






Deep learning method of stochastic reconstruction of three-dimensional digital cores from a two-dimensional image

Juan Li ^{1,2,*} Qizhi Teng ^{1,†} Ningning Zhang ¹ Honggang Chen ¹ and Xiaohai He ¹

¹College of Electronics and Information Engineering, Sichuan University, Chengdu 610065, China

²College of Electrical Engineering, Northwest Minzu University, Lanzhou 730030, China



(Received 16 August 2022; revised 11 April 2023; accepted 6 May 2023; published 30 May 2023)

Digital cores can characterize the true internal structure of rocks at the pore scale. This method has become one of the most effective ways to quantitatively analyze the pore structure and other properties of digital cores in rock physics and petroleum science. Deep learning can precisely extract features from training images for a rapid reconstruction of digital cores. Usually, the reconstruction of three-dimensional (3D) digital cores is performed by optimization using generative adversarial networks. The training data required for the 3D reconstruction are 3D training images. In practice, two-dimensional (2D) imaging devices are widely used because they can achieve faster imaging, higher resolution, and easier identification of different rock phases, so replacing 3D images with 2D ones avoids the difficulty of acquiring 3D images. In this paper, we propose a method, named EWGAN-GP, for the reconstruction of 3D structures from a 2D image. Our proposed method includes an encoder, a generator, and three discriminators. The main purpose of the encoder is to extract statistical features of a 2D image. The generator extends the extracted features into 3D data structures. Meanwhile, the three discriminators have been designed to gauge the similarity of morphological characteristics between cross sections of the reconstructed 3D structure and the real image. The porosity loss function is used to control the distribution of each phase in general. In the entire optimization process, a strategy using Wasserstein distance with gradient penalty makes the convergence of the training process faster and the reconstruction result more stable; it also avoids the problems of gradient disappearance and mode collapse. Finally, the reconstructed 3D structure and the target 3D structure are visualized to ascertain their similar morphologies. The morphological parameter indicators of the reconstructed 3D structure were consistent with those of the target 3D structure. The microstructure parameters of the 3D structure were also compared and analyzed. The proposed method can achieve accurate and stable 3D reconstruction compared with classical stochastic methods of image reconstruction.

DOI: [10.1103/PhysRevE.107.055309](https://doi.org/10.1103/PhysRevE.107.055309)

I. INTRODUCTION

The microstructure of porous media and the physical properties of solids and fluids determine several macroscopic properties of the media [1–3]. These properties include petrophysical properties and transport properties in the oil reservoir [4–6], such as pore structures, permeability, electrical conductivity, wettability, capillary pressure, and relative permeability. In many fields of technology [7,8], understanding and simulating the internal structure of porous media is of great theoretical and practical significance. In recent years, three-dimensional (3D) imaging methods, such as x-ray computed tomography (CT) [9–11] and focused ion beam scanning electron microscope [12] have made rapid progress, which allow us to directly obtain 3D images of rocks ranging from the micrometer to nanometer scale. However, two-dimensional (2D) imaging devices have advantages of low cost, fast scanning speed, and higher resolution. Therefore, for the analysis of 3D microstructures, it is important to reconstruct statistically equivalent 3D images from the 2D images [13,14].

In recent decades, a variety of methods have been proposed and developed for reconstruction of porous media. The primary algorithms include optimization-based methods [13–26], multiple-point statistical method (MPS) [27], direct sampling (DS) [28,29], cross-correlation-based simulation [30–32], and deep learning-based methods [33–37]. It is generally known that the premise of these reconstruction methods is that 2D training images must satisfy the requirements of stationarity and ergodicity. It means that the 2D images must be able to statistically characterize the features of the entire 3D images [38,39]. Although starting from different perspectives, the core idea of these methods is the same, i.e., they extract morphological features (e.g., two-point correlation function or MPS function) from a given 2D training image or 3D image, and then they reproduce these features in the 3D reconstructed image.

A typical example of optimization-based methods is the simulated annealing (SA) method [13,14], which is a general and flexible method and has been used in the reconstruction of porous media. It can use any number of cost functions of any type as the objective function. To improve the initially developed SA algorithm, a multitude of scholars have successively proposed various optimization methods, including hierarchical simulated annealing [15–17], multithread

*ljiuanslz@163.com

†Corresponding author: qzteng@scu.edu.cn

SA algorithm [20,21], higher-order correlation functions [17,22,23], weighted correlation function [24–26], and co-occurrence correlation function [40].

The MPS method is another classical and commonly used reconstruction method. The original MPS [27] extracts local multiple-point features by scanning a training image with a specific template to obtain the probability of the center pixel conditioned on its neighborhood pixels. Owing to the use of multiple-point information, MPS effectively reproduces the morphology of porous media (e.g., geometry and connectivity) very well, especially for digital cores with complex geometries. The original MPS requires rescanning the training image using a template for each new data event, which makes the reconstruction process extremely time-consuming. To overcome this shortcoming, some special methods, such as single normal equation simulation [41], layer-by-layer reconstruction method [42], DS method [28,29], and MPS method accelerated based on deep learning (DL) [22] can be introduced.

With the emergence and rapid development of DL [43], original ideas and innovative methods have emerged. Owing to the powerful ability of DL to extract features from training images and to accelerate the speed of reconstruction based on GPU frameworks, the rapid development of DL brings another direction for the reconstruction of digital cores. Thus, by designing a reasonable network structure, the parameters of a network can be automatically learned according to the cost functions, which frees us from the complexity of manual feature designs. Another important advantage is that once a network has been trained, input images are reconstructed immediately.

Moreover, there have been several recent developments in deep generative models. All these approaches have shown satisfactory results in generating natural images because the deep hierarchical structure can capture the complex structure of the data; they can generate more realistic data than the traditional generative models. The most dominant generative models are variational autoencoders (VAEs) [44] and generative adversarial networks (GANs) [45]. VAEs pair the coding network with the decoding network or generative network, and their loss functions are expressed as mean squared errors; the generated images tend to be blurred. GANs are unsupervised generative methods with a powerful ability to learn data characteristics.

The two models of GANs are trained simultaneously: a generative model to synthesize the image and a discriminative model to distinguish the real image from the synthesized one. Although GANs can generate relatively clear images, their lack of convergence is problematic. The prominent conundrums of GANs are gradient disappearance, gradient explosion, and mode collapse [46], which make the training process relatively unstable. Currently, a lot of work has been devoted to enhance the quality of the generated images. The conditional GAN (CGAN) [47] improves the mapping problem by adding conditional probability to generators and discriminators. In addition to improving the quality of generated data, it is also necessary to match the generated data with the conditions in CGAN. The principle of deep convolutional GAN (DCGAN) [48] is basically the same as that of GAN, but it replaces the original multilayer perceptron of the generator and discriminator with a convolutional neural

network [49]. This method greatly enhances the quality of the generated images and accelerates the convergence speed. Wasserstein GAN (WGAN) [50] brings the Earth mover distance into measuring the minimum distance of each joint distribution between the real and the generated images, which theoretically improves the instability of GANs in the training process. Although WGAN has made better progress in terms of stable training, the generated samples are sometimes poor, owing to the use of the weight clipping strategy to force the discriminator to meet the judgment. Therefore, Wasserstein GAN with gradient penalty (WGAN-GP) [51] is a strategy that uses Wasserstein distance as the loss function and gradient penalty as the Lipschitz constraint so the discriminator network in GAN does not perform the classification task but acts as a critic. The method makes the training of WGAN more stable and generates higher quality images. Therefore, GANs have a wide application prospect in the aspect of 3D core reconstruction.

Mosser *et al.* [52] applied a GAN to reconstruct the solid-void image of porous media with a size of 64^3 , which allowed implicitly characterizing the probability distribution of 3D image data sets. Subsequently, they extended this method to generate oolitic Ketton limestone micro-CT unsegmented gray-scale images [53]. The results showed that GANs can effectively and accurately generate 3D images of porous media with statistical properties and morphological characteristics similar to those of the training images. Shams *et al.* [54] combined CGAN with a statistical methodology to generate homogeneous and heterogeneous 3D porous media from a 2D image. The reconstruction time was accelerated to nearly 1000-fold using CGANs compared with traditional statistical reconstruction methods. Feng *et al.* [55] proposed an end-to-end 3D reconstruction framework using Bicycle-GAN, which can achieve the reconstruction of statistically equivalent 3D structures from a 2D image. They validated the reconstruction results on two isotropic media and a nonstationary porous medium. The method achieved fast, accurate, and stable reconstruction. Zhang *et al.* [56] proposed a hybrid reconstruction model from a 2D image to a 3D porous media. This method combined the VAE and GAN. Meanwhile, Zhang *et al.* [57] combined the advantages of GAN and VAE to reconstruct 3D digital cores. To generate diverse 3D images and alleviate mode collapse of GANs, determinantal point process was used for feature extraction by the discriminator. Zha *et al.* [58] applied Wasserstein GANs with gradient penalty (WGAN-GP) to reconstruct 2D shale images. The training time was much shorter when this method was used to avoid the gradient disappearance of GANs. Valsecchi *et al.* [59] introduced an algorithm for reconstructing 3D porous media from 2D images by utilizing GAN. The discriminator of the model evaluated the three sections of a 3D image with existing 2D images. Volkhonskiy *et al.* [60] also proposed a depth generative model with encoder, generator, and discriminator modules to recover the entire 3D structure based on given central 2D slices. Zhang *et al.* [61] presented a 3D reconstruction algorithm of porous media using an improved WGAN-GP with 3D images as training data sets, which greatly improved the efficiency and image quality of 3D reconstruction. Cao *et al.* [62] designed a combined model of InfoGAN and style-based GAN led by prior information (CISGAN) to generate more

manageable and reasonable large-size digital rocks based on small sample sets. The model utilized the Wasserstein distance as the loss function and the gradient penalty as the constraint to resolve the problem of gradient instability. Li *et al.* [63] applied WGAN-GP to reconstruct rocks of Berea sandstone and Ketton limestone from 2D to 2D images and extended the size from 64×64 pixels to 128×128 pixels. However, the training data in the above 3D reconstruction were 3D images.

In this paper, we take advantage of homogeneous and isotropic 2D digital core images to automatically extract the statistical features of 2D images through feature learning, and we reconstruct 3D images with similar statistical features based on a generative adversarial approach. Therefore, this paper introduces an improved method that combines the encoder of VAE and WGAN-GP to generate 3D digital cores from 2D core images, and is named EWGAN-GP. The model consists of an encoder, a generator, and three discriminators. We collapse the VAE decoder and WGAN generator into one by letting them share parameters and training them jointly. The VAE encoder is used to extract the statistical features of a 2D image, and the generator extends these features to the 3D image data. While estimating the features of the 3D image, three discriminators are applied to judge the reconstruction effect of each slice along three orthogonal directions (x , y , z). The porosity loss function is utilized to constrain structures of 3D images as a whole. In the optimization processes of the encoder, generator, and discriminators, the strategy based on WGAN-GP makes the training process converge faster and the reconstruction results more stable. The proposed method circumvents the problems of reconstruction, such as gradient disappearance, mode collapse, and gradient explosion. To demonstrate the generative capability of the model on different cores, we conducted tests on five kinds of core images. In addition, we visualized and quantitatively compared the generated 3D image with the target image.

The rest of this paper is organized as follows. Section II elaborates the designed core reconstruction method and framework. The assessment methods for reconstruction are given in Sec. III. Section IV presents the reconstructed results and analysis. In Sec. V, we make a summary.

II. METHODOLOGY

Here, we first introduce the motivation of our proposed model. Next, the principles of VAE and GAN are elaborated. Second, we describe the framework for reconstructing 3D digital cores from 2D images, including the main principles and network structure, as well as the loss function. Finally, we develop the training algorithm of this model.

A. Motivation

Cores from different reservoirs in various regions differ significantly in morphology and also in quantity, which creates the following problems. First, it is difficult to find adequate 3D images for training, such as for sandstones, carbonates, shales, and fractures. Second, the core material of some old reservoirs changes with time, but the bulk of stored core 2D images will be preserved. Therefore, since a large number of 2D images have been accumulated, it is of great

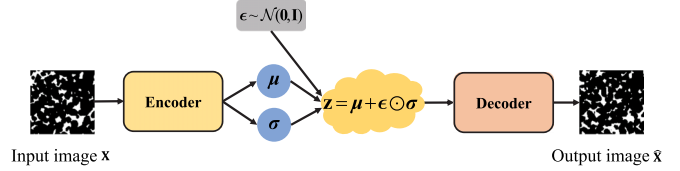


FIG. 1. Architecture of VAE.

research significance and application value to make full use of the 2D data of these cores.

B. Basic principles of VAE and GAN

1. Introduction to VAE

A VAE is a generative model that combines the characteristics of DL and statistical learning. It exploits the powerful ability of deep models in fitting nonlinearities. It is often used for feature extraction and dimensionality reduction of data. A VAE is composed of an encoder and a decoder, and its structure is shown in Fig. 1. The elementwise product is denoted by \odot . The input image \mathbf{x} is encoded into a latent vector \mathbf{z} , then the decoder gets the output image $\hat{\mathbf{x}}$ by decoding the resultant latent vector \mathbf{z} .

The encoder forecasts the probability distribution function $q_\phi(\mathbf{z} | \mathbf{x})$ of the input image \mathbf{x} , where ϕ are the variational parameters. The decoder forecasts the probability distribution function $p_\theta(\hat{\mathbf{x}} | \mathbf{z})$ of the output image $\hat{\mathbf{x}}$, where θ are the generative model parameters. The definitions of $q_\phi(\mathbf{z} | \mathbf{x})$ and $p_\theta(\hat{\mathbf{x}} | \mathbf{z})$ are as follows:

$$\mathbf{z} \sim \text{encoder}(\mathbf{x}) = q_\phi(\mathbf{z} | \mathbf{x}), \quad (1a)$$

$$\hat{\mathbf{x}} \sim \text{decoder}(\mathbf{z}) = p_\theta(\hat{\mathbf{x}} | \mathbf{z}), \quad (1b)$$

where $\text{encoder}(\cdot)$ and $\text{decoder}(\cdot)$ represent the encoding and the decoding functions, respectively.

