Transp Porous Med (2018) 125:81–103
https://doi.org/10.1007/s11242-018-1039-9

Stochastic Reconstruction of an Oolitic Limestone by Generative Adversarial Networks

Lukas Mosser[1] · **Olivier Dubrule**[1] · **Martin J. Blunt**[1]

Received: 28 October 2017 / Accepted: 9 March 2018 / Published online: 21 March 2018

Abstract Stochastic image reconstruction is a key part of modern digital rock physics and material analysis that aims to create representative samples of microstructures for upsampling, upscaling and uncertainty quantification. We present new results of a method of three-dimensional stochastic image reconstruction based on generative adversarial neural networks (GANs). GANs are a family of unsupervised learning methods that require no a priori inference of the probability distribution associated with the training data. Thanks to the use of two convolutional neural networks, the discriminator and the generator, in the training phase, and only the generator in the simulation phase, GANs allow the sampling of large and realistic volumetric images. We apply a GAN-based workflow of training and image generation to an oolitic Ketton limestone micro-CT unsegmented gray-level dataset. Minkowski functionals calculated as a function of the segmentation threshold are compared between simulated and acquired images. Flow simulations are run on the segmented images, and effective permeability and velocity distributions of simulated flow are also compared. Results show that GANs allow a fast and accurate reconstruction of the evaluated image dataset. We discuss the performance of GANs in relation to other simulation techniques and stress the benefits resulting from the use of convolutional neural networks . We address a number of challenges involved in GANs, in particular the representation of the probability distribution associated with the training data.

Keywords Stochastic modeling · Neural networks · Porous media · Computer tomography

✉ Lukas Mosser
lukas.mosser15@imperial.ac.uk

Olivier Dubrule
o.dubrule@imperial.ac.uk

Martin J. Blunt
m.blunt@imperial.ac.uk

[1] Department of Earth Science and Engineering, Imperial College London, London SW7 2AZ, UK

1 Introduction

The microstructural characteristics of porous media play an important role in the understanding of numerous scientific and engineering applications such as the recovery of hydrocarbons from subsurface reservoirs (Blunt et al. 2013), sequestration of CO_2 (Singh et al. 2017) or the design of new batteries (Siddique et al. 2012). Modern microcomputer tomographic (micro-CT) methods have enabled the acquisition of high-resolution three-dimensional images at the scale of individual pores. Increased resolution comes at the cost of longer image acquisition time and limited sample size. Individual samples allow numerical and experimental assessment of the effective properties of the porous media, but give no insight into the variance of key microstructural properties. Therefore, an efficient method to generate representative volumetric models of porous media that allow the assessment of the effective properties is required. The generated images serve as an input to a digital rock physics workflow to represent the computational domain for numerical estimation of key physical properties (Berg et al. 2017).

Statistical methods aim at reconstructing porous media based on spatial statistical properties such as two-point pore–grain correlation functions. Quiblier (1984) has presented an extensive overview of the early literature of porous media reconstruction and provided an extension of the method of Joshi (1974) by reconstructing three-dimensional porous media based on the empirical covariance function and probability density function obtained from two-dimensional thin sections. Other statistical methods such as simulated annealing (Yeong and Torquato 1998; Jiao et al. 2008) allow high-quality three-dimensional reconstruction and incorporation of numerous statistical descriptors of porous media. Pant (2016) introduced a multi-scale simulated annealing algorithm allowing simulation of three-dimensional porous media at much lower computational cost than previous methods.

Methods to incorporate higher-order multi-point statistical (MPS) properties of porous media have been developed. These MPS functions are implicitly defined by two- or three-dimensional training images. Simulation algorithms based on multi-point statistics are therefore considered as training image-based algorithms. MPS simulation was originally developed in the context of generating realistic geological structures (Guardiano and Srivastava 1993; Caers 2001; Mariethoz and Caers 2014). With the advent of micron-resolution X-ray tomography (micro-CT imaging) (Flannery et al. 1987), which provides training images, MPS simulation techniques have been successfully applied to the stochastic reconstruction of three-dimensional porous media (Okabe and Blunt 2004, 2005, 2007).

Tahmasebi et al. (2012) and Tahmasebi and Sahimi (2012, 2013) have introduced a patch-based approach where sub-domains are simulated along a pre-defined path and populated based on a cross-correlation distance criterion (CCSIM). This approach is similar to the image quilting algorithm by Efros and Freeman (2001) and Mariethoz and Caers (2014) but corrects mismatching patches in overlapping or neighboring domains. Tahmasebi et al. (2017) present a method for fast reconstruction of granular porous media from a single two- or three-dimensional training image using a method closely related to CCSIM. They obtain significant speedup in computational time by incorporating a fast Fourier transform and a multi-scale approach. A graph-based approach is used to resolve non-physical regions at the boundaries of simulated patches of grains.

Object-based methods describe the material domain by locating geometrical bodies of random size at locations provided by a spatial point process. The so-called Boolean model is a particular case where the randomly placed bodies, typically spheres, are allowed to overlap (Matheron 1975; Chiu et al. 2013). Object-based methods may also allow interaction of particles to be incorporated. They have successfully been used to describe complex and heterogeneous materials (Torquato 2013).

Process models reconstruct the pore and grain structure of materials by mimicking how they were formed. Øren and Bakke (2003) have created reconstructions of sandstones by reproducing the natural processes of sedimentation, compaction and diagenesis.

This contribution presents a training image-based method of image reconstruction using a class of deep generative methods called generative adversarial networks (GANs) first introduced by Goodfellow et al. (2014). Recently, Mosser et al. (2017) have shown that GANs allow the reconstruction of three-dimensional porous media based on segmented volumetric images. Their study applied GANs to three segmented images of rock samples. They showed that GANs represent a computationally efficient method for the fast generation of large volumetric images that capture the statistical and morphological features, as well as the effective permeability.

We expand on the work of Mosser et al. (2017) and investigate the ability of generative adversarial networks to create stochastic reconstructions of an unsegmented micro-CT scan of a larger oolitic Ketton limestone sample. We evaluate the four Minkowski functionals for the three-dimensional datasets as a function of the gray-level threshold. In addition to the numerical evaluation of permeability as shown by Mosser et al. (2017), we compare velocity distributions of the original porous medium and samples obtained from the GAN. We also provide details of the convolution approach used by GANs. Furthermore we evaluate the reconstruction process within the trained generative function and highlight the parametric and differentiable nature of the obtained generative function. We evaluate the computational cost of GAN-based image simulation with reported values of computational run time for a variety of other reconstruction methods of equal reconstruction quality. We also investigate how the image representation evolves along the different layers of the GAN network, and discuss the benefits that can be derived from the differentiable nature of the parameterization used by GANs.

2 Generative Adversarial Networks

Generative adversarial networks are a deep learning method for generating samples from arbitrary probability distributions (Goodfellow et al. 2014; Goodfellow 2017). GANs do not impose any a priori model on the probability density function and are therefore also referred to as an implicit method. Without the need to specify an explicit model, GANs provide efficient sampling methods for high-dimensional and intractable density functions.

In the case of CT images of porous media, we can define an image x to be a sample of a real, unknown probability density function (pdf) of images p_{data} of which we have acquired a number of samples which serve as training images. In our example, the training set is comprised of 5832 sub-domains (64^3 voxel) of the original micro-CT image. Sub-domains are extracted without any overlap, and each training image represents the originally acquired dataset.

