAN reconstructions (5, 6 and 7 stages) measured by MPC curves in the X, Y and Z directions

Direction	Difference degrees (DDs)		
	5 stages	6 stages	7 stages
X	0.052	0.013	0.112
Y	0.012	0.010	0.007
Z	0.030	0.011	0.124

Bold emphasis means the bold numbers are best compared to other statistics in the same metrics

Table 23 The reconstruction times and usage of CPU, GPU and memory by SNESIM, MSPGAN, ConSinGAN and SRCSGAN for ten shale reconstructions

	SNESIM	MSPGAN	ConSinGAN	SRCSGAN
Average CPU usage	56%	40%	35%	35%
Average GPU usage	0	95%	90%	93%
Average memory usage	49%	30%	40%	41%
Time for the first reconstruction (s)	1790	7756	17,991	19,030
Time for the other 9 reconstructions (s)	15,800	1	1	1

Bold emphasis means the bold numbers are best compared to other statistics in the same metrics

Table 24 The reconstruction times using different-sized LR_TI_SH to reconstruct different-sized HR_TI_SH by SRCSGAN

Size of LR_TI_SA (voxel)	Size of HR_TI_SA (voxel)	Reconstruction time (s)
32^3	64^3	9769
40^3	80^3	19,031

4.3.12 The Influence of the Size of the LR Training Image

Suppose the sizes of LR_TI_SH are 32^3 and 40^3 , respectively; correspondingly, the sizes of HR_TI_SH are set to 64^3 and 80^3 voxels, respectively. The reconstruction times using different-sized LR_TI_SH to reconstruct different-sized HR_TI_SH by SRCSGAN are shown in Table 24. It is seen that the reconstruction time of 40^3 is much longer than that of 32^3 . Hence, it is concluded that a small increase in the size of the LR image causes a sharp increase in reconstruction time, also similar to the sandstone reconstruction.

5 Conclusion

With the development of deep learning, some GAN-based variants have been proposed for the 3D reconstruction of porous media, but they often need abundant images as training data. Meanwhile, due to the limitations of hardware equipment and the real-time

requirements of information transmission and processing, the image data in practical applications are often LR images and their quantity is possibly insufficient for the training tasks.

To address the above issues, a 3D reconstruction method SRCSGAN based on ConSinGAN and residual networks is proposed for the SR reconstruction of porous media, which can realize SR stochastic reconstruction from LR images in the situation of insufficient training data. The experimental results show that SRCSGAN has favorable reconstruction quality, whose advantages are summarized as follows:

1. Compared to SNESIM, MSPGAN and ConSinGAN, SRCSGAN has proved its accuracy in the SR reconstruction of porous media by sandstone and shale samples in the metrics of pore space, permeability, autocorrelation functions, MPC curves and pore network models.
2. SRCSGAN can save and reuse the training model after the first training, so the subsequent reconstructions can be finished in a very short time when a large number of reconstruction tasks need to be performed, showing its high efficiency in multiple reconstruction tasks.

The main disadvantage of SRCSGAN lies in the lengthy training time in the first-round reconstruction, which is almost a common challenge for most deep learning methods. The problem will become more serious when the reconstruction size is quite large. The improvement of hardware conditions and optimized design in the architecture of deep learning networks are two possible ways to solve the problem. Our study will continuously focus on optimizing the architecture of SRCSGAN to hopefully shorten the training time in the first-round reconstruction.

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

Availability of data and materials Not applicable.

Code availability Not applicable.

Declarations

Conflict of interest The authors declare that they have no conflicts of interest.

References

- Anindita, K., Sumanta, B., Chittabarni, S., Souptik, B.: An axis based mean filter for removing high-intensity salt and pepper noise. In: 2020 IEEE Calcutta Conference (CALCON), Kolkata, India, pp. 363–367 (2020)
- Avizo: Avizo User's Guide, 9th edn. FEI, USA (2015)
- Costanza, R.M.S., Estabrook, B.D., Fouhey, D.F.: Representative elementary volume estimation for porosity, moisture saturation, and air-water interfacial areas in unsaturated porous media. Data quality implications. *Water Resour. Res.* **47**(7), 07513.1–07513.12 (2011)
- Dong, H., Blunt, M.J.: Pore-network extraction from micro-computerized-tomography images. *Phys. Rev. E* **80**, 036307 (2009)
- Dong, C., Loy, C.C., He, K., Tang, X.: Learning a deep convolutional network for image super-resolution. In: European Conference on Computer Vision, pp. 184–199. Springer, Cham (2014)