The definition of the loss function in VAE [44] is defined as

$$\mathcal{L}_{\text{VAE}} = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\hat{\mathbf{x}} | \mathbf{z})] - D_{\text{KL}}(q_\phi(\mathbf{z} | \mathbf{x}) \| p_\theta(\mathbf{z})), \quad (2)$$

where the prior probabilities are the multivariate Gaussian normal distribution model, i.e., $p_\theta(\mathbf{z}) \sim \mathcal{N}(\mathbf{z}; 0, I)$. The first term in the loss function of VAE is the reconstruction error, which aims to make the generated image less different from the input image. The second term is a regularizer, which makes the distribution returned by the encoder close to the standard normal distribution, also known as the Kullback-Leibler (KL) divergence.

2. Introduction to GAN

In 2014, Goodfellow *et al.* [45] proposed GANs, which can reproducibly and creatively generate a large number of images that meet the needs of image generation. They have been applied in many areas, including image inpainting [64], super-resolution [65,66], and style transfer [67]. The basic structure of GAN is shown in Fig. 2.

The original GANs include two models: a generator and discriminator. The input of the generator is a normally distributed random vector \mathbf{Z} , and the output is the generated fake

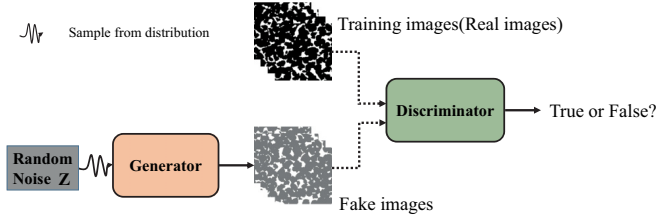


FIG. 2. Architecture of GAN.

image. The data distribution of the generated fake image is denoted by $p_G(\mathbf{x}; \theta)$, where the parameters θ are determined by the network parameters. The distribution of the real data is $p_{\text{data}}(\mathbf{x})$. The ultimate goal of adversarial training is to find the θ that makes the distribution $p_G(\mathbf{x}; \theta)$ approximate to the distribution $p_{\text{data}}(\mathbf{x})$. The discriminator determines whether the input image is from real data or from a fake one, and then outputs a scalar value of 0 or 1. The purpose of training a discriminator is to assign the correct labels to the training image and generated one. The aim of training a generator is to make the generated image close to the real one so the discriminator cannot estimate whether the input is real or not. The definition of loss function for GANs is a minimization-maximization problem described as

$$\min_G \max_D V(D, G) = \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})} [\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z} \sim p_z(\mathbf{z})} [\log(1 - D(G(\mathbf{z})))] \quad (3)$$

The training of GANs is a two-step process. In the first step, the parameters of the generator are fixed and the discriminator is trained to maximize

$$\mathcal{L}_D = \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})} [\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z} \sim p_z(\mathbf{z})} [\log(1 - D(G(\mathbf{z})))] \quad (4)$$

In the second step, the parameters of the discriminator are fixed. The "fake" images $G(\mathbf{z})$ are generated by the generator from the hidden latent space \mathbf{Z} . The latent space is comprised of independent normally or uniformly distributed random variables. The generator is trained to minimize

$$\mathcal{L}_G = \mathbb{E}_{\mathbf{z} \sim p_z(\mathbf{z})} [\log(1 - D(G(\mathbf{z})))] \quad (5)$$

In general, we do not minimize the Generator but maximize it.

The solutions to this optimization problem have been proved to be a Nash equilibrium in which each participant reaches a local minimum of the designed loss function. The key problem of GANs is how to train the generator and discriminator. The fake data produced by the generator can fool the discriminator with falsehoods, but the discriminator can also judge the fake data produced by the generator. GANs exhibit a high degree of instability during training and require a lot of debugging to find a set of hyperparameters that achieve stable training. Many researchers have published heuristic methods that have been fulfilled to stabilize GANs training, such as one-sided label smoothing or adding white noise to the input layer of the networks.

C. Structure of EWGAN-GP

The reconstruction of a 3D image from a 2D image is essentially a problem of dimensional expansion. We know that the 2D image is one of the 3D images, and a 2D image should correspond to a number of 3D structures. Therefore, we reconstruct 3D structures with similar features based on the statistical features of 2D images. The first requirement of the reconstruction is to accurately extract the required statistical features from 2D images using DL methods. The second one is to reproduce the statistical features of 2D images into 3D structures using dimensional expansion methods. These two requirements can be naturally associated with the encoder model for extracting image features and the GAN model for generating images. However, the GANs mentioned above have a lot of disadvantages, such as training instability and model collapse. If two models are simply concatenated, they do not satisfy the design requirements. Below we introduce our method in detail.

According to 2D-to-3D reconstruction requirement, we put forward the EWGAN-GP network frameworks shown in Fig. 3. The symbol \oplus represents the addition of the corresponding elements of vectors. Our proposed method consists of three parts: (i) an encoder E , (ii) a generator G , and (iii) three discriminators D_{xy} , D_{yz} , and D_{zx} . The main function of the encoder E is to extract the statistical features of a 2D image. The encoder E learns the mapping between the input image \mathbf{x}_{2D} and the latent vector space \mathbf{Z}_{enc} . The mapping is denoted as the distribution $p(\mathbf{z}_{\text{enc}}|\mathbf{x}_{2D})$. The generator G generates a 3D structure $\tilde{\mathbf{x}}_G$ by feeding the combination of the sampled latent vector \mathbf{z}_{enc} and the Gaussian noise $\mathbf{z}_{\text{noise}}$ to the generator G . For homogeneous and isotropic cores, all of their cross sections along three orthogonal directions (x , y , z) have similar morphological and statistical properties to the 2D images. Therefore, to compare the similarity between the generated 3D structure with the 2D image, we design three discriminators to judge the discrepancy between all cross sections of the generated 3D structure along the three orthogonal directions (x , y , z) and the input 2D image. To constrain the pore properties of 3D structures as a whole, we propose a porosity loss function, and we will elaborate on the networks of the EWGAN-GP, loss functions, and the training algorithm.

1. Architecture of encoder

The network structure of the encoder is shown in Fig. 4. The encoder removes the redundant information in the 2D image so as to achieve the purpose of dimensionality reduction. For the input 2D image, the encoder extracts the statistical properties of the image in terms of the mean $\boldsymbol{\mu}$ and the standard deviation $\boldsymbol{\sigma}$, and then randomly samples to obtain the actual encoding \mathbf{z}_{enc} . The encoder includes six downsampling layers, each consisting of a 2D convolution (Conv2D) function, a leaky rectified linear unit (LeakyReLU) activation function, and a batch normalization (BatchNorm2D) function. The start and end downsampling layers do not incorporate the BatchNorm2D function. The Conv2D is a 2D convolution function that uses a convolution kernel to traverse the input image to extract the features of the data. We choose LeakyReLU as the activation function, so the information will not be lost when the input information is less

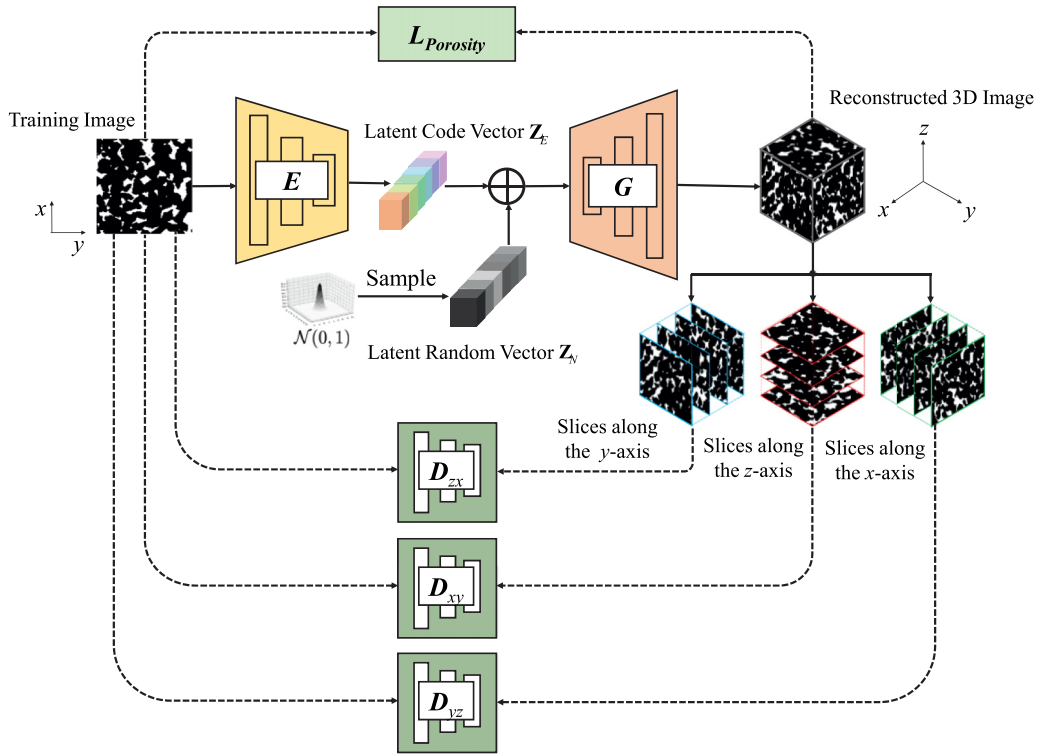


FIG. 3. Architecture of EWGAN-GP.

than 0. The LeakyReLU activation function gives a negative slope for the negative part, which alleviates the problem of gradient disappearance and expands the range of function representation. Assuming that x denotes the data computed by the network, the mathematical formula of the activation function is

$$\text{LeakyReLU}(x) = \begin{cases} x, & x \geq 0 \\ \alpha x, & x < 0, \alpha \in (0, 1), \end{cases} \quad (6)$$

where α controls the gradient of the negative part of the linear function. The role of the BatchNorm2D function is to ensure that the input of each layer of the neural network keeps the same distribution during the training. Meanwhile, it can alleviate the difficulty of training the neural network and speed up the learning process. The main parameters of the encoder are listed in Table I.

2. Architecture of generator

The function of the generator is to produce the corresponding 3D structure from the extracted 2D image features (represented by latent vectors). The network structure of the

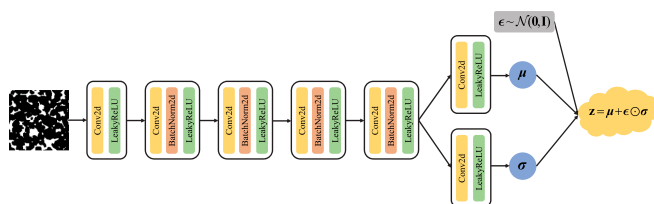


FIG. 4. Architecture of encoder.

generator is given in Fig. 5. The generator is composed of six 3D transposed convolution (ConvTranspose3D) functions, which extend the dimension from two dimensions to three dimensions. After each 3D transposed convolution operation, a batch normalization (BatchNorm3D) function is added. Considering the complexity of the network, we choose the ReLU activation function without worrying about information loss. To make the training more effective and to increase the diversity of the output, we add Gaussian noise to the input layer. For the last layer of the network, we only adopt the ConvTranspose3D function and the hyperbolic tangent function Tanh, so the real values of our final output are limited between -1 to 1 . The activation function Tanh is an s-shaped saturated activation function. If x denotes the data computed by the neural network, then

$$\text{Tanh}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \quad (7)$$

Therefore, as the last layer of the generator, the activation function Tanh normalizes the calculation results into $(-1, 1)$. These functions are more controllable than ReLU functions. The main parameters of the generator are given in Table II.

3. Architecture of discriminators

The generated 3D structure has similar statistical and morphological features to the 2D image. It means the three orthogonal sections of the 3D structure also have similar morphological features to the 2D image. The designed discriminators are used to compare each of the three orthogonal sections of the 3D structure with the 2D image.

TABLE I. Main parameters of encoder.

Layer	Function	Filers	Kernel	Stride	Padding	Batch Normal	Activation
1	Conv2D	64	4	2	1	No	LeakyReLU
2	Conv2D	128	4	2	1	Yes	LeakyReLU
3	Conv2D	256	4	2	1	Yes	LeakyReLU
4	Conv2D	512	4	2	1	Yes	LeakyReLU
5	Conv2D	1024	4	2	1	Yes	LeakyReLU
6	Conv2D	128	4	1	0	No	No

Accordingly, we construct three discriminators D_{xy} , D_{yz} , D_{zx} , which, respectively, assess the difference between the 2D training image and the slices of the 3D structure along three orthogonal directions. The network structure of the three discriminators is the same because all three slices of the 3D structure are compared with the 2D training image. The network structure of the corresponding discriminator is provided in Fig. 6. For the generated cube volume with edge length l pixels, a total of $3l$ 2D slices can be obtained by slicing one pixel along the x , y , and z directions. For each generated 3D structure, the three discriminators can compare each slice with the correspondent 2D training image. Because the objective of the discriminators is to differentiate the real images from the slices, the core of the network is implemented by a series of 2D convolution (Conv2D) functions. After each convolution operation, a nonlinear activation function ReLU is appended. The discriminators not only need to meet Lipschitz continuity [68], but also their gradient must be limited to a certain numerical range. Here we apply gradient penalty to each sample in each batch. We do not use batch normalization functions here since they may break the Lipschitz continuity of the discriminator. The main parameters of the discriminator are given in Table III.