GANs consist of two functions: a generator whose role it is to generate samples of the unknown density $p_{\text{data}}(\mathbf{x})$ and a discriminator function D that tries to distinguish between samples from the training set and synthetic images created by the generator. The generator G is defined by its parameters θ and performs a mapping from a random prior $\mathbf{z}$ to the image domain:

$$\mathbf{z} \sim \mathcal{N}(0,1)^{d\times 1\times 1\times 1} \tag{1}$$

$$G_\theta : \mathbf{z} \rightarrow \mathbb{R}^{1\times 64\times 64\times 64} \tag{2}$$

where d is the dimensionality of the random prior.

The discriminator $D_\omega(\mathbf{x})$ assigns a probability to an image x being a sample of the true data distribution p_{data}:

$$D_\omega : \mathbb{R}^{1\times 64\times 64\times 64} \rightarrow [0,1] \tag{3}$$

where values close to 1 represent a high probability of being a sample of $\mathbf{x} \sim p_{\text{data}}(\mathbf{x})$.

We represent both the generator $G_\theta(\mathbf{z})$ and the discriminator $D_\omega(\mathbf{x})$ by differentiable neural networks with parameters θ and ω, respectively. This allows us to use backpropagation combined with mini-batch gradient descent to optimize the generator and discriminator according to the functional:

$$\min_\theta \max_\omega \{\mathbb{E}_{\mathbf{x}\sim p_{\text{data}}}[\log\ D_\omega(\mathbf{x})] + \mathbb{E}_{\mathbf{x}\sim p_{\mathbf{z}}}[\log(\ 1 - D_\omega(G_\theta(\mathbf{z})))]\} \tag{4}$$

The optimization criterion of the generator and discriminator (Eq. 4) is solved sequentially in a two-step procedure. We first train the discriminator to maximize its ability to distinguish real from fake samples. This is done in a supervised manner by training the discriminator on known real samples (Label 1) and samples created by the generator (Label 0). The binary cross-entropy is used as an objective function to compute the misclassification error:

$$H(\mathbf{y},\mathbf{y}') = -\sum_i (y_i \log(y_i') + (1-y_i)\log(1-y_i')) \tag{5}$$

where $\mathbf{y}'$ is a vector containing the output probability assigned by the discriminator for each element of a given mini-batch of samples. For each mini-batch of real images, we therefore optimize $H(\mathbf{1},\mathbf{y}')$ and for all fake samples $H(\mathbf{0},\mathbf{y}')$ (Eq. 5). The error is back-propagated while keeping the parameters of the generator constant.

In a second step, we train the generator to maximize its ability to "fool" the discriminator into misclassifying the images provided by the generator as real images. This is performed by computing the binary cross-entropy of the output of the discriminator on a mini-batch sampled from the generator $G_\theta(\mathbf{z})$ and requiring that the created labels be close to one, thereby computing $H(\mathbf{1},\mathbf{y}')$. The parameters of the generator are then modified to optimize $H(\mathbf{1},\mathbf{y}')$ by applying stochastic gradient descent while keeping the parameters of the discriminator constant.

Training of these networks is often challenging due to the competing objective functions of the generator and discriminator. Recently, new objective functions and training heuristics have greatly improved the training process of GANs (Arjovsky et al. 2017; Berthelot et al. 2017).

GANs follow a different training scheme from other stochastic reconstruction methods (Sect. 1). There are two phases in GAN-based reconstruction: training and generation. Training is expensive, requiring modern graphics processing units (GPU) and for three-dimensional datasets large GPU memory. Parallelization of the training process across numerous GPUs reduces time for training the network. Nevertheless, finding a set of hyper-parameters, that is, a network architecture (number of filters, types, order of layers and activation functions) that leads to the desired quality can require significant trial and error.

The second phase of GAN-based reconstruction, the generation of individual samples, is extremely fast. All operations in the generator network can be represented as matrix–vector operations which are executed efficiently on modern GPU systems and take on the order of seconds for modern GPUs, as shown later in this paper.

3 Dataset

The sample used in this study is an oolitic limestone of Jurassic age (169–176 million years). The spherical to ellipsoidal grains consist of 99.1% calcite and 0.9% quartz (Menke et al. 2017). Inter- and intra-granular porosity can be observed, as well as significant amounts of unresolved sub-resolution microporosity. This is characterized by the various shades of gray in individual grains, where the interaction of sub-resolution porosity with X-rays penetrating the sample during imaging leads to an increase in intermediate gray-level values (Fig. 1). The sample was imaged using a Zeiss XRM 510 with a voxel size of 27.8 μm. The size of the image domain after resampling to 8 bit resolution is 900^3 voxels. We subdivide the original image into a training set of non-overlapping 5832 images at a size of 64^3 voxels. We define a sequential randomized pass over the full training set as an epoch. Evaluation of the effective properties is performed at larger image sizes than the training images to judge whether the GAN is able to generalize to larger domains. To evaluate the reconstruction

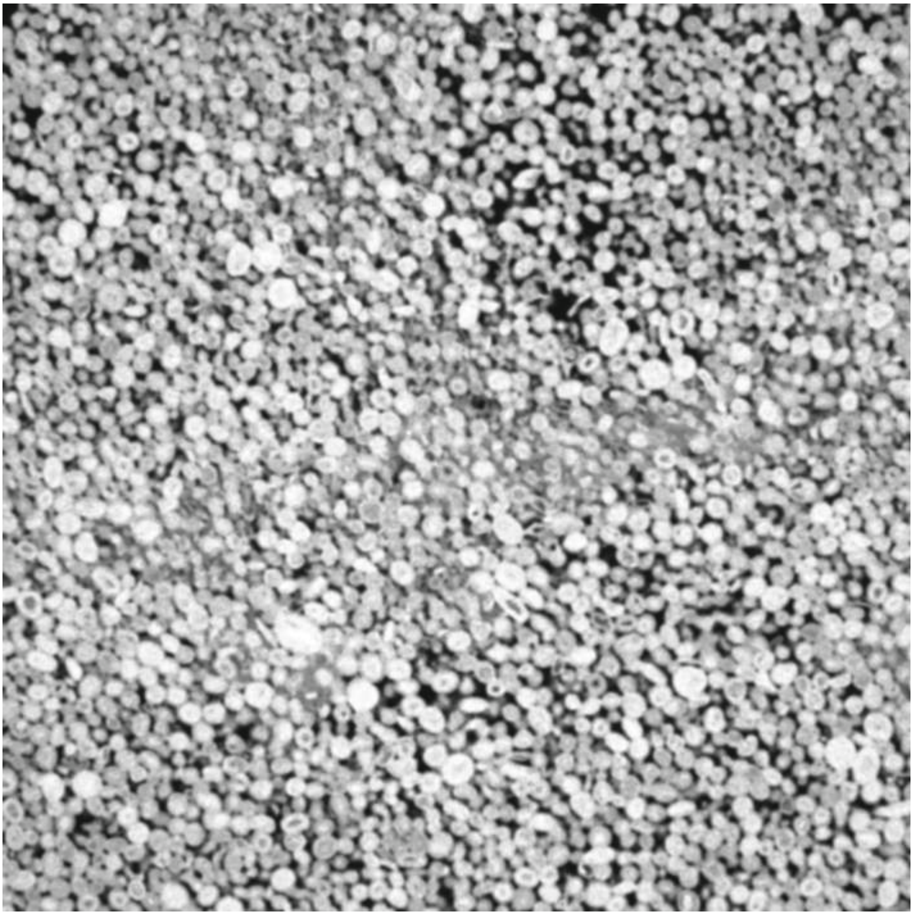

Fig. 1 Two-dimensional gray-level cross section of the three-dimensional micro-CT image of the studied oolitic Ketton limestone sample. The image has a size of 900^3 voxels and was acquired with a voxel size of 27.8 μm. Histogram equalization was applied to the image prior to its use as a training image

quality of the GAN model, we randomly extract 64 images at a size of 200^3 voxels with no overlap from the original training image (Fig. 1) which we refer to as the validation set. A synthetic validation set was created by sampling 64 images at a size of 200^3 voxels from the trained GAN model. To perform numerical computation of the effective permeability as well as measure the two-point correlation function, all images of the synthetic and original Ketton validation set were segmented using Otsu thresholding (Otsu 1975). Minkowski functionals were evaluated for the unsegmented validation sets.