- Feng, J., He, X., Teng, Q., Ren, C., Chen, H., Li, Y.: Accurate and fast reconstruction of porous media from extremely limited information using conditional generative adversarial network. *Phys. Rev. E*. **100**(3), 33308 (2019). <https://doi.org/10.1103/PhysRevE.100.033308>
- Fredrich, J.T., Lindquist, W.B.: Statistical characterization of the three-dimensional microgeometry of porous media and correlation with macroscopic transport properties. *Int. J. Rock Mech. Min. Sci.* **34**(3–4), Paper No. 085 (1997)
- Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. *Adv. Neural. Inf. Process. Syst.* **3**, 2672–2680 (2014)
- Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of Wasserstein GANs. *NIPS* (2017). [arXiv:1704.00028](https://arxiv.org/abs/1704.00028)
- Hazlett, R.D.: Statistical characterization and stochastic modeling of pore networks in relation to fluid flow. *Math. Geol.* **29**(6), 801–822 (1997)
- He, K.M., Zhang, X.Y., Ren, S.Q., et al.: Deep residual learning for image recognition. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
- Helene, B., Didier, R.: Truncated Gaussian and derived methods. *Compte Rendus Geosci.* **348**(7), 510–519 (2015)
- Hou, J., Zhang, S.K., Sun, R.Y., Li, Z.Q., Li, Y.B.: Reconstruction of 3D network model through CT scanning. In: 69th European Association of Geoscientists and Engineers Conference and Exhibition 2007: Securing the Future. Incorporating SPE EUROPEC 2007, vol. 6, pp. 3398–3408 (2007)
- Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: convolutional architecture for fast feature embedding (2014). [arXiv:1408.5093](https://arxiv.org/abs/1408.5093)
- Kim, S.M., Park, J.C., Lee, K.H.: Depth-image based full 3D modeling using trilinear interpolation and distance transform. In: Proceedings of the 2005 International Conference on Augmented Tele-Existence (ICAT '05). Association for Computing Machinery, New York, NY, USA, 259–260 (2005)
- Kim, J., Lee, J.K., Lee, K.M.: Accurate image super-resolution using very deep convolutional networks. In: Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1646–1654 (2016)
- Krishnan, S., Journel, A.G.: Spatial connectivity: From variograms to multiple-point measures. *Math. Geol.* **35**(8), 915–925 (2003)
- Ledig, C., Theis, L., Huszar, F., et al.: Photo realistic single image super-resolution using a generative adversarial network. In: Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4681–4690 (2017)
- Lim, B., Son, S., Kim, H., Nah, S., Mu, Lee, K.: Enhanced deep residual networks for single image super-resolution. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 136–144 (2017)
- Lymberopoulos, D.P., Payatakes, A.C.: Derivation of topological, geometrical, and correlational properties of porous media from pore-chart analysis of serial section data. *J. Colloid Interface Sci.* **150**(1), 61–80 (1992)
- Mosser, L., Dubrule, O., Blunt, M.J.: Reconstruction of three-dimensional porous media using generative adversarial neural net-works. *Phys. Rev. E*. **96**(4), 043309 (2017)
- Okabe, H., Blunt, M.J.: Prediction of permeability for porous media reconstructed using multiple-point statistics. *Phys. Rev. E Comput. Geosci. Stat. Nonlinear Soft Matter Phys.* **70**(62), 066135 (2004)
- Okabe, H., Blunt, M.J.: Pore space reconstruction using multiple-point statistics. *J. Pet. Sci. Eng.* **46**(1–2), 121–137 (2005)
- Otsu, N.: A threshold selection method from gray-level histograms. *IEEE Trans. Syst. Man Cybern.* **9**(1), 62–66 (1979)
- Shaham, T.R., Dekel, T., Michaeli, T.: Singan: learning a generative model from a single natural image. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 4570–4580 (2019)
- Singh, M., Mohanty, K.K.: Permeability of spatially correlated porous media. *Chem. Eng. Sci.* **55**(22), 5393–5403 (2000)
- Strebelle, S.: Conditional simulation of complex geological structures using multiple-point statistics. *Math. Geol.* **34**, 1–21 (2002)
- Tobias, H., Matthew, F., Oliver, W., Stefan, W.: Improved techniques for training single-image GANs. In: 2020 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020). [arXiv:2003.11512](https://arxiv.org/abs/2003.11512)
- Valsecchi, A., Damas, S., Tubilleja, C., Arechalde, J.: Stochastic reconstruction of 3D porous media from 2D images using generative adversarial networks. *Neurocomputing* **399**, 227–236 (2020)
- Vogel, H.J., Roth, K.: Quantitative morphology and network representation of soil pore structure. *Adv. Water Resour.* **24**(3–4), 233–242 (2001)

- Zhang, T., Du, Y., Huang, T., Yang, J., Lu, F., Li, X.: Reconstruction of porous media using ISOMAP-based MPS. *Stoch. Environ. Res. Risk A* **30**(1), 395–412 (2016)
- Zhang, T., Ji, X., Lu, F.: 3D reconstruction of porous media by combining scaling transformation and multi-scale discrimination using generative adversarial networks. *J. Petrol. Sci. Eng.* **209**, 109815 (2022)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.