D. Loss functions

The loss functions are an essential part of DL. Well-designed loss functions help the network to converge fast and get good generalizability results. The loss function of VAE in EWGAN-GP is defined as

$$\mathcal{L}_{\text{VAE}} = \mathcal{L}_{\text{like}}^{\text{pixel}} + \mathcal{L}_{\text{prior}}. \quad (8)$$

The loss function $\mathcal{L}_{\text{like}}^{\text{pixel}}$ of VAE describes the difference between the generated image and the input image in each layer; $\mathcal{L}_{\text{prior}}$ is the KL divergence, which makes the distribution returned by the encoder approximate the standard normal distribution.

According to the literature of GANs [68], when the discriminator reaches optimality, the optimized cost function is

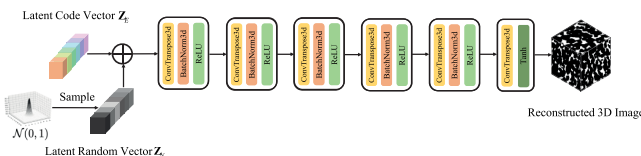


FIG. 5. Architecture of generator.

expressed by JS divergence, i.e.,

$$\mathcal{L}^* = 2D_{\text{JS}}(p_{\text{data}} \parallel p_G) - 2 \lg 2, \quad (9)$$

where p_{data} denotes the distribution of the real data, and p_G the distribution of the generated data. Jensen-Shannon divergence (JS divergence) is a non-symmetric measurement of similarity between two distributions, which is defined as

$$D_{\text{JS}}(p_{\text{data}} \parallel p_G) = \frac{1}{2} \text{KL} \left(p_{\text{data}} \parallel \frac{p_{\text{data}} + p_G}{2} \right) + \frac{1}{2} \text{KL} \left(p_G \parallel \frac{p_{\text{data}} + p_G}{2} \right),$$

where KL is the Kullback-Leibler divergence (KL divergence) defined by the following equation

$$\text{KL}(p_{\text{data}} \parallel p_G) = \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})} \left[\log \left(\frac{p_{\text{data}}}{p_G} \right) \right].$$

Under the condition of optimal discriminator, the loss function \mathcal{L}_G of the generator is equated to minimize the JS divergence between the distribution p_{data} of the real data and the distribution p_G of the generated data. However, due to the small overlap between p_{data} and p_G in the high-dimensional space, it is difficult to let p_G approximate to p_{data} by optimizing JS divergence. This will lead to the gradient disappearance and training instability of GANs.

The WGAN provides an efficient distance measure between real and generated data by minimizing the Wasserstein distance $W(p_G, p_{\text{data}})$. Compared with the JS divergence, the advantage of this distance is that it reflects the proximity of the two distributions even if there is no overlap between them. This distance can be considered as the minimum cost of transferring the generated data distribution p_G to the real data distribution p_{data} . Ideally, $W(p_G, p_{\text{data}})$ is always continuous and differentiable, which can be expressed as

$$W(p_G, p_{\text{data}}) = \frac{1}{K} \sup_{\|f\|_L \leq K} \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})} [f(\mathbf{x})] - \mathbb{E}_{\tilde{\mathbf{x}} \sim p_G(\tilde{\mathbf{x}})} [f(\tilde{\mathbf{x}})], \quad (10)$$

where the supremum is over all the K -Lipschitz functions f , and $\tilde{\mathbf{x}}$ represents the generated data, which is equal to $G(\mathbf{z})$. The function f can be fitted with a neural network and K exists by limiting all parameters in the neural network to a certain range.

The Lipschitz constraint is implemented by directly controlling the gradient norm of the discriminator's output. To simplify this constraint and improve the training speed, we

TABLE II. Main parameters of generator.

Layer	Function	Filers	Kernel	Stride	Padding	Batch Normal	Activation
1	ConvTranspose3D	512	4	2	0	Yes	ReLU
2	ConvTranspose3D	256	4	2	1	Yes	ReLU
3	ConvTranspose3D	128	4	2	1	Yes	ReLU
4	ConvTranspose3D	64	4	2	1	Yes	ReLU
5	ConvTranspose3D	32	4	2	1	Yes	ReLU
6	ConvTranspose3D	1	4	1	1	No	Tanh

think of directly controlling the gradient norm of the discriminator's output regarding its input. Let $p_{\hat{\mathbf{x}}}$ be the distribution of $\hat{\mathbf{x}}$, where $\hat{\mathbf{x}}$ is the uniform sampling on the line between real data and generated data. The loss function of the discriminator in WGAN-GP is obtained as

$$\mathcal{L}_D = \mathbb{E}_{\tilde{\mathbf{x}} \sim p_G} [D(\tilde{\mathbf{x}})] - \mathbb{E}_{\mathbf{x} \sim p_r} [D(\mathbf{x})] + \lambda \mathbb{E}_{\hat{\mathbf{x}} \sim p_{\hat{\mathbf{x}}}} [(\|\nabla_{\hat{\mathbf{x}}} D(\hat{\mathbf{x}})\|_2 - 1)^2], \quad (11)$$

where λ is a constant (generally given the value 10).

The loss function of the generator in WGAN-GP is

$$\mathcal{L}_G = -\mathbb{E}_{\tilde{\mathbf{x}} \sim p_G} [D(\tilde{\mathbf{x}})]. \quad (12)$$

During training, the stochastic Adam optimizes the parameters of the discriminator and generator by iteratively updating \mathcal{L}_D and \mathcal{L}_G , respectively.

In this method, Wasserstein distance with gradient penalty is taken here as a measure of the similarity between the generated and the real images. The reconstruction loss function \mathcal{L}_{rec} of the generator G and slice loss functions of the three discriminators $\mathcal{L}_{\text{dis}}(D_{xy}, D_{yz}, D_{zx})$ are given below.

The loss function \mathcal{L}_{rec} of the generator G is defined as

$$\mathcal{L}_{\text{rec}}(G) = -\mathbb{E}[D_{xy}((G(E(\mathbf{x}), \mathbf{z}_{\text{noise}})) \cdot S_{xy})] - \mathbb{E}[D_{yz}((G(E(\mathbf{x}), \mathbf{z}_{\text{noise}})) \cdot S_{yz})] - \mathbb{E}[D_{zx}((G(E(\mathbf{x}), \mathbf{z}_{\text{noise}})) \cdot S_{zx})]. \quad (13)$$

The generated 3D structure is sent to the discriminators. This function is used to compute the reconstruction loss along three orthogonal directions, where the symbol \cdot indicates the slicing operation along the x , y , and z directions.

The loss function $\mathcal{L}_{\text{dis}}(D_{xy}, D_{yz}, D_{zx})$ of the three discriminators D_{xy} , D_{yz} , and D_{zx} is defined as

$$\mathcal{L}_{\text{dis}}(D_{xy}, D_{yz}, D_{zx}) = -\mathcal{L}'_{\text{dis}} + \mathcal{L}''_{\text{dis}} + \lambda \mathcal{L}'''_{\text{dis}}, \quad (14)$$

where λ is set as 10.

(1) The loss functions calculated by the three discriminators for training images are

$$\mathcal{L}'_{\text{dis}} = \mathbb{E}[D_{xy}(\mathbf{x})] + \mathbb{E}[D_{yz}(\mathbf{x})] + \mathbb{E}[D_{zx}(\mathbf{x})]. \quad (15)$$

(2) Given a training image \mathbf{x}_r , the corresponding generated 3D structure is $\mathbf{x}_G \sim G(E(\mathbf{x}_r), \mathbf{z}_{\text{noise}})$. The loss functions calculated by the three discriminators for the slices along three orthogonal directions of the generated 3D structure \mathbf{x}_G are

$$\mathcal{L}''_{\text{dis}} = \mathbb{E}[D_{xy}(\mathbf{x}_G \cdot S_{xy})] + \mathbb{E}[D_{yz}(\mathbf{x}_G \cdot S_{yz})] + \mathbb{E}[D_{zx}(\mathbf{x}_G \cdot S_{zx})]. \quad (16)$$

(3) Given a training image \mathbf{x}_r , the corresponding generated 3D structure is $\mathbf{x}_G \sim G(E(\mathbf{x}_r), \mathbf{z}_{\text{noise}})$. Let $\epsilon_{xy}, \epsilon_{yz}, \epsilon_{zx} \in U[0, 1]$ be randomly selected numbers, then randomly selected samples $\hat{\mathbf{x}}_{yz}, \hat{\mathbf{x}}_{zx}, \hat{\mathbf{x}}_{xy}$ along the x , y , and z directions, respectively, are expressed as

$$\begin{aligned} \hat{\mathbf{x}}_{yz} &= \epsilon_{yz} \mathbf{x}_r + (1 - \epsilon_{yz})(\mathbf{x}_G \cdot S_{yz}), \\ \hat{\mathbf{x}}_{zx} &= \epsilon_{zx} \mathbf{x}_r + (1 - \epsilon_{zx})(\mathbf{x}_G \cdot S_{zx}), \\ \hat{\mathbf{x}}_{xy} &= \epsilon_{xy} \mathbf{x}_r + (1 - \epsilon_{xy})(\mathbf{x}_G \cdot S_{xy}). \end{aligned} \quad (17)$$

The evaluation of the discriminators for the selected samples can be calculated using the formula

$$\begin{aligned} \mathcal{L}'''_{\text{Dis}} &= \mathbb{E}[(\|\nabla_{\hat{\mathbf{x}}_{xy}} D_{xy}(\hat{\mathbf{x}}_{xy})\|_2 - 1)^2] \\ &+ \mathbb{E}[(\|\nabla_{\hat{\mathbf{x}}_{yz}} D_{yz}(\hat{\mathbf{x}}_{yz})\|_2 - 1)^2] \\ &+ \mathbb{E}[(\|\nabla_{\hat{\mathbf{x}}_{zx}} D_{zx}(\hat{\mathbf{x}}_{zx})\|_2 - 1)^2]. \end{aligned} \quad (18)$$

For the generated 3D structure, we implement constraints on the statistical and morphological features in three directional slices. In order to better evaluate the 3D structure as a whole, we compare the porosity between the 2D and 3D images, so that the porosity of the generated 3D structure better approximates the porosity of the training image. The proposed porosity loss function is defined as

$$\mathcal{L}_{\text{porosity}} = \|(G(E(\mathbf{x}), \mathbf{z}_{\text{noise}}))_{\text{porosity}} - \mathbf{x}_{\text{porosity}}\|_2^2 \quad (19)$$

Therefore, the loss functions of the whole network are

$$\begin{aligned} \mathcal{L}_{\text{total}}(E, G, D) &= \mathcal{L}_{\text{VAE}} + \mathcal{L}_{\text{dis}}(D_{xy}, D_{yz}, D_{zx}) \\ &+ \mathcal{L}_{\text{rec}}(G) + \lambda_{\text{porosity}} \mathcal{L}_{\text{porosity}}, \end{aligned} \quad (20)$$

where $\lambda_{\text{porosity}}$ is the weighting factor of the porosity loss function.

The complete training procedure about EWGAN-GP structure is described in Algorithm 1.

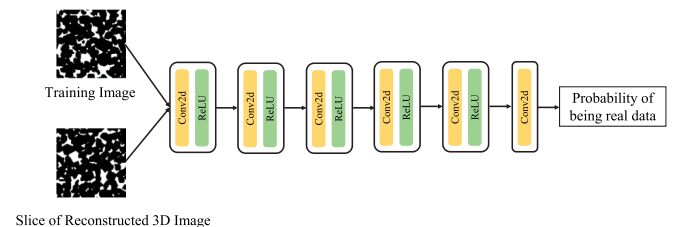


FIG. 6. Architecture of discriminator.

TABLE III. Main parameters of discriminator.

Layer	Function	Filers	Kernel	Stride	Padding	Activation
1	Conv2D	32	4	2	1	ReLU
2	Conv2D	64	4	2	1	ReLU
3	Conv2D	128	4	2	1	ReLU
4	Conv2D	256	4	2	1	ReLU
5	Conv2D	512	4	2	1	ReLU
6	Conv2D	1	4	1	0	No

III. RECONSTRUCTION EVALUATION CRITERIA

In this paper, to validate the reconstruction performance of the proposed method, we compared statistical feature functions and morphological descriptors, including two-point correlation function, lineal-path function, two-point cluster function and local porosity distribution (LPD) function, as shown in Fig. 7. Before introducing the experimental results, we make a brief presentation of the morphological descriptors.

A. Morphological descriptors

Regarding correlation functions, we primarily adopted Torquato's definition [1]. Here, we are mainly concerned with two-phase cores. In the n -dimensional Euclidean space \mathbb{R}^n ($n = 1, 2, 3$), the indicator function of position \mathbf{u} is defined as

$$I^{(i)}(\mathbf{u}) = \begin{cases} 1, & \mathbf{u} \in V_i \\ 0, & \mathbf{u} \in \bar{V}_i, \end{cases} \quad (21)$$

where $V_i \in \mathbb{R}^n$ is the region covered by phase i (equal to 1 or 2) and $\bar{V}_i \in \mathbb{R}^n$ is the region covered by the other phase.

Figure 7 depicts the definitions of the two-point correlation function, the lineal-path function, and the two-point cluster function. The two-point correlation function $S_2(\mathbf{u}_1, \mathbf{u}_2)$ [1] represents the correlation between two randomly selected points \mathbf{u}_1 and \mathbf{u}_2 in spatial distribution. The probability that two points are located to the same phase is defined as

$$S_2(\mathbf{u}_1, \mathbf{u}_2) = \langle I(\mathbf{u}_1)I(\mathbf{u}_2) \rangle, \quad (22)$$

where angular brackets $\langle \cdot \rangle$ express an ensemble average.

For homogeneous and isotropic cores, the two-point correlation function is only related to the distance $r = |\mathbf{u}_1 - \mathbf{u}_2|$ of these two points. This indicates that in a homogeneous core, the distribution of the two-point correlation function in the three orthogonal directions (x, y, z) is relatively close. Therefore, the two-point correlation function of a 2D image can be used to approximate the two-point correlation function in 3D space. For convenience, the two-point correlation function is briefly denoted by $S_2(r)$. Usually, r is generally elided and expressed as S_2 . When $r=0$, the two-point correlation function is the porosity.