3.1 Neural Network Architecture and Training

Radford et al. (2015) proposed to remove fully connected layers in the input and output of the generator network. They represent the input layer for the latent random vector by a reshaping operation, followed by a stack of strided convolutional layers. Jetchev et al. (2016) introduced the SGAN architecture where the input latent vector has spatial dimension and is immediately followed by a set of convolution operations. This allows images to be generated that are larger than the training images. They also provide evidence that sampling using the SGAN network architecture represents a stationary, ergodic and strongly mixing stochastic process. Our generator architecture represents a fully convolutional network without reshaping operations. The fully convolutional nature of the generator allows us to create images of arbitrary size by providing latent random vectors with larger spatial dimensionality, e.g., $\mathbf{z} \sim \mathcal{N}(0, 1)^{d \times m \times n \times o}$. During training, m, n and o are of size one, which results in an image of 64^3 voxels. For image generation, m, n and o may be of any integer size. The main difference to the SGAN architecture of Jetchev et al. (2016) is therefore that at training time the input random vector has a spatial dimension of one and the output of the discriminator is a single scalar value.

In Fig. 2, we show an example of a convolution and transposed convolution operation for the two-dimensional case. The convolution is performed by sliding a filter kernel w_i (Eq. 6) over the input feature map x_i (Eq. 7) (Dumoulin and Visin 2016). We rewrite this as an efficient matrix vector operation (Eq. 8) by unrolling the discrete convolution:

$$\mathbf{W} = \begin{pmatrix} w_0 & w_1 & w_2 & 0 & w_3 & w_4 & w_5 & 0 & w_6 & w_7 & w_8 & 0 & 0 & 0 & 0 & 0 \\ 0 & w_0 & w_1 & w_2 & 0 & w_3 & w_4 & w_5 & 0 & w_6 & w_7 & w_8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & w_0 & w_1 & w_2 & 0 & w_3 & w_4 & w_5 & 0 & w_6 & w_7 & w_8 & 0 \\ 0 & 0 & 0 & 0 & 0 & w_0 & w_1 & w_2 & 0 & w_3 & w_4 & w_5 & 0 & w_6 & w_7 & w_8 \end{pmatrix} [4 \times 16] \qquad (6)$$

The input image $\mathbf{x}$, in this case a single-channel 4×4 image, and the output $\mathbf{y}$ are represented as one-dimensional vectors:

$$\mathbf{x}\ [16 \times 1],\ \ \mathbf{y}\ [4 \times 1] \qquad (7)$$

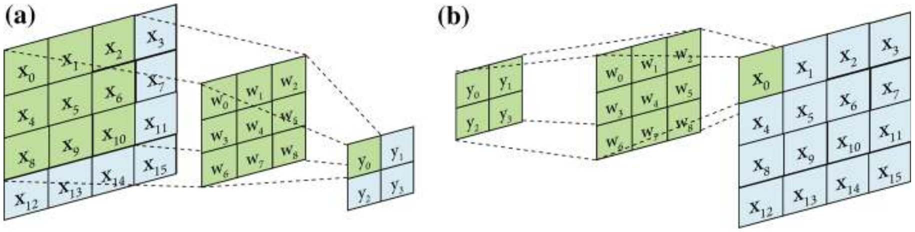

Fig. 2 Example of a discrete convolution (**a**) and equivalent transposed convolution operation (**b**) for a 3×3 filter kernel size applied to a 4×4 feature map. The active regions to compute the output value are shaded green

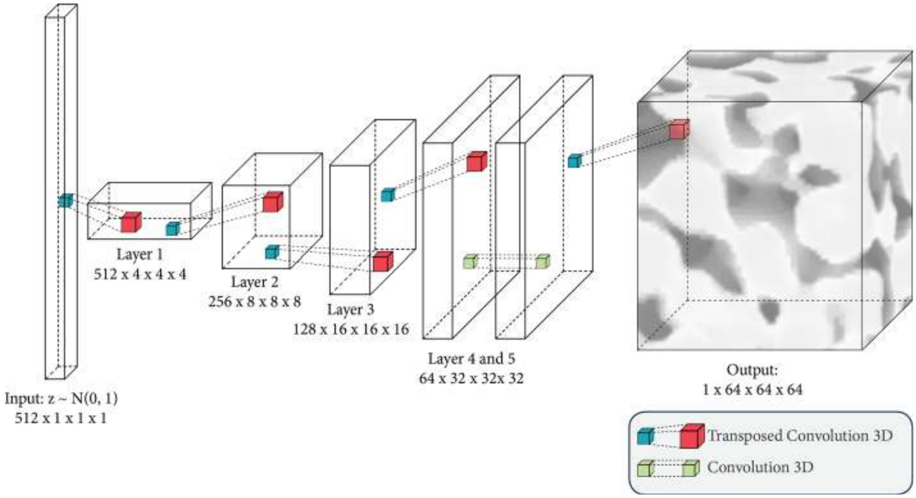

Fig. 3 Architecture of the neural network used to represent the generator function $G_\theta(\mathbf{z})$. The latent vector $\mathbf{z}$ is passed through a fully convolutional feed-forward neural network. Transposed convolution operations upsample the image in each layer. A single convolutional layer is introduced prior to the final network layer to reduce artifacts due to upsampling using transposed convolution

This allows us to perform the discrete convolution:

$$\mathbf{W} * \mathbf{x} = \mathbf{y} \tag{8}$$

and we can define the transpose operation:

$$\mathbf{W}^{\mathrm{T}} * \mathbf{y}' = \mathbf{x}' \tag{9}$$

where $\mathbf{W}$, $\mathbf{x}$, $\mathbf{y}$, $\mathbf{x}'$ and $\mathbf{y}'$ are defined according to Eqs. (6) and (7). For each convolutional layer of the network, the input features are convolved with a number of independent filter kernels $\mathbf{W}$.

The generator consists of a series of three-dimensional transposed convolutions. In each layer, the number of weight kernels is reduced by a factor of $\frac{1}{2}$. Before the final transposed convolution, we add an additional convolutional layer (Fig. 3). Each layer in the network except the last is followed by a batch normalization (Ioffe and Szegedy 2015) and a leaky rectified linear unit (LeakyReLU) activation function. The final transposed convolution in the generator is followed by a hyperbolic tangent activation function (Tanh) (LeCun et al. 1998). A representation of each activation function used in the network is shown in Fig. 4.

We represent the discriminator as a convolutional classification network with binary output using as input the real samples of the 64^3 voxel training set (Label 1) and synthetic realizations of equal size created by the generator (Label 0). Each layer in the network consists of a three-dimensional convolution operation followed by batch normalization and a LeakyReLU activation function. The final convolutional layer outputs a single value between 0 and 1 (Sigmoid activation) which corresponds to the probability that the input image belongs to the original training set or in other words that it is a real image.

We distinguish two sets of parameters for training: The set of weights of a network comprises the adjustable parameters of the filter kernels for convolutional and neurons for linear network layers. The so-called hyper-parameters define the network architecture and training scheme, e.g., the number of filters per layer, the number of convolutional layers or learning rates. A chosen set of hyper-parameters defines different networks with their

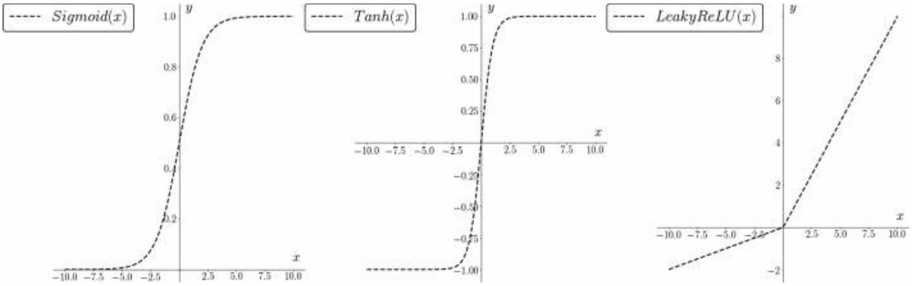

Fig. 4 Activation functions used in the generator and discriminator networks

own weights (parameters) which are adapted using a mini-batch gradient descent method at training time.

In total, 8 models have been trained on the Ketton image dataset. The main hyper-parameters that were varied for each model are the number of filters in the generator and discriminator, N_{GF} and N_{DF}, respectively, as well as the number of convolutional layers before the final transposed convolution in the generator. The dimensionality of the latent random vector $\mathbf{z}$ was kept constant at a size of $512 \times 1 \times 1 \times 1$. Learning was performed by stochastic gradient descent using the ADAM optimizer with momentum constants $\beta_1 = 0.5$, $\beta_2 = 0.999$ and a constant learning rate of 2×10^{-4}. Network training was performed on eight NVIDIA K40 GPUs using a mini-batch size of 64 images and the total run time of each training run is 8 h.

To train the pair of networks $G_\theta(\mathbf{z})$ and $D_\omega(\mathbf{x})$, we make use of two heuristic stabilization methods. First, Gaussian noise ($\mu = 0, \sigma = 0.1$) is added to the input of the discriminator which is annealed linearly over the first 300 epochs of training. A theoretical analysis of why adding Gaussian noise helps to stabilize GAN training was performed by Kaae Sønderby et al. (2016). In addition, we make use of a second stabilization method called label switching. Label switching represents a heuristic stabilization method with the aim of weakening the discriminator during the early stages of training. This heuristic stabilization method is performed by training the discriminator every N steps for one step with switched labels of the input real and generator simulated images; a real image is expected to be labeled as false and generated images as real. This corresponds to switching the expected labels of the input image mini-batches in Eq. (5).

Among the eight models tested, the network architecture generating realizations with the smallest mismatch with respect to the evaluated statistical and physical properties is presented in Table 1. The presented model has hyper-parameters of $N_{\mathrm{DF}} = N_{\mathrm{GF}} = 64$. Training was stopped after 170 epochs, i.e., full iterations of the training set of images. The generator consists of 27.9×10^6 adjustable parameters and 11.0×10^6 parameters for the discriminator. Visual inspection of the generated images and empirical computation of morphological and statistical properties were used as a measure for reconstruction performance at each iteration.

After training, the generator was used to create 64 reconstructions at a size of 200^3 voxels by sampling from the noise prior $\mathbf{z}$ (Eq. 1) and performing the mapping from the latent space to the image space (Eq. 2). Figure 5 shows slices through 32 non-overlapping sub-domains of the Ketton validation set and slices through 32 synthetic validation samples generated by the GAN model. The samples shown represent a random set of the generator output and were not selected by hand for their visual or statistical quality. The following sections present the a posteriori calculations of statistical, morphological and effective properties of these

Table 1 Architecture of the generator and discriminator networks

Layer	Type	Filters	Kernel	Stride	Padding	Batch normalization	Activation
Generator							
1	ConvTransp3D	512	$4 \times 4 \times 4$	1	0	Yes	LeakyReLU
2	ConvTransp3D	256	$4 \times 4 \times 4$	2	1	Yes	LeakyReLU
3	ConvTransp3D	128	$4 \times 4 \times 4$	2	1	Yes	LeakyReLU
4	ConvTransp3D	64	$4 \times 4 \times 4$	2	1	Yes	LeakyReLU
5	Conv3D	64	$3 \times 3 \times 3$	1	1	Yes	LeakyReLU
6	ConvTransp3D	1	$4 \times 4 \times 4$	2	1	No	Tanh
Discriminator							
1	Conv3D	64	$4 \times 4 \times 4$	2	1	No	LeakyReLU
2	Conv3D	128	$4 \times 4 \times 4$	2	1	Yes	LeakyReLU
3	Conv3D	256	$4 \times 4 \times 4$	2	1	Yes	LeakyReLU
4	Conv3D	512	$4 \times 4 \times 4$	2	1	Yes	LeakyReLU
5	Conv3D	1	$4 \times 4 \times 4$	1	0	No	Sigmoid

The generator is a fully convolutional version of a DCGAN (Radford et al. 2015) with one additional convolution layer prior to the final transposed convolution. LeakyReLU activation functions were used for all layers except the last

64 synthetic validation images in comparison to the extracted validation set of the original Ketton image (Fig. 5).

3.2 Two-Point Probability Functions

The two-point probability functions $S_2(\mathbf{r})$ allow the first- and second-order moments of a microstructure to be characterized. We define the isotropic non-centered two-point probability function $S_2(\mathbf{r})$ as the probability that two arbitrary points separated by a distance $\|\mathbf{r}\|$ are located in the same phase, i.e., grain or void phase of the microstructure. While $S_2(\mathbf{r})$ may be defined for both phases of a porous medium, we compute the two-point probability function with respect to the pore phase only.

$$S_2(\mathbf{r}) = \mathbf{P}(\mathbf{x} \in P, \mathbf{x} + \mathbf{r} \in P) \quad \text{for } \mathbf{x}, \mathbf{r} \in \mathbb{R}^3 \tag{10}$$

$S_2(0)$ is equal to the porosity of the porous medium. Stabilization of $S_2(\mathbf{r})$ occurs around a value of ϕ^2 as the distance tends toward infinity. In addition, the specific surface area S_V can be determined from the slope of the two-point probability function at the origin $S_V = -4S_2'(0)$ (Berryman 1987).

We calculate $S_2(\mathbf{r})$ numerically using the lattice point algorithm (Jiao et al. 2008). Figure 6 shows the directional two-point probability function for 64 200^3 voxel sub-domains of the original Ketton validation set (gray) and the GAN-generated realizations (red). We find that the 64 GAN-generated realizations lie within the standard deviation of the experimental $S_2(\mathbf{r})$ computed for the 64 original Ketton images.

Due to the ellipsoidal nature of the grains found in the Ketton limestone, a significant oscillation can be observed in all three orthogonal directions. This “hole effect” is characteristic of periodic media (Torquato and Lado 1985). The hole effect found in the training image dataset is reproduced by the samples generated by the GAN model, indicating the preservation of periodic features in the pore microstructure of the synthetic images.