The lineal-path function [69] is also denoted as a multiple-point connectivity probability, which portrays the connectivity of a phase in the image. The function $L(\mathbf{r})$ represents the probability that a line segment \mathbf{r} of length $r = |\mathbf{r}|$ is

Algorithm 1 EWGAN-GP algorithm for the reconstructions of isotropic cores.

Require: θ_E, θ_G , and θ_D , the trainable parameters for E, G , and D respectively; the number of discriminator iterations per generator iteration n_D ; the batch size m ; volume edge length l ; the gradient penalty coefficient λ ; the hyperparameter of porosity loss $\lambda_{\text{porosity}}$; and Adam optimizer hyperparameters α, β_1, β_2 .

```

1: while  $\theta_G$  has not converged do
2:   Discriminators training:
3:   for  $t = 0, \dots, n_D$  do
4:     for  $i = 0, \dots, m$  do
5:       Sample  $\{\mathbf{x}_r\} \sim p_r(\mathbf{x}_r)$  a batch of the real data
6:       Sample  $\mathbf{z} \sim p(\mathbf{z})$  a batch of random noise
7:        $\mathbf{x}_f \leftarrow G_{\theta_G}(\mathbf{x}_r, \mathbf{z})$  generate a 3D volume
8:       for  $a = 1, 2, 3$  do
9:         for  $d = 1, \dots, l$  do
10:           $\mathbf{x}_{f-a} \leftarrow$  2D slice of  $\mathbf{x}_f$  at depth  $d$  along axis  $a$ 
11:          Sample a random number  $\epsilon \sim U[0, 1]$ 
12:           $\mathbf{x}_{\text{random}} \leftarrow \epsilon \mathbf{x}_{f-a} + (1 - \epsilon) \mathbf{x}_r$ 
13:           $\mathcal{L}_D^{(a,d)} \leftarrow D_{\theta_D}(\mathbf{x}_{f-a}) - D_{\theta_D}(\mathbf{x}_r)$ 
14:             $+ \lambda (\|\nabla_{\mathbf{x}_{\text{random}}} D_{\theta_D}(\mathbf{x}_{\text{random}})\|_2 - 1)^2$ 
15:          end for
16:           $\theta_D \leftarrow \text{Adam}(\nabla_{\theta_D} \frac{1}{m} \sum_{a=1}^3 \sum_{d=1}^l \mathcal{L}_D^{(a,d)})$ 
17:        end for
18:      end for
19:    Generator and encoder training:
20:    for  $i = 0, \dots, m$  do
21:      Sample  $\{\mathbf{x}_r\} \sim p_r(\mathbf{x}_r)$  a batch of the real data
22:       $\mathbf{z}_E \leftarrow E(\mathbf{x}_r)$ 
23:      Sample  $\mathbf{z} \sim p(\mathbf{z})$  a batch of random noise
24:       $\mathbf{x}_f \leftarrow G_{\theta_G}(\mathbf{x}_r, \mathbf{z})$  generate a 3D volume
25:       $\mathcal{L}_{\text{KL}} \leftarrow D_{\text{KL}}(q(\mathbf{z}_E|\mathbf{x}_r) \parallel p(\mathbf{z}_E))$ 
26:       $\mathcal{L}_{\text{like}}^D \leftarrow -\mathbb{E}_{q(\mathbf{z}_E|\mathbf{x}_r)}[\log p_{\theta_G}(\mathbf{x}_f|\mathbf{z}_E)]$ 
27:       $\theta_E \leftarrow \text{Adam}(\nabla_{\theta_E} \frac{1}{m} (\mathcal{L}_{\text{KL}} + \mathcal{L}_{\text{like}}^D))$ 
28:      for  $a = 1, 2, 3$  do
29:        for  $d = 1, \dots, l$  do
30:           $\mathbf{x}_{f-a} \leftarrow$  2D slice of  $\mathbf{x}_f$  at depth  $d$  along axis  $a$ 
31:           $\mathcal{L}_G^{(a,d)} \leftarrow -D_{\theta_G}(\mathbf{x}_{f-a})$ 
32:           $\mathcal{L}_{\text{porosity}} \leftarrow \|\mathbf{x}_{f_{\text{porosity}}} - \mathbf{x}_{r_{\text{porosity}}}\|_2^2$ 
33:        end for
34:      end for
35:       $\theta_G \leftarrow \text{Adam}(\nabla_{\theta_G} \frac{1}{m} \sum_{a=1}^3 \sum_{d=1}^l (\mathcal{L}_G^{(a,d)} + \lambda_{\text{porosity}} \mathcal{L}_{\text{porosity}}))$ 
36:    end for
37:  end while

```

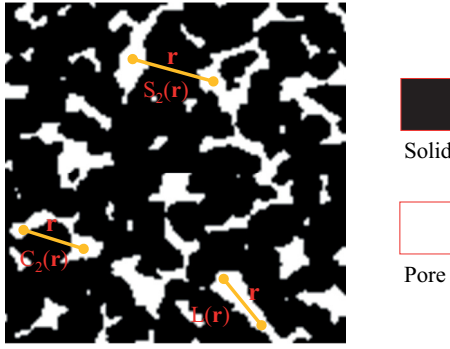



FIG. 7. Three morphological descriptors.

completely contained in the same phase. Usually, we calculate the lineal-path function of the 3D structure in three orthogonal directions. In an isotropic homogeneous core, the lineal-path function is only related to length r of the line segment \mathbf{r} , so it can be expressed as $L(r)$ or L . For the pore phase, the lineal-path function can be defined as follows:

$$L(\mathbf{r}) = \text{Prob}\{I(\mathbf{u}) = 1, I(\mathbf{u} + 1) = 1, \dots, I(\mathbf{u} + \mathbf{r}) = 1\}. \tag{23}$$

When \mathbf{u} is the pore phase, then $I(\mathbf{u}) = 1$; the other phase, then $I(\mathbf{u}) = 0$. It is obvious that the porosity is $\phi = L(0)$.

The two-point cluster function C_2 [70] is another important function that depicts the connectivity of a core. The cluster is an independent connected region in an image. The two-point cluster function C_2 expresses the probability that two points are in the same cluster. From definitions, C_2 represents higher-order connectivity information than S_2 .

B. Local porosity distribution

LPD [71,72] is also a generally used method to represent the geometry of cores. The porosity of each local location is measured by a measurement unit that traverses the 2D or 3D image. The local porosity is defined as

$$\phi(\vec{\mathbf{r}}, L) = \frac{V[P \cap K(\vec{\mathbf{r}}, L)]}{V[K(\vec{\mathbf{r}}, L)]}, \tag{24}$$

where P is the pore phase in 2D or 3D image, $K(\vec{\mathbf{r}}, L)$ represents a measurement unit with an edge length L centered at the lattice vector $\vec{\mathbf{r}}$, and V denotes the area or the volume of the measurement unit. Thus, the LPD can be represented as

$$\mu(\phi, L) = \frac{1}{m} \sum \delta[\phi - \phi(\vec{\mathbf{r}}, L)], \tag{25}$$

where m is the sum of times the measurement unit $K(\vec{\mathbf{r}}, L)$ is traversed. $\delta(\cdot)$ is the Dirac function, defined as $\delta[\phi - \phi(\vec{\mathbf{r}}, L)] = \begin{cases} 1, & \phi = \phi(\vec{\mathbf{r}}, L) \\ 0, & \phi \neq \phi(\vec{\mathbf{r}}, L) \end{cases}$.

Therefore, $\mu(\phi, L)$ represents the probability distribution of the porosity range in $[\phi, \phi + d\phi]$ with a measurement unit of an edge length L .

The LPD function $\mu(\phi, L)$ can reflect the homogeneity of the core to some extent. The measurement results of local porosity are directly related to the size of the measurement unit. The curves of the LPD calculated by measurement units of different sizes are different. When size L of the measure-

ment unit is fixed, the function $\mu(\phi, L)$ reflects the overall porosity distribution of the core with porosity ϕ as the independent variable. The concentration of its distribution reflects the good or bad homogeneity of the 3D structure. The more concentrated the curve distribution is, the better the homogeneity is, and vice versa.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

A. Data set and parameter settings

In this paper, we mainly focused on the 3D reconstruction of two-phase core images. The training data sets used in our model were composed of 2D images, and they had no corresponding 3D structure.

During model training, the hyperparameter $\lambda_{\text{porosity}}$ is given in Eq. (20) and set to 500. The batch size and the number of noise channels added are set to 8 and 32, respectively. We used the Adam optimizer and set the learning rates to 0.0001 for the encoder, generator, and discriminators. We trained the discriminators twice before training the generator and the encoder once, iteratively updating their parameters.

B. Results and analyses

To test the effectiveness of the algorithm, we chose five types of homogeneous core images, including Bentheimer sandstones, Fontainebleau sandstones with two different porosities, Berea sandstones, and another type of Ketton limestones. The 2D images with their corresponding 3D structure were chosen as the test set, and the results are exhibited by visual and quantitative comparison. In addition, to validate the stability and precision of the proposed method, we performed ten reconstructions for each 2D image. At the same time, the results of ten reconstructions and the average values were compared with the target structure in the light of morphological description functions and LPD. Furthermore, the 3D structural parameters of the reconstructed structure and the target structure were also analyzed and evaluated. In our paper, for an isotropic core, three morphological descriptors were calculated along three orthogonal directions and then averaged. In our experiments, we primarily focused on 128^3 image reconstruction.

1. Bentheimer sandstone reconstruction

Bentheimer sandstones [73] are considered to be ideal laboratory core samples because they have lateral continuity and homogeneous block-scale nature. In addition, they have limited amount of minerals, a constant grain size distribution, porosity, permeability and dielectrical values, which make them suitable for experimental research and corresponding theoretical comparisons. Thus, Bentheimer sandstones were adopted to verify our method. The size of the selected Bentheimer sandstones was 128^3 with the resolution $3.004 \mu\text{m}$. The porosity of this sandstone was 0.2222. Further, to account for the reconstruction effect of our method, we compared our method with other DL methods [56], here named Zhang’s method.

A 2D Bentheimer sandstone image with $\phi_{\text{porosity}} = 0.2238$ was the input for 3D reconstruction. Figure 8 depicts the comparison of the Bentheimer sandstone image reconstruc-

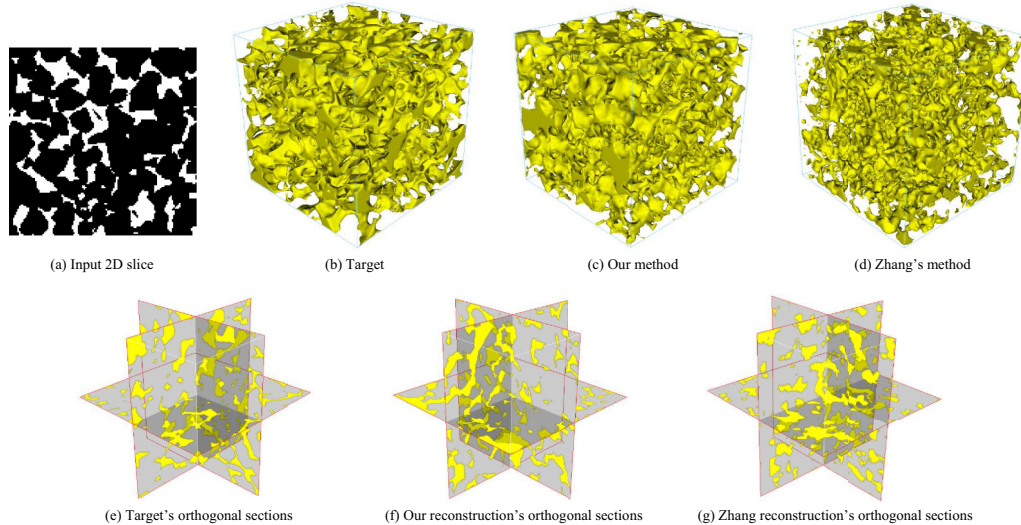


FIG. 8. Sample 1: Visual comparison of reconstruction results of Bentheimer sandstone.

tion results. Figures 8(a) and 8(b) are the input 2D slice and target 3D structure, respectively. Figures 8(c) and 8(d) are two reconstructed 3D structures with our method and Zhang's method, respectively. Figures 8(e)–8(g) are the orthogonal sections of Figs. 8(b) and 8(d), respectively.

The two reconstructed structures resembled the original 3D structure in the exterior appearance and interior of structure. In addition, the orthogonal sections of the two reconstructed 3D structures can learn the morphological properties and textures of the 2D images. From the perspective of visual inspection, two methods can reproduce the connectivity and isotropy of sandstone. However, our constructions much resemble the target 3D structure according to morphological characteristics. Thus, we can see that the reconstructed 3D structure with our method effectively reproduced the connectivity and morphological features of the original 3D structure, and also could maintain the homogeneity and pore connectivity. Meanwhile, the reconstructed 3D structure showed continuous constant variation in pore distribution.

However, the visualization and the observation of the target structure and two reconstructed structures were not sufficient to adequately measure the accuracy and stability of the proposed method. Therefore, we gave a comprehensive quantitative comparison below, including some morphological descriptor functions and LPDs, and 3D microstructure pore throat parameters. The capability of the proposed method was further confirmed.

Compared to the target structure's porosity 0.2222, the average porosity of the structures reconstructed by our method and Zhang's method are 0.2208 ± 0.0029 (mean \pm standard deviation) and 0.2146 ± 0.0047 , respectively. Figures 9(a)–9(c) show the results of the three morphological descriptors (i.e., S_2 , L , and C_2) of the target microstructure and two generated microstructures. Figure 9(d) gives the comparison of the LPDs. The local porosity has two key values: (i) the peak value, which always closely corresponds to the porosity of the core, and (ii) the concentration of the distribution (i.e., the width of the curve), which indicates the homogeneity of a core. In Fig. 9, the blue short dot line represents the average of the ten reconstructions of Zhang's method. Both Zhang's

method and our method can depict statistical features. These statistical descriptors of the generated microstructures are consistent with those of the target structure. We can obviously find that the sizes of the peak value and the concentration of porosity distributions between the two reconstructed microstructures and the target images were similar. However, our method exhibits better behavior than Zhang's method because the curve of our method is closer to that of the target 3D structure.