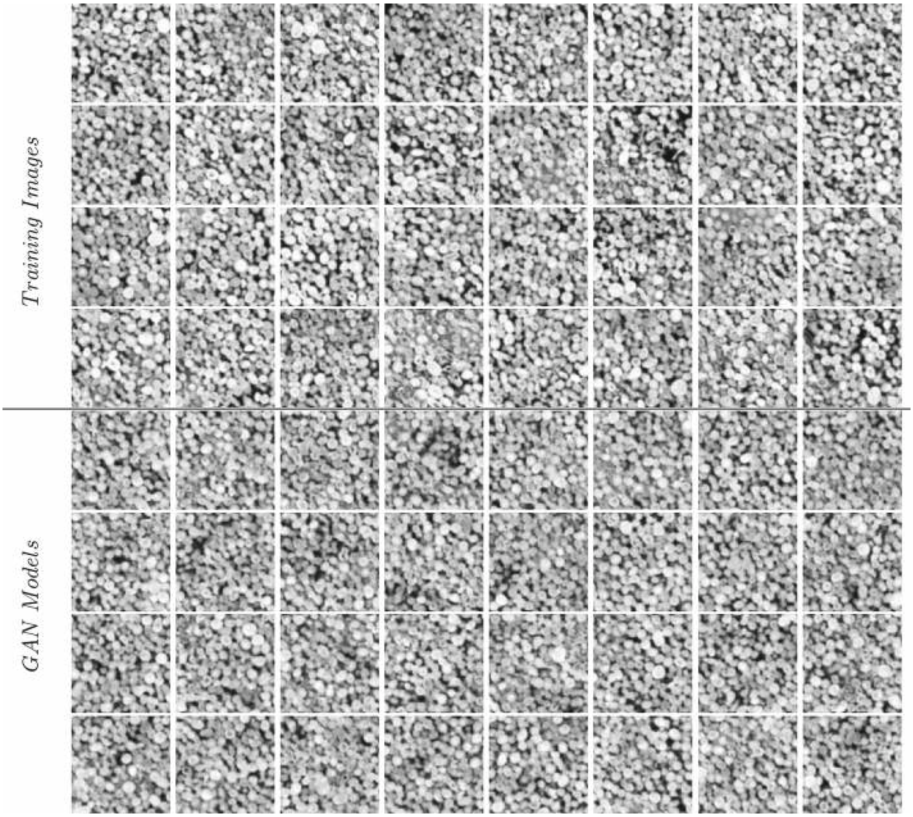

Fig. 5 Cross sections of the 200^3 voxel sub-domains of the Ketton micro-CT image (top) and synthetic realizations obtained from the trained generator of the generative network (bottom)

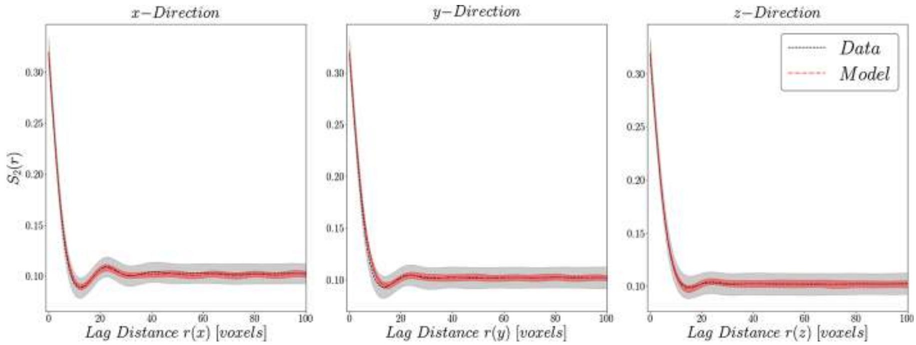

Fig. 6 Comparison of the two-point probability function $S_2(\mathbf{r})$ measured along the Cartesian axes for Ketton image sub-domains and GAN-generated realizations. $S_2(\mathbf{r})$ was measured on images after thresholding using Otsu's method. Gray and red shaded areas, respectively, show the variation around the average behavior $(\mu \pm \sigma)$ of 64 images of the Ketton image and GAN-generated validation set

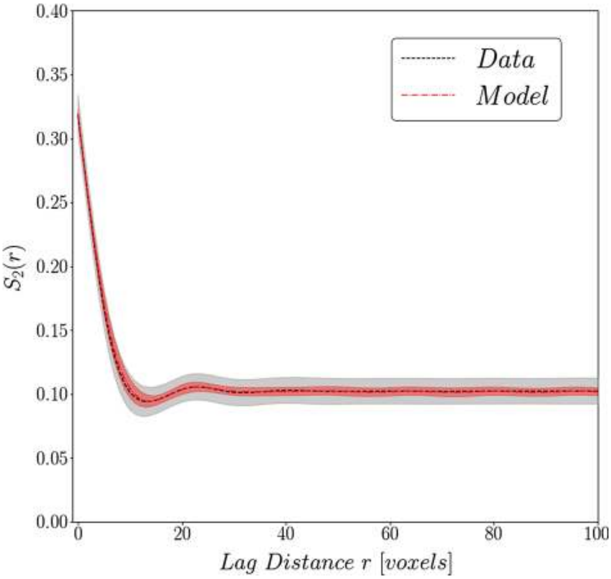

Fig. 7 Radial average of the average two-point probability function $S_2(\mathbf{r})$) for 64 dataset sub-domains and GAN-generated images. Excellent agreement of the average behavior can be observed (dashed line), whereas a lower variation around the mean behavior can be observed for the GAN-generated images

Good agreement between the real and synthetic microstructures can be observed for the radial averaged two-point probability function (Fig. 7). For both the radial averaged and directional estimates of $S_2(\mathbf{r})$, a tight clustering around the mean can be observed, whereas the real porous medium shows a larger degree of variation around the mean.

3.3 Minkowski Functionals

To evaluate the ability of the trained GAN model to capture the morphological properties of the studied Ketton limestone, we compute four integral geometric properties that are closely related to the set of Minkowski functionals as a function of the image gray value.

For any n-dimensional body we can define $n + 1$ Minkowski functionals to characterize morphological descriptor of the grain–pore body structures (Mecke 2000). The Minkowski functional of zeroth order is equivalent to the porosity of a porous medium and defined as:

$$\phi = M_0 = \frac{V_{\text{pore}}}{V} \tag{11}$$

where V_{pore} corresponds to the pore volume and V to the bulk volume of the porous medium.

We measure the specific surface area S_V defined as an integral geometric relationship:

$$S_V = \frac{M_1}{V} = \frac{1}{V}\int \mathrm{d}S \tag{12}$$

where M_1 is the Minkowski functional of first order. In three dimensions, M_1 corresponds to the surface area of the pore–grain interface. Both S_V and ϕ can be obtained by estimation of the two-point probability function $S_2(\mathbf{r})$ (Sect. 3.2). The specific surface area S_V has dimensions of $\frac{1}{\text{length}}$ and its inverse can be used to define a characteristic length scale of the porous medium.

The Minkowski functional of order 2, the integral of mean curvature, M_2, can be related to the shape of the pore space due to its measure of the curvature of pore–grain interface. We use a bulk volume average of the specific surface area defined as:

$$\kappa_V = \frac{M_2}{V} = \frac{1}{2V} \int \left(\frac{1}{r_1} + \frac{1}{r_2} \right) \mathrm{d}S \tag{13}$$

where r_1 and r_2 are the principal radii of curvature of the pore–grain interface.

The Euler characteristic, χ_V, is a measure of connectivity that is proportional to the dimensionless third-order Minkowski functional M_3:

$$\chi_V = \frac{M_3}{4\pi V} = \frac{1}{4\pi V} \int \frac{1}{r_1 r_2} \mathrm{d}S \tag{14}$$

We evaluate these four image morphologic properties at each of the 256 gray-level values of the 200^3 voxel Ketton image sub-domains and the GAN-generated realizations. This allows us to describe the porous medium as a set of characteristic functions dependent on a global truncation value ρ for each of the four Minkowski functionals (Schmähling 2006; Vogel et al. 2010). To compute the four properties at each threshold level ρ, the publicly available microstructure analysis software library Quantim was used (Vogel 2008).

Figure 8 compares these four estimated properties as a function of the image threshold value for the Ketton image (gray) and the samples generated by the GAN model (red). The shaded regions correspond to the variation around the mean $\mu \pm \sigma$ for both synthetic and real image datasets. The same 64 samples of the validation set used in the evaluation of the two-point probability function have been used for this analysis. Additionally, the vertical dashed lines represent the range of the threshold values obtained by Otsu's method when applied to the individual images. This allows an estimate of the error region that is significant when introducing a thresholding method based on a global truncation value such as Otsu's method.

Our analysis of the GAN-based models shows excellent agreement for the porosity $\phi(\rho)$, specific surface area $S_V(\rho)$ and integral of mean curvature $\kappa_V(\rho)$ as a function of the threshold value ρ. For these three properties, a low error is introduced when applying global thresholding. The fourth property, the specific Euler characteristic, $\chi_V(\rho)$, shows an error of 20% in the range of global thresholding values with good agreement outside this range. This implies that care must be taken when segmenting an image—real or generated—to preserve the connectivity of the pore space. As for the covariances, we also observe that the scatter produced by the GAN simulations is less than the scatter of the training set.

3.4 Permeability and Velocity Distributions

To validate GAN-based model generation for uncertainty evaluation and numerical computations, it is key that the generated samples capture the relevant physical properties of the porous media that the model was trained on. The permeability and, moreover, the local velocity distributions represent the key properties of the porous medium (Menke et al. 2017).

To evaluate the ability of GAN-based models to capture the permeability and in situ velocity distributions of the Ketton training images, we solve the Stokes equation on a segmented representation of each of the 64 Ketton sub-domains and 64 synthetic pore representations created by the GAN model. The segmented representations used to estimate the two-point probability functions were reused for this evaluation. A finite difference-based method adapted for binary representations of voxel-based pore representations was used to compute the effective permeability from the derived velocity field (Mostaghimi et al. 2013). The effective permeability was computed in the three Cartesian directions.

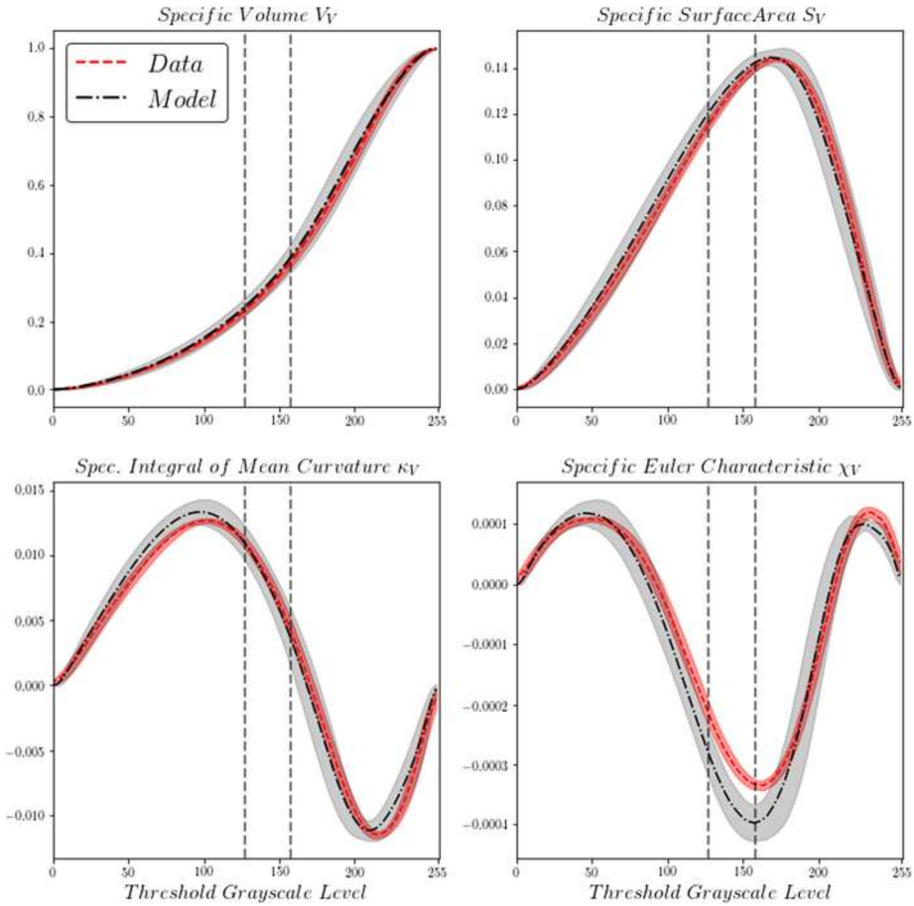

Fig. 8 Four Minkowski functionals as a function of the segmentation threshold. The shaded regions show the variation of the properties around the mean $\mu \pm \sigma$. Vertical dashed lines show the region of segmentation thresholds obtained by applying Otsu's method

$$\nabla \cdot \mathbf{v} = 0 \tag{15a}$$

$$\mu \nabla^2 \mathbf{v} = \nabla p \tag{15b}$$

We present the resulting distribution of estimated permeability values as a function of the effective porosity:

$$\phi_{\text{eff}} = \frac{V_{\text{flow}}}{V} \tag{16}$$

where V_{flow} is the volume of the connected porosity.

Our results (Figs. 9, 10) show that the GAN model generates stochastic reconstructions that capture the average permeability of the original training image at a scale of 200^3 voxels, with the majority of samples closely centered around the average effective permeability of the Ketton subsets.

The velocity distributions of the numerical simulations performed on the Ketton validation dataset and generated realizations were normalized by the average cell-centered velocity following the approach of Alhashmi et al. (2016) and a histogram with 256 logarithmically spaced bins in a range from 10^{-4} to 10^2 for each simulation was obtained.

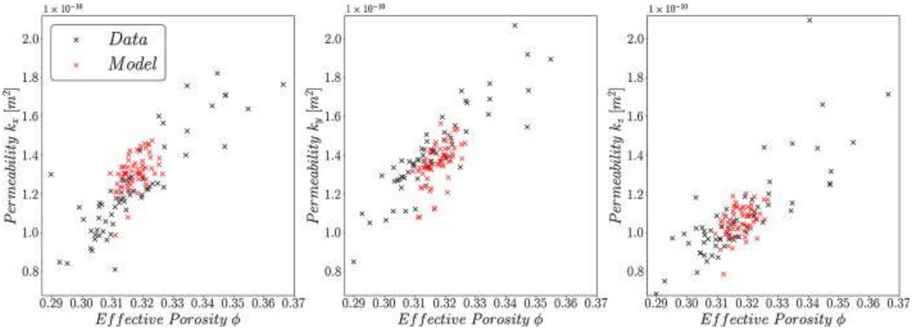

Fig. 9 Directional permeability computed on the validation dataset (64 images with 200^3 voxels) extracted from the original Ketton limestone micro-CT dataset and realizations obtained from the GAN model. Values of permeability obtained from the synthetic images are tightly clustered around the mean of the original dataset

Fig. 10 Averaged permeability for the original image datasets and synthetic realizations obtained from the GAN model