In addition, the storage and transport properties about the microstructure are mainly affected by the distribution and volume of pores, as well as the size and distribution of the throat connecting the pores. The shape factor is directly related to the morphology of the pores. The coordination number and effective permeability reveal the pore connectivity. We contrasted the 3D structure of the target with two reconstructed structures through some commonly used 3D microstructure parameters, as listed in Table IV. From the table, we can find that the parameters of our results are closer to those of the target CT compared with the results of Zhang's method.

From the above analysis, through both visual and quantitative comparisons, our method performs well and produces different results than Zhang's method.

2. Fontainebleau sandstone reconstruction

Fontainebleau sandstones [74] are usually chosen as a reference standard for validating models due to their special properties. They consist of well-sorted single-crystal quartz grains. They do not have clay and present only intergranular pores with multiple variations.

Two kinds of Fontainebleau sandstones with different porosities were selected here for 3D reconstructions. The size of the selected Fontainebleau sandstones is 128^3 with the resolution $7.5 \mu\text{m}$. The average porosities of the two sandstones were 0.2016 and 0.1631, respectively. Two slice images with porosity 0.2014 and 0.1636 were selected as input images for testing the model.

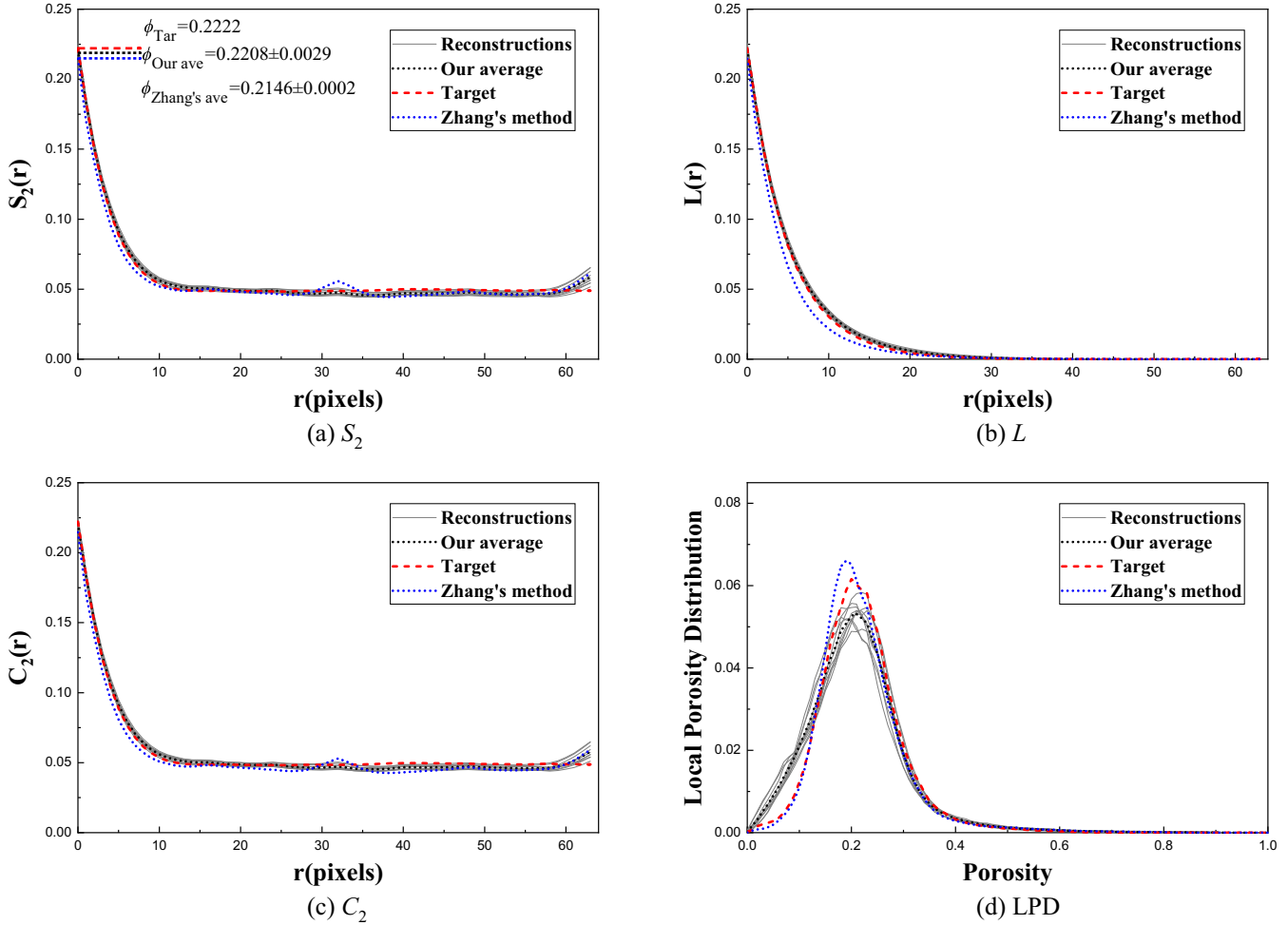


FIG. 9. Sample 1: Comparison of S_2 , L , C_2 , and LPD among the ten reconstructions, their average, and target.

(a) The first-type porosity of Fontainebleau sandstone reconstruction. The comparison of the sandstone image reconstruction results is shown in Fig. 10, where Figs. 10(a)–10(c) are the input 2D slice, target 3D structure, and reconstructed 3D structure, respectively; Figs. 10(d) and 10(e) are the orthogonal sections of Figs. 10(b) and 10(c), respectively.

We can clearly observe that the reconstructed 3D structure effectively recreated the connectivity and morphological features of the original 3D structure. It is clear that the proposed method could keep the homogeneity and pore connectivity,

and the transformation of the 3D pore space can also be seen. In addition, the orthogonal sections of the reconstructed 3D structure can well recur the morphological properties and textures of the 2D images. Further, we made a comprehensive quantitative comparisons of some morphological descriptor functions, LPDs, and 3D microstructure pore throat parameters.

The average porosity of the reconstructed structure was 0.2009 ± 0.0057 , which is very close to the target structure’s porosity 0.2016. Figures 11(a)–11(c) show the comparisons

TABLE IV. Sample 1: Comparison of microstructure parameters.

Parameters	Target	Average of our reconstructions	Average of Zhang’s method
Amount of pores	702	696	1046
Amount of throats	1720	1605	1829
Average shape factor	0.030	0.030	0.029
Average size of pore radius (m)	2.29×10^{-5}	2.26×10^{-5}	2.02×10^{-5}
Average size of throat radius (m)	9.69×10^{-6}	9.74×10^{-6}	9.35×10^{-6}
Average volume of pore (m^3)	9.27×10^{-13}	9.42×10^{-13}	6.05×10^{-13}
Average volume of throat (m^3)	5.75×10^{-14}	5.68×10^{-14}	4.86×10^{-14}
Average ratio of radius size of pore and throat	0.282	0.282	0.276
Average coordination number	4.903	4.618	3.499
Effective permeability (m^2)	1.80×10^{-12}	1.60×10^{-12}	8.61×10^{-13}

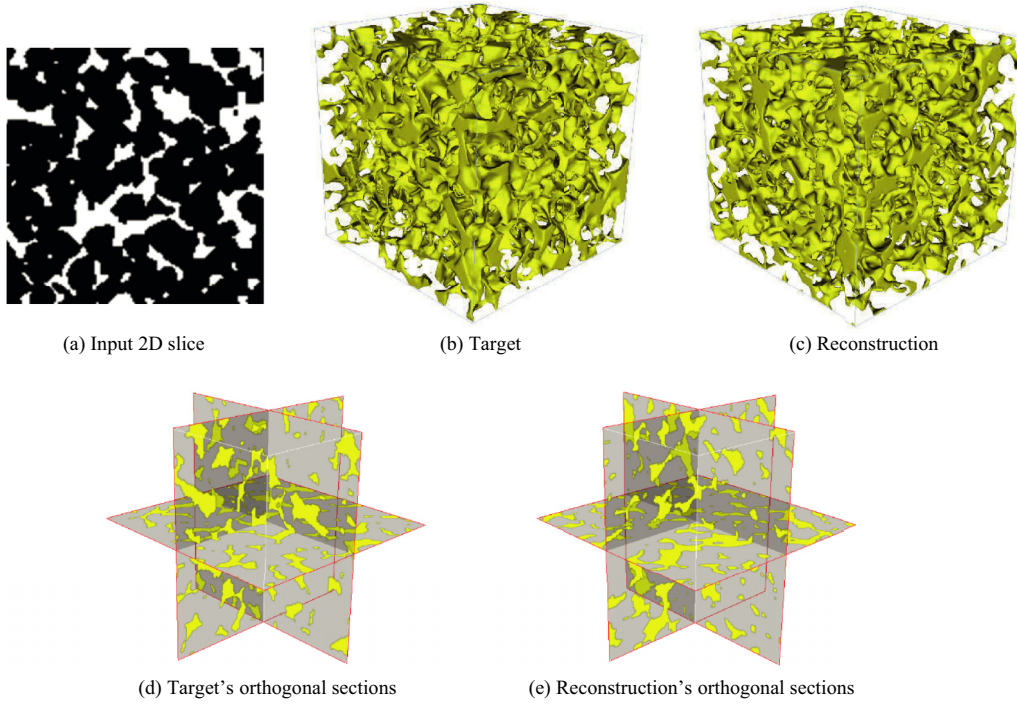


FIG. 10. Sample 2: Visual comparison of reconstruction results of Fontainebleau sandstone.

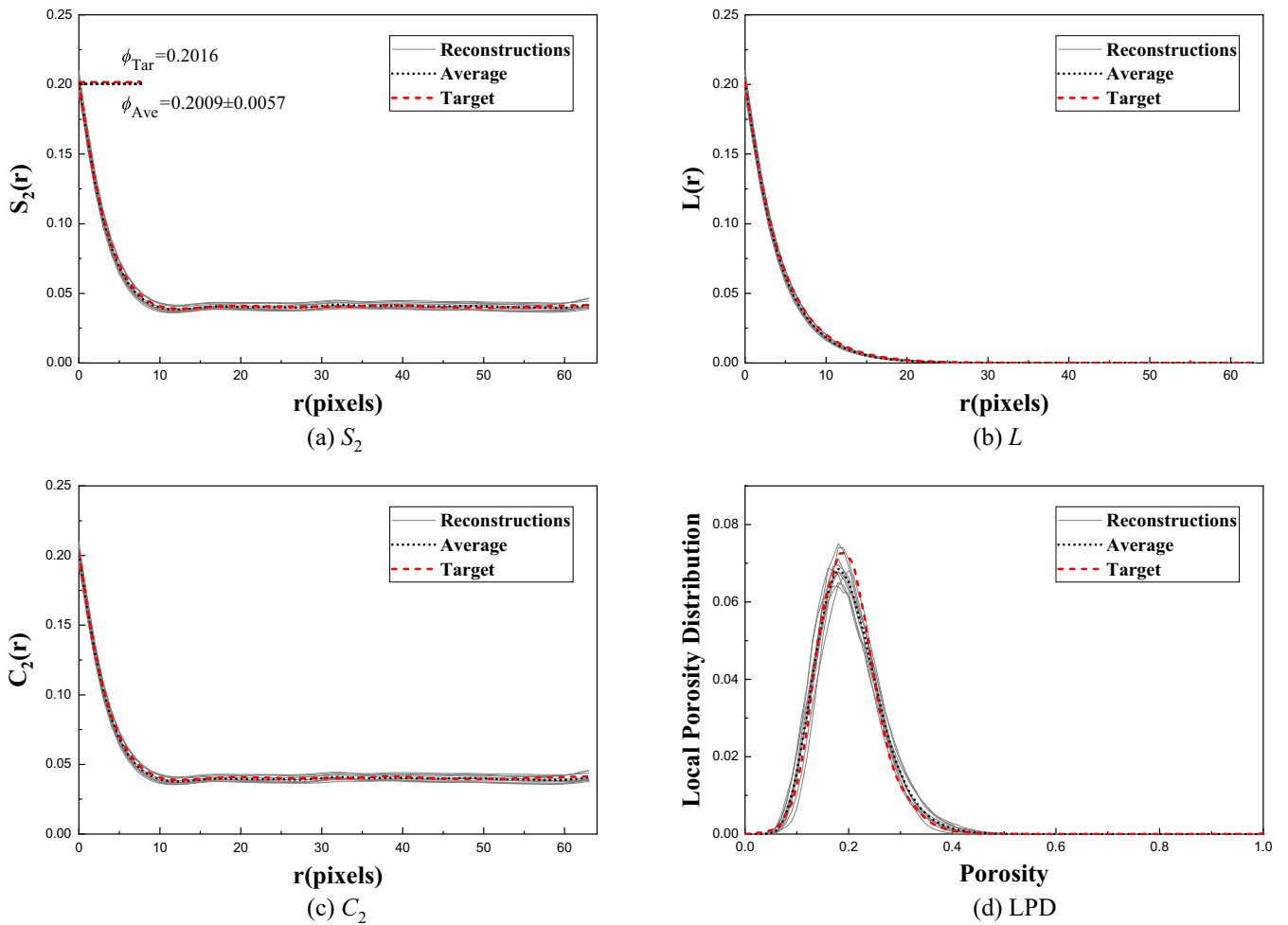


FIG. 11. Sample 2: Comparison of S_2 , L , C_2 and LPD among the ten reconstructions, their average, and target.

TABLE V. Sample 2: Comparison of microstructure parameters.