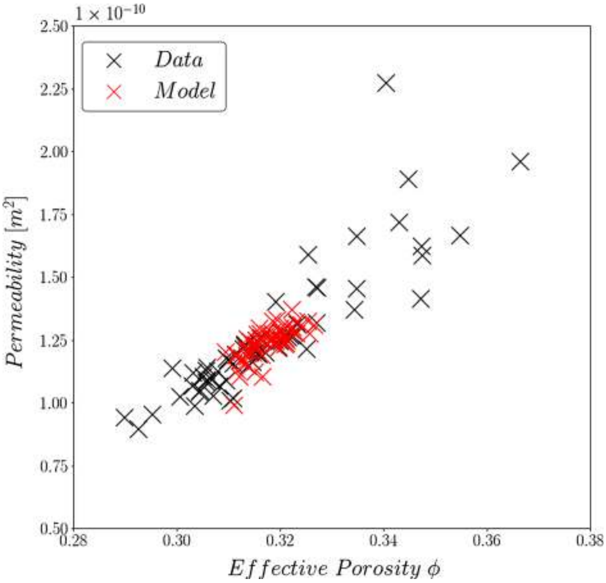

Figure 11 shows the per-bin arithmetic average of the bin frequencies and a bounding region of one standard deviation $\mu \pm \sigma$ as the shaded area. Due to the high range of velocities spanning six orders of magnitude, the x-axis is represented in logarithmic scaling.

Visually, the distributions of the generated samples and Ketton sub-domains are nearly equivalent with minor deviations in the frequency of the very high and very low velocities. For the GAN model, low velocities are more abundant than in the original image, whereas the opposite is true for high velocities.

To evaluate whether the velocity distributions obtained from numerical simulation of flow for the GAN-generated images are statistically similar to distributions representative of the original image dataset, we perform a two-sample Kolmogorov–Smirnov test. The null hypothesis H_0 states that two samples are of the same underlying distribution. Define $D_{n,m}$ as:

$$D_{n,m} = \sup_x |F_{1,n}(x) - F_{2,m}(x)| \tag{17}$$

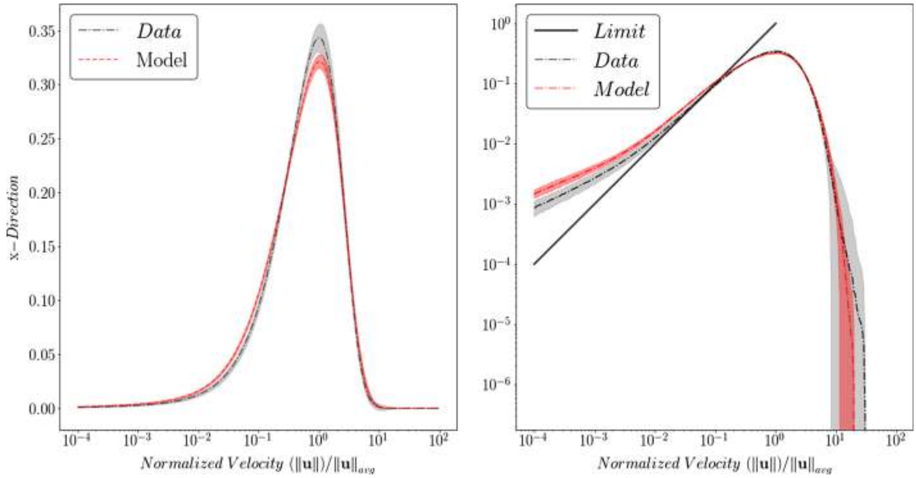

Fig. 11 Comparison of probability density functions of the magnitude of velocity extracted from the centers of voxels in the pore space divided by the average flow velocity plotted on semilogarithmic (left) and double-logarithmic axes (right). The combinations of 64 simulations on sub-domains obtained from the original dataset and 64 generated realizations of the GAN model are shown. Shaded regions highlight the variation around the mean of all simulations $\mu \pm \sigma$. The solid line shows the homogeneous limit velocity distribution for a single capillary tube

Table 2 Results of the two-sample Kolmogorov–Smirnov test for equality of velocity distributions computed on the image dataset and generated realizations

Direction	$D_{n,m}$	$D_{0.05}$
x	0.09	0.12
y	0.09	
z	0.07	

The null hypothesis of distributional equality is to be accepted at a significance level of $\alpha = 0.05$ for all three directional velocity distributions

and the null hypothesis H_0 is rejected if

$$D_{n,m} > c(\alpha)\sqrt{\frac{n+m}{nm}} \tag{18}$$

where n and m are the sample sizes, respectively, and $c(\alpha) = \sqrt{-\frac{1}{2}\ln(\frac{\alpha}{2})}$. All tests were performed at a significance level of $\alpha = 0.05$ for the per-bin average velocity distributions presented in Fig. 11 (dashed curves).

For all three directions, the null hypothesis can be accepted at the 5% significance level based on the $D_{0.05}$ statistic, giving evidence to the visual similarity between the velocity distributions of the real Ketton images and their synthetic counterparts (Table 2).

4 Discussion

We have presented the results of training a generative adversarial network on a micro-CT image of the oolitic Ketton limestone. The image morphological properties were evaluated as a function of the image threshold level and it was shown that the generated images capture the textural features of the original training image. Two-point statistics and effective properties computed on segmented representations of the individual sub-domains have also shown excellent agreement between the realizations generated by the GAN model and subsets of the Ketton image. Nevertheless there remain a number of open questions that need to be addressed.

The predicted statistical and morphological properties have shown a tight bound around the average behavior of the training image. This indicates that there is less variation in the generated samples than in the training samples. This behavior can have a number of origins.

The training images can be regarded as samples of the unknown multivariate pdf $p_{\text{real}}(\mathbf{x})$, which is likely to be multimodal. The original formulation of the GAN objective function (Goodfellow et al. 2014) has been shown to lead to unimodal pdfs, even if the training set pdf itself is multimodal (Goodfellow 2017). The behavior of a generator to represent multimodal pdfs by a pdf with fewer modes is called mode collapse (Goodfellow 2017). This behavior may occur due to the fact that there is no incentive for diversity in GAN training.

Visually the images generated by the presented GAN model are nearly indistinguishable from their real counterparts (Fig. 5). Minkowski functionals and statistical parameters allow us to perform an evaluation of the reconstruction quality. Nevertheless, this does not rule out the fact that the generator may be memorizing the training set, show mode collapse behavior or output a low diversity of synthetic samples. A generator showing one or more of these behaviors will falsely indicate low errors in the Minkowski functionals, statistical and effective properties.

By visual inspection of the validation set generated by the GAN model, no evidence of identical or repeated features in the generated images could be found. Following the approach by Radford et al. (2015), we perform an interpolation between two points in the latent space $\mathbf{z}$:

$$\mathbf{z}_{\text{start}}, \mathbf{z}_{\text{end}} \in \mathcal{N}(0,1)^{512\times1\times1\times1}, \ \beta \in [0,1] \tag{19a}$$

$$\mathbf{z}_{\text{inter}} = \beta\, \mathbf{z}_{\text{start}} + (1-\beta)\, \mathbf{z}_{\text{end}} \tag{19b}$$

where β is a range of numbers from zero to one. This provides evidence of the generator's ability to learn meaningful representations and shows the absence of memorization.

The smooth transition between the starting image $G_\theta(\mathbf{z}_{\text{start}})$ and the endpoint $G_\theta(\mathbf{z}_{\text{end}})$ shown in Fig. 12 indicates that the generator has not memorized the training set and has instead learned a lower-dimensional representation $\mathbf{z}$ that results in meaningful features of the pore–grain microstructure. Definition of GAN training objectives compatible with high-diversity

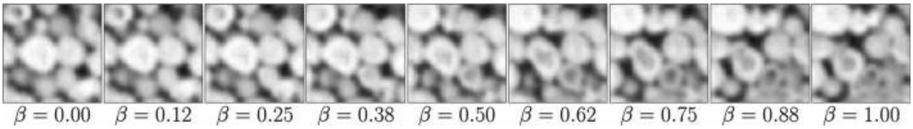

Fig. 12 Interpolation in the latent space $\mathbf{z}$ performed for the evaluated generator G_θ shows a smooth interpolation between the start latent random vector $\mathbf{z}_{\text{start}}$ ($\beta = 1$) and the end point $\mathbf{z}_{\text{end}}$ ($\beta = 0$). An example feature of this can be seen by a bright calcite grain present in the left most image slowly being transformed into a spherical grain with significant microporosity

samples showing no mode collapse and stable training remains an open problem. Che et al. (2016) have presented a summary of recent advances to counteract mode collapse and have proposed a regularization method to improve GAN output variety. Reformulations of the GAN training criterion (Eq. 4) based on the Wasserstein distance (WGAN-GP) (Gulrajani et al. 2017) and other training approaches to GANs such as EBGAN (Zhao et al. 2016) or DRAGAN (Kodali et al. 2017) show the ability to model multimodal densities and allow stable training.

It is important to note that the output of the generator is parameterized by the stochastic latent random vector and can be optimized due to the differentiable nature of the generative neural network. This is a powerful concept that has been leveraged in a number of applications in computer vision. Inpainting is the task of creating semantically meaningful content where missing data exist. Commonly this is a task performed where objects are occluded or only partially visible. In microstructural applications and often at larger geological scales, lower-dimensional information may be more readily available than acquiring a full three-dimensional image, e.g., thin sections of porous media. Constraining images to these data is referred to as conditioning and can be reformulated as an inpainting problem. Yeh et al. (2016) introduced a framework for inpainting using GANs where the latent random vector can be optimized with regard to a perceptual objective function determined by the discriminator and a mismatch between the observed data and the output of the generator. In other work, we have shown that the method of Yeh et al. (2016) can be applied and produces stochastic three-dimensional samples that honor the given two- and one-dimensional conditioning data (Mosser et al. 2018).

While the input and output to the GAN generator and discriminator are well defined, the interior mechanics of the neural network that result in high-quality reconstructions are not well understood. Rather than treating GANs as a black-box mechanism, it is of interest to evaluate the behavior of the generator and discriminator in more detail. In Fig. 13, we have extracted the generator's output after each layer's activation function (following the convolution operation and batch normalization).

Based on the consecutive upsampling of the noise prior **z** by each transposed convolution in the generator, we observe a multi-scale feature representation of the final image. Early layers, where the spatial dimensions of the images are small, can be related to global features in the generator output. The final layers create highly detailed representations of the structural features of the reconstructed images. This view of the generator's behavior also helps identify deficiencies in the network's architecture. In layers 3 and 4, we see repeated noise that appears to be following a grid like structure. This is due to the transposed convolutional operation and in parts is diminished by the additional convolution operation prior to the last upsampling operation. This could be alleviated by the use of other convolution-based upsampling layers

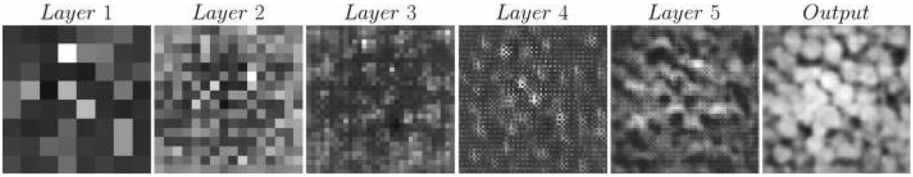

Fig. 13 Representations of the noise prior **z** as it is propagated through the generator G_θ. Each layer adds to a multi-scale reconstruction of the final image $G_\theta(\mathbf{z})$. The shallow layers 1, 2 and 3 introduce global features of the final image, whereas deeper layers add high-fidelity details to the output image. Significant noise is still present in layer 3 due to the use of transposed convolution operations, but reduced by the convolution in layer 5

such as the sub-pixel convolution operation (Shi et al. 2016) or interpolation upsampling (nearest neighbor, bilinear, trilinear).

The discriminator's role is simply to label images as real or "fake," but it also is a critical component in the ability of the generator to learn features in the original image space. The discriminator, in order to distinguish GAN-generated from real training images, needs to learn a unique set of features that distinguish real samples from fake ones. As such, for future work, it may be of interest to use a GAN trained discriminator for classification or feature extraction (Arora and Zhang 2017).

Nevertheless, we can perform a similar operation as for the generator and inspect some of the features learned by the discriminator. Figure 14 shows a set of 5 learned filters applied to an image of the Ketton training set. At shallow layers, we find that the discriminator has learned to identify the pore space (layer 1, second row) as well as a number of edges. Deeper layers in the network represent more abstract features, and after layer 2, no original feature of the pore space is distinguishable.

Considering that the samples used to evaluate the statistical and effective properties were not chosen by hand but represent a random group of generated images based on the GAN model, further improvement can be obtained in the reconstruction results. The discriminator may be used as an evaluation criterion for samples where higher values obtained from the discriminator $D(G_\theta(\mathbf{z}))$ indicate that the samples are closer to the real training image dataset. In this way, high-quality reconstructions may be "cherry-picked" by choosing representations that score values $D(\mathbf{x})$ close to one (real label) from a much larger set of reconstructions.

The computational effort to perform image reconstruction using GANs can be split into two parts: training time and generation time. The training time is the total time required to find a set of parameters of the generator that allows generation at sufficient image quality. We define generation time as the total time required to initialize a neural network and the associated parameters obtained during the training phase and the generation of the images by passing a latent random vector $\mathbf{z}$ through the generator to obtain an image $\mathbf{x} \sim G_\theta(\mathbf{z})$. To create one realization from a GAN, it is necessary to train the generator–discriminator pairing only once; therefore, training time is a fixed computational cost. Once trained, the generator can simply be reused for each new realization.

We have performed benchmarking of our GAN model in terms of the computational time. Training was performed on eight Nvidia K40 GPUs and the total training time was 8 h. We evaluate the generation time of 100 realizations based on this set of pre-trained parameters. Each benchmark consists of the following steps: initialization of the generator parameters, sampling and initializing a latent random vector in GPU memory and finally applying the generator to the latent random vector $\mathbf{x} \sim G_\theta(\mathbf{z})$ to create a realization with 450^3 voxels. When sampling 100 realizations the first step, the initialization of the pre-trained generator parameters, is only required once and is not repeated for subsequent sampling operations. We have repeated this benchmarking exercise ten times on an NVIDIA V100 GPU and have quoted the average total run times. Our benchmark shows that the average run time to perform sampling of 100 realizations with 450^3 voxels is 100 s.

The main limitations in computational effort come from two factors: the training time and available GPU memory. In the future, we expect the training time to decrease, due to greater performance of GPUs and development of novel GAN training methods that allow faster convergence. Furthermore GAN-based image synthesis for large spatial domains requires large amounts of GPU memory, for example reconstruction with 450^3 voxel requires more than 10 gigabytes of GPU memory.

Recently, a number of algorithms have been developed to perform high-quality reconstruction of porous media based on training images (Jiao et al. 2009; Zachary and Torquato