Parameters	Target	Average of reconstructions
Amount of pores	829	923
Amount of throats	1850	1771
Average shape factor	0.030	0.031
Average size of pore radius (m)	1.35×10^{-5}	1.34×10^{-5}
Average size of throat radius (m)	5.91×10^{-6}	6.15×10^{-6}
Average volume of pore (m^3)	1.86×10^{-13}	1.67×10^{-13}
Average volume of throat (m^3)	1.31×10^{-14}	1.33×10^{-14}
Average ratio of radius size of pore and throat	0.279	0.271
Average coordination number	4.466	3.840
Effective permeability (m^2)	5.10×10^{-13}	5.00×10^{-13}

of the three morphological functions (i.e., S_2 , L , and C_2) of the target microstructure and generated microstructure. It can be seen that the three statistical functions of the generated microstructure matched well with those of the real images, which proves that our method was able to accurately reconstruct and recover the connectivity of the pores.

On the other hand, to farther validate the proposed method to generate the local distribution of pores, we compared the LPD. Figure 11(d) shows the comparison of the local porosity. We can clearly observe that the LPD of the target 3D structure was very close to that of the reconstructed 3D structure by our method; the size of the peak value and the concentration of porosity distribution were similar.

In addition, we compared the 3D structure of the target with reconstructed images through 3D microstructure parameters. Table V shows the comparison of 3D microstructure parameters between the target and the average of ten reconstructions. We can observe that the 3D microstructure parameters of the generated images are similar to those of the target images, indicating that our method can accurately imitate the pore structure and spatial connectivity. On the other hand, the number of pores in our reconstructed 3D structure was slightly larger than that in the target structure, which is due to the existence of some isolated small pores, resulting in a certain low pore volume and average coordination number.

(b) *The second-type porosity of Fontainebleau sandstone reconstruction.* To verify the generalizability of our method, a 2D Fontainebleau sandstone image with $\phi_{\text{Porosity}} = 0.1636$ was input for the reconstruction of the 3D structure. Figure 12 shows the contrast of the sandstone image reconstruction results, where Figs. 12(a)–12(c) are the input 2D slice, target 3D structure, and reconstructed 3D structure, respectively; Figs. 12(d) and 12(e) are the orthogonal sections of Figs. 12(b) and 12(c), respectively.

From the visual point of view, the reconstructed image was similar to the original 3D image. The reconstructed structure inherited the homogeneity of the target structure and the connectivity of the pores, while its 3D pore space was also transformed.

The average porosity of the generated structure was 0.1634 ± 0.0014 , which is very approximate to the target structure's porosity 0.1631. Figures 13(a)–13(c) exhibit the comparison results of the three morphological descriptors (i.e., S_2 , L , and C_2) of the target microstructure and

reconstructed microstructure. Figure 13(d) shows the comparison of the LPDs. These statistical descriptors of the generated microstructure matched well with those of the real images.

In addition, with regard to pores and throats, we also compared the 3D structure of the target with reconstructed images through commonly used 3D microstructure parameters, as shown in Table VI. We can observe that the 3D microstructure parameters of the generated images captured the structural features of the target images. Meanwhile, due to the existence of some isolated small pores, the number of pores in the reconstructed 3D structure increased compared with that in the target, which led to a lower pore volume and average coordination number.

3. Berea sandstone reconstruction

Berea sandstones [75,76] have been widely used for many years by the petroleum industry as a standard material for core analysis studies and laboratory core experiments. The Berea sandstones are a medium to fine-grained fluvial-phase sandstones. These sandstones are relatively well distributed and well characterized. They consist of subangular, well-sorted, rounded quartz grains, but also contain minor feldspar, dolomite, and clay, most of which are free of preexisting cracks. Thus, Berea sandstones were selected to verify our method. The size of the selected Berea sandstones was 128^3 with the resolution $5.549 \mu\text{m}$. The porosity of this sandstone was 0.2099.

A 2D Berea sandstone image with $\phi_{\text{porosity}} = 0.2069$ was input for 3D reconstruction. Figure 14 depicts the comparison of the Berea sandstone image reconstruction results. Figures 14(a)–14(c) are the input 2D slice, target 3D structure, and reconstructed 3D structure, respectively; Figs. 14(d) and 14(e) are the orthogonal sections of Figs. 14(b) and 14(c), respectively.

The exterior and interior of the reconstructed image resembled the original 3D image. The reconstructed 3D structure showed continuous variation in pore distribution. Moreover, we made a comprehensive quantitative comparison between the generated image and the target 3D image. Microstructure pore throat parameters include the above four evaluation functions, as well as 3D microstructure pore throat parameters.

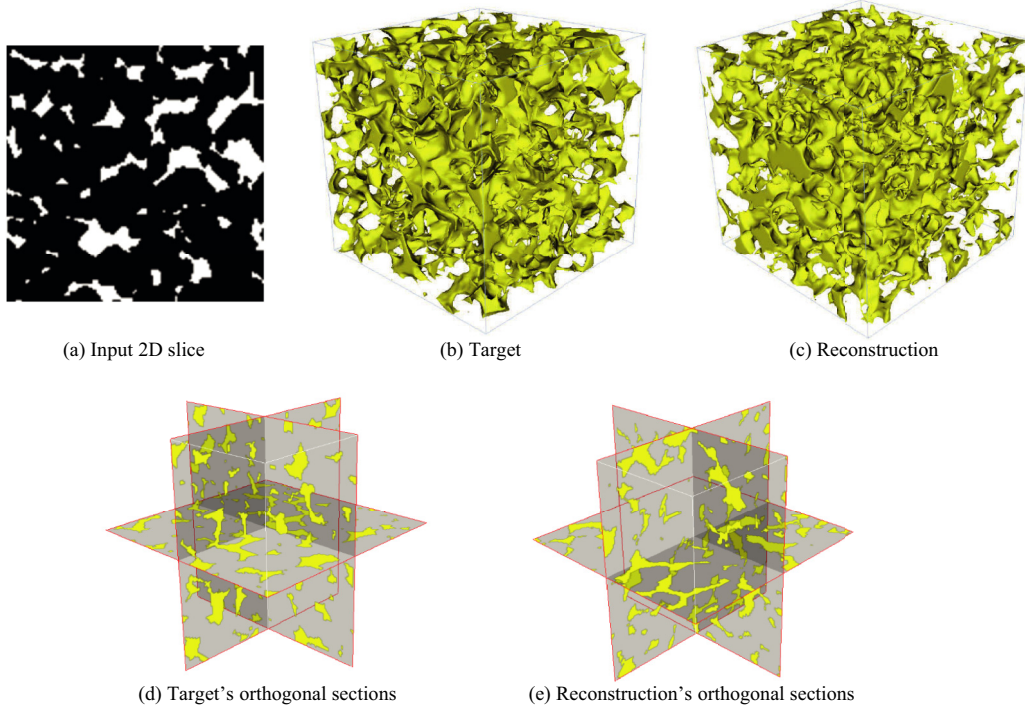


FIG. 12. Sample 3: Visual comparison of reconstruction results of Fontainebleau sandstone.

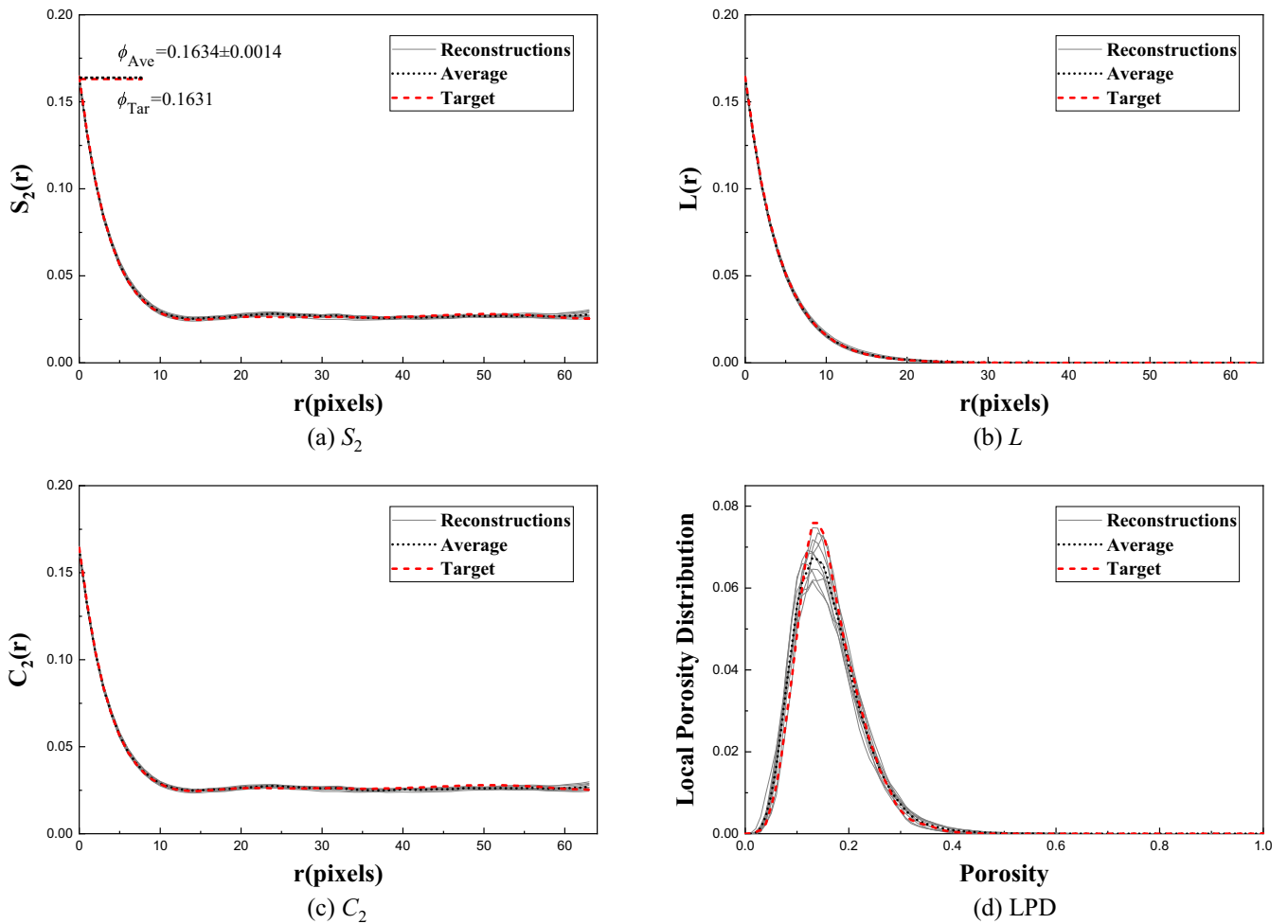


FIG. 13. Sample 3: Comparison of S_2 , L , C_2 , and LPD among the ten reconstructions, their average, and target.

TABLE VI. Sample 3: Comparison of microstructure parameters.

Parameters	Target	Average of reconstructions
Amount of pores	646	687
Amount of throats	1360	1257
Average shape factor	0.030	0.030
Average size of pore radius (m)	2.71×10^{-5}	2.68×10^{-5}
Average size of throat radius (m)	1.17×10^{-6}	1.22×10^{-6}
Average volume of pore (m^3)	1.52×10^{-12}	1.56×10^{-12}
Average volume of throat (m^3)	1.29×10^{-13}	1.44×10^{-13}
Average ratio of radius size of pore and throat	0.285	0.286
Average coordination number	4.214	3.665
Effective permeability (m^2)	1.16×10^{-13}	1.06×10^{-13}

Compared to the target structure's porosity 0.2099, the average porosity of the reconstructed Berea structure was 0.2103 ± 0.0047 . Figures 15(a)–15(c) show the results of the three morphological descriptors (i.e., S_2 , L , and C_2) of the target microstructure and generated microstructure. Figure 15(d) displays the comparison of the LPDs. These statistical descriptors of the generated microstructure matched well with those of the target images. We can clearly see that the sizes of the peak value and the concentration of porosity distributions between the reconstructed microstructure and the target images were similar.

In addition, to further investigate its internal mechanism, we compared the 3D structure of the target with reconstructed images through some commonly used 3D microstructure parameters, as listed in Table VII. We can observe that the 3D microstructure parameters of the generated images matched the pore structure and spatial connectivity of the target im-

ages. The existence of some isolated small pores increased the number of pores in the reconstructed 3D structure compared with that in the target, which resulted in a lower pore volume and average coordination number.

4. Ketton reconstruction

Ketton limestones [77] are mainly nonferroan calcite grains. The microstructure shows that the samples have reduced porosity. The samples comprise large grains compared to the entire image size from Figs. 16(a) and 16(b). The size of the Ketton limestones was 128^3 with resolution $15.2 \mu\text{m}$. The porosity of the samples was 0.1442.

A 2D Ketton image with $\phi_{\text{porosity}} = 0.1448$ was input for 3D reconstruction. Figure 16 describes the comparison of the Ketton image reconstruction results. Figures 16(a)–16(c) are the input 2D slice, target 3D structure, and our reconstructed

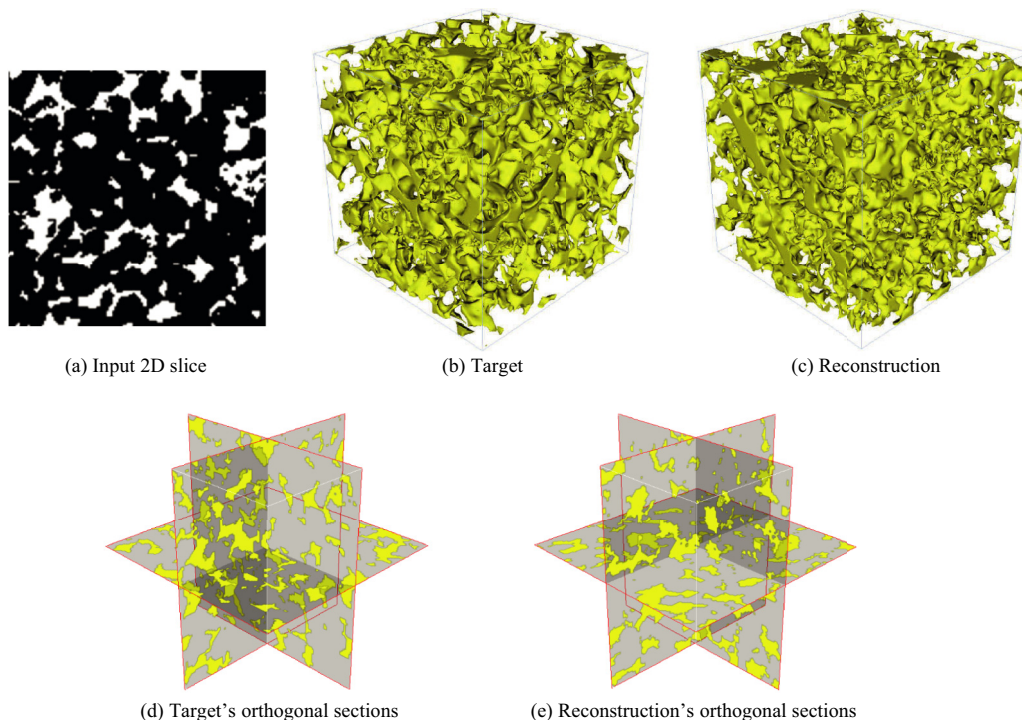
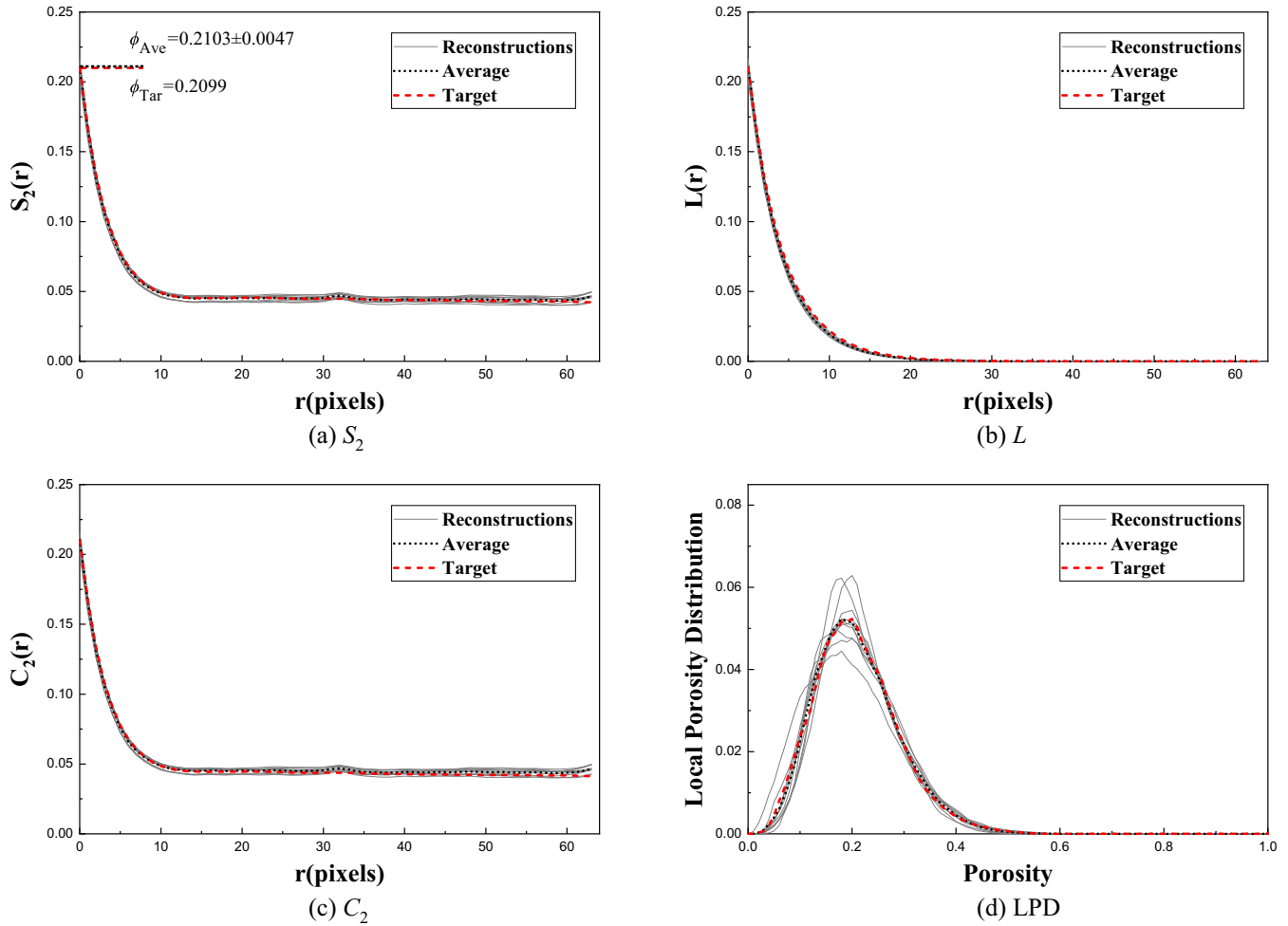


FIG. 14. Sample 4: Visual comparison of reconstruction results of Berea sandstone.


 FIG. 15. Sample 4: Comparison of S_2 , L , C_2 , and LPD among the ten reconstructions, their average, and target.

3D structure, respectively; Figs. 16(d) and 16(e) are the orthogonal sections of Figs. 16(b) and 16(c), respectively.

There is no significant difference between the reconstructed image and the original 3D image in the exterior appearance and interior of structure. The reconstructed 3D structure preserved the large grains well and showed some sparse small pores in pore distribution. Moreover, we contrasted some of the significant microstructure pore throat parameters in the generated image and the target 3D image.

Compared to the target structure's porosity 0.1442, the average porosity of the reconstructed Ketton structure was 0.1390 ± 0.0076 . Figures 17(a)–17(c) display the results of the three morphological descriptors (i.e., S_2 , L , and C_2) of the target microstructure and generated microstructure. Figure 17(d) demonstrates the comparison of the LPDs. These statistical descriptors of the generated microstructure accord closely with those of the target images. From the LPD, we can clearly find that there are similarities in porosity distributions

TABLE VII. Sample 4: Comparison of microstructure parameters.

Parameters	Target	Average of reconstructions
Amount of pores	972	1110
Amount of throats	2081	2158
Average shape factor	0.029	0.029
Average size of pore radius (m)	1.76×10^{-5}	1.67×10^{-5}
Average size of throat radius (m)	8.20×10^{-6}	8.17×10^{-6}
Average volume of pore (m^3)	5.20×10^{-13}	4.50×10^{-13}
Average volume of throat (m^3)	3.75×10^{-14}	3.90×10^{-14}
Average ratio of radius size of pore and throat	0.286	0.292
Average coordination number	4.283	3.887
Effective permeability (m^2)	1.03×10^{-12}	8.52×10^{-13}

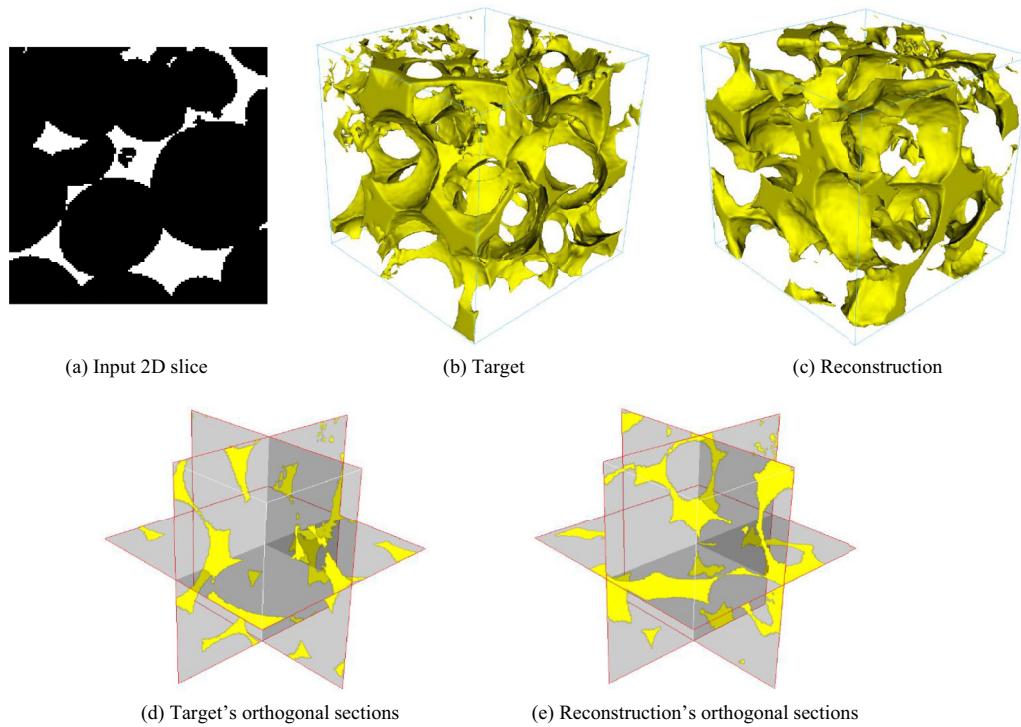


FIG. 16. Sample 5: Visual comparison of reconstruction results of Ketton.

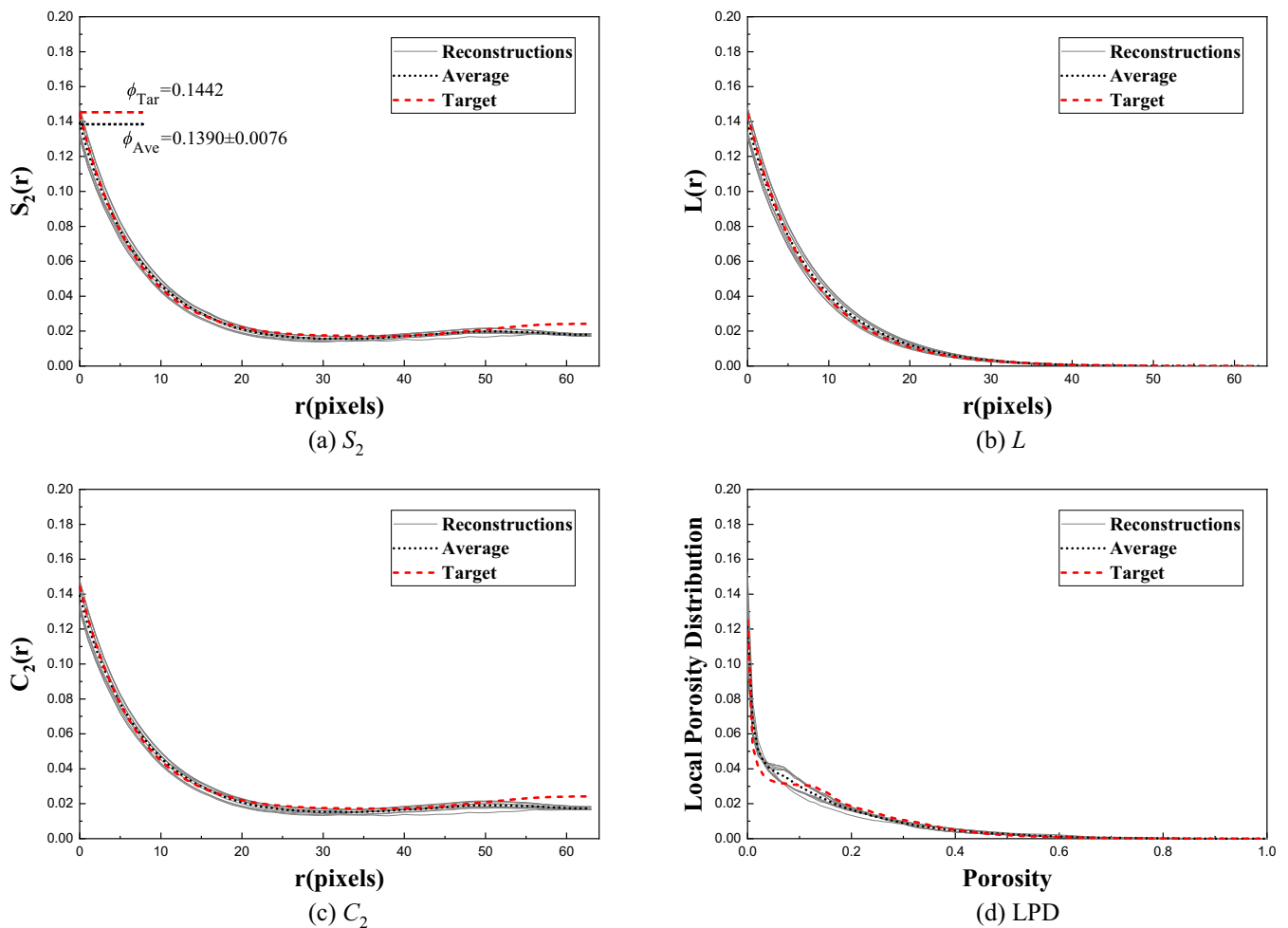


FIG. 17. Sample 5: Comparison of S_2 , L , C_2 , and LPD among the ten reconstructions, their average, and target.

TABLE VIII. Sample 5: Comparison of microstructure parameters.

Parameters	Target	Average of reconstructions
Amount of pores	175	141
Amount of throats	389	246
Average shape factor	0.029	0.029
Average size of pore radius (m)	3.95×10^{-5}	4.24×10^{-5}
Average size of throat radius (m)	1.79×10^{-5}	1.94×10^{-5}
Average volume of pore (m^3)	5.30×10^{-12}	6.43×10^{-12}
Average volume of throat (m^3)	3.43×10^{-13}	5.42×10^{-13}
Average ratio of radius size of pore and throat	0.300	0.292
Average coordination number	4.457	3.514
Effective permeability (m^2)	8.67×10^{-12}	3.25×10^{-12}

significantly between the reconstructed microstructure and the target images.

Further, we compared the 3D structure of the target with reconstructed images through 3D microstructure parameters, as given in Table VIII. We can observe that there are some small differences in the microstructure parameters and spatial connectivity between the 3D generated images and the target images.

V. CONCLUSION

For digital cores with complex topology, it is a prominent and difficult problem to generate 3D structures with equivalent statistical features from 2D slices. In this paper, to address this conundrum, we present a deep generative model for reconstructing 3D structures by extracting statistical features of 2D slices. When training the model, we used the Wasserstein distance with gradient penalty as the evaluation criterion for two different images. Thus, a large number of diverse 3D structures with similar statistical features can be generated from a single input 2D slice. The designed method was applied on five different core images, all of which achieved satisfactory results. All the reconstructed 3D structures had statistical features consistent with the target 3D structures. Further, we also compare our method with Zhang's methods. In terms of visual and quantitative comparisons, our method not only has a better reconstruction effect but also can be applied to homogeneous and isotropic kinds of porous media. Through experimental verification, the designed method has the following advantages. First, our method can achieve accurate, efficient, and stable 3D reconstruction. More importantly, immediately after training, a 3D structure can be reconstructed from a 2D slice. Second, the method is designed for statistical reconstruction of 3D structures based on homogeneity and isotropy 2D slices. Therefore, the method is not only suitable for the 3D reconstruction of core images but also for 3D reconstruction of homogeneous

and isotropic porous media, such as silica or battery materials. However, our method also has some flaws. For heterogeneous and anisotropic cores, our method cannot be used to reconstruct cores with such characteristics by doing experiments. For the Ketton limestone in Figs. 16(a) and 16(b), a larger size reconstruction is needed to better characterize this core. This is an important direction that our method needs to be extended in further.

The reconstructed 3D structure was similar to the target 3D structure in terms of visualization. The reconstructed statistical parameter metrics agreed well with those of the target 3D structure, such as the two-point correlation function, lineal-path function, two-point cluster function, and LPD. However, there was a slight discrepancy in the comparison of microstructure parameters of the 3D structure, for two reasons. First, our network consists of an encoder, a generator, and three discriminators, and it is difficult to train these networks simultaneously, which results in our network not being optimally trained. Second, the loss function used in the network structure for generating 3D structures from 2D images is not suitable for solving this problem. If more adequate loss functions were designed, such as adding constraints for the two-point correlation function, then the accuracy of 3D reconstruction from 2D images could be further improved.

It is worth noting that the method we designed is suitable for homogeneous and isotropic cores, while real cores will be larger in size or more heterogeneous; therefore, we aim to continue our work in this direction.

ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China (Grant No. 62071315) and in part by Fundamental Research Funds for the Central Universities of China (Grant No. 31920180051).

- [1] S. Torquato and H. Haslach Jr., *Appl. Mech. Rev.* **55**, B62 (2002).
 [2] P. M. Adler and J.-F. Thovert, *Appl. Mech. Rev.* **51**, 537 (1998).

- [3] T. Bultreys, W. De Boever, and V. Cnudde, *Earth-Sci. Rev.* **155**, 93 (2016).
 [4] C. F. Berg, *Transp. Porous Media* **103**, 381 (2014).

- [5] A. Gupta, A. Cecen, S. Goyal, A. K. Singh, and S. R. Kalidindi, *Acta Mater.* **91**, 239 (2015).
- [6] M. Sahimi, *Flow and Transport in Porous Media and Fractured Rock from Classical Methods to Modern Approaches*, (John Wiley & Sons, 2011).
- [7] L. Zhu, C. Zhang, C. Zhang, X. Zhou, Z. Zhang, X. Nie, W. Liu, and B. Zhu, *Geofluids* **2019**, 1 (2019).
- [8] R. Bostanabad, Y. Zhang, X. Li, T. Kearney, L. C. Brinson, D. W. Apley, W. K. Liu, and W. Chen, *Prog. Mater. Sci.* **95**, 1 (2018).
- [9] M. Tianshou and C. Ping, *Pet. Explor. Dev.* **41**, 249 (2014).
- [10] H. Li, S. Singh, N. Chawla, and Y. Jiao, *Mater. Charact.* **140**, 265 (2018).
- [11] S. Schlüter, A. Sheppard, K. Brown, and D. Wildenschild, *Water Resour. Res.* **50**, 3615 (2014).
- [12] T. Sakamoto, in *Proceedings of the Compendium of Surface and Interface Analysis* (Springer, Berlin, 2018), pp. 181–186.
- [13] C. L. Y. Yeong and S. Torquato, *Phys. Rev. E* **57**, 495 (1998).
- [14] C. L. Y. Yeong and S. Torquato, *Phys. Rev. E* **58**, 224 (1998).
- [15] S. K. Alexander, P. Fieguth, M. A. Ioannidis, and E. R. Vrscaj, *Math. Geosci.* **41**, 357 (2009).
- [16] D. D. Chen, Q. Teng, X. He, Z. Xu, and Z. Li, *Phys. Rev. E* **89**, 013305 (2014).
- [17] M. Gao, Q. Teng, X. He, C. Zuo, and Z. J. Li, *Phys. Rev. E* **93**, 012140 (2016).
- [18] L. M. Pant, S. K. Mitra, and M. Secanell, *Phys. Rev. E* **90**, 023306 (2014).
- [19] Y. Jiao, F. H. Stillinger, and S. Torquato, *Phys. Rev. E* **76**, 031110 (2007).
- [20] Y. Ju, J. Zheng, M. Epstein, L. Sudak, J. Wang, and X. Zhao, *Comput. Methods Appl. Mech. Eng.* **279**, 212 (2014).
- [21] Y. Ju, Y. Huang, J. Zheng, X. Qian, H. Xie, and X. Zhao, *Comput. Geosci.* **101**, 10 (2017).
- [22] J. Feng, Q. Teng, X. He, and X. Wu, *Acta Mater.* **159**, 296 (2018).
- [23] Y. Ju, Y. Huang, W. Gong, J. Zheng, H. Xie, L. Wang, and X. Qian, *IEEE Trans. Geosci. Remote Sens.* **57**, 1873 (2018).
- [24] K. M. Gerke, M. V. Karsanina, R. V. Vasilyev, and D. Mallants, *Europhys. Lett.* **106**, 66002 (2014).
- [25] K. M. Gerke and M. V. Karsanina, *Europhys. Lett.* **111**, 56002 (2015).
- [26] K. M. Gerke, M. V. Karsanina, and R. Katsman, *Phys. Rev. E* **100**, 053312 (2019).
- [27] F. B. Guardiano and R. M. Srivastava, in *Proceedings of the Geostatistics Troia '92* (Springer, Berlin, 1993), Vol. 1, pp. 133–144.
- [28] G. Mariethoz, P. Renard, and J. Straubhaar, *Water Resour. Res.* **46** (2010).
- [29] G. Mariethoz and P. Renard, *Math. Geosci.* **42**, 245 (2010).
- [30] P. Tahmasebi and M. Sahimi, *Phys. Rev. E* **85**, 066709 (2012).
- [31] P. Tahmasebi, F. Javadpour, and M. Sahimi, *Transp. Porous Media* **110**, 521 (2015).
- [32] P. Tahmasebi, *Water Resour. Res.* **53**, 5980 (2017).
- [33] Y. Li, X. He, Q. Teng, J. Feng, and X. Wu, *Phys. Rev. E* **97**, 043306 (2018).
- [34] Y. Li, Q. Teng, X. He, J. Feng, and S. Xiong, *J. Pet. Sci. Eng.* **174**, 968 (2019).
- [35] Z. Xia, Q. Teng, X. Wu, J. Li, and P. Yan, *Phys. Rev. E* **104**, 045308 (2021).
- [36] J. Fu, S. Cui, S. Cen, and C. Li, *Comput. Methods Appl. Mech. Eng.* **373**, 113516 (2021).
- [37] J. Fu, D. Xiao, D. Li, H. R. Thomas, and C. Li, *Comput. Methods Appl. Mech. Eng.* **390**, 114532 (2022).
- [38] J. B. Boisvert, M. J. Pyrcz, and C. V. Deutsch, *Nat. Resour. Res.* **16**, 313 (2007).
- [39] P. W. Mirowski, D. M. Tetzlaff, R. C. Davies, D. S. McCormick, N. Williams, and C. Signer, *Math. Geosci.* **41**, 447 (2009).
- [40] J. Feng, Q. Teng, X. He, L. Qing, and Y. Li, *Comput. Mater. Sci.* **144**, 181 (2018).
- [41] S. Strebelle, *Math. Geol.* **34**, 1 (2002).
- [42] M. L. Gao, X. H. He, Q. Z. Teng, C. Zuo, and D. D. Chen, *Phys. Rev. E* **91**, 013308 (2015).
- [43] A. Agrawal and A. Choudhary, *MRS Commun.* **9**, 779 (2019).
- [44] D. P. Kingma and M. Welling, [arXiv:1312.6114](https://arxiv.org/abs/1312.6114).
- [45] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, *Adv. Neural Inform. Process. Syst.* **27** (2014).
- [46] Y.-J. Cao, L.-L. Jia, Y.-X. Chen, N. Lin, C. Yang, B. Zhang, Z. Liu, X.-X. Li, and H.-H. Dai, *IEEE Access* **7**, 14985 (2018).
- [47] M. Mirza and S. Osindero, [arXiv:1411.1784](https://arxiv.org/abs/1411.1784).
- [48] A. Radford, L. Metz, and S. Chintala, [arXiv:1511.06434](https://arxiv.org/abs/1511.06434).
- [49] A. Krizhevsky, I. Sutskever, and G. E. Hinton, *Adv. Neural Inform. Process. Syst.* **25** (2012).
- [50] M. Arjovsky, S. Chintala, and L. Bottou, in *Proceedings of the International Conference on Machine Learning* (PMLR, 2017), pp. 214–223.
- [51] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, *Adv. Neural Inform. Process. Syst.* **30** (2017).
- [52] L. Mosser, O. Dubrule, and M. J. Blunt, *Phys. Rev. E* **96**, 043309 (2017).
- [53] L. Mosser, O. Dubrule, and M. J. Blunt, *Transp. Porous Media* **125**, 81 (2018).
- [54] R. Shams, M. Masihi, R. B. Boozarjomehry, and M. J. Blunt, *Adv. Water Resour.* **158**, 104064 (2021).
- [55] J. Feng, Q. Teng, B. Li, X. He, H. Chen, and Y. Li, *Comput. Methods Appl. Mech. Eng.* **368**, 113043 (2020).
- [56] F. Zhang, Q. Teng, H. Chen, X. He, and X. Dong, *Comput. Mater. Sci.* **186**, 110018 (2021).
- [57] T. Zhang, P. Xia, and F. Lu, *J. Pet. Sci. Eng.* **207**, 109151 (2021).
- [58] W. Zha, X. Li, Y. Xing, L. He, and D. Li, *Adv. Geo-Energy Res.* **4**, 107 (2020).
- [59] A. Valsecchi, S. Damas, C. Tubilleja, and J. Arechalde, *Neurocomputing* **399**, 227 (2020).
- [60] D. Volkhonskiy, E. Muravleva, O. Sudakov, D. Orlov, E. Burnaev, D. Koroteev, B. Belozarov, and V. Krutko, *Phys. Rev. E* **105**, 025304 (2022).
- [61] T. Zhang, Q. Liu, X. Wang, X. Ji, and Y. Du, *Comput. Geosci.* 105151 (2022).
- [62] D. Cao, Z. Hou, Q. Liu, and F. Fu, *J. Pet. Sci. Eng.* **208**, 109590 (2022).
- [63] Y. Li, X. He, W. Zhu, M. AlSinan, H. Kwak, and H. Hoteit, in *Proceedings of the International Petroleum Technology Conference* (OnePetro, 2022).
- [64] J. Feng, X. He, Q. Teng, C. Ren, H. Chen, and Y. Li, *Phys. Rev. E* **100**, 033308 (2019).

- [65] Y. Wang, Q. Teng, X. He, J. Feng, and T. Zhang, *Comput. Geosci.* **133**, 104314 (2019).
- [66] H. Chen, X. He, H. Yang, J. Feng, and Q. Teng, *Expert Syst. Appl.* **193**, 116440 (2022).
- [67] L. A. Gatys, A. S. Ecker, and M. Bethge, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, New York, 2016), pp. 2414–2423.
- [68] M. Arjovsky and L. Bottou, [arXiv:1701.04862](https://arxiv.org/abs/1701.04862).
- [69] B. Lu and S. Torquato, *Phys. Rev. A* **45**, 922 (1992).
- [70] S. Torquato, J. Beasley, and Y. Chiew, *J. Chem. Phys.* **88**, 6540 (1988).
- [71] R. Hilfer, *Phys. Rev. B* **45**, 7115 (1992).
- [72] J. Hu and P. Stroeven, *Cem. Concr. Res.* **35**, 233 (2005).
- [73] A. E. Peksa, K.-H. A. Wolf, and P. L. Zitha, *Mar. Pet. Geol.* **67**, 701 (2015).
- [74] D. A. Coker, S. Torquato, and J. H. Dunsmuir, *J. Geophys. Res.: Solid Earth* **101**, 17497 (1996).
- [75] P. L. Churcher, P. R. French, J. C. Shaw, and L. L. Schramm, in *SPE International Symposium on Oilfield Chemistry* (OnePetro, 1991).
- [76] P.-E. Øren and S. Bakke, *J. Pet. Sci. Eng.* **39**, 177 (2003).
- [77] D. Emery, J. Hudson, J. Marshall, and J. Dickson, *J. Geol. Soc.* **145**, 621 (1988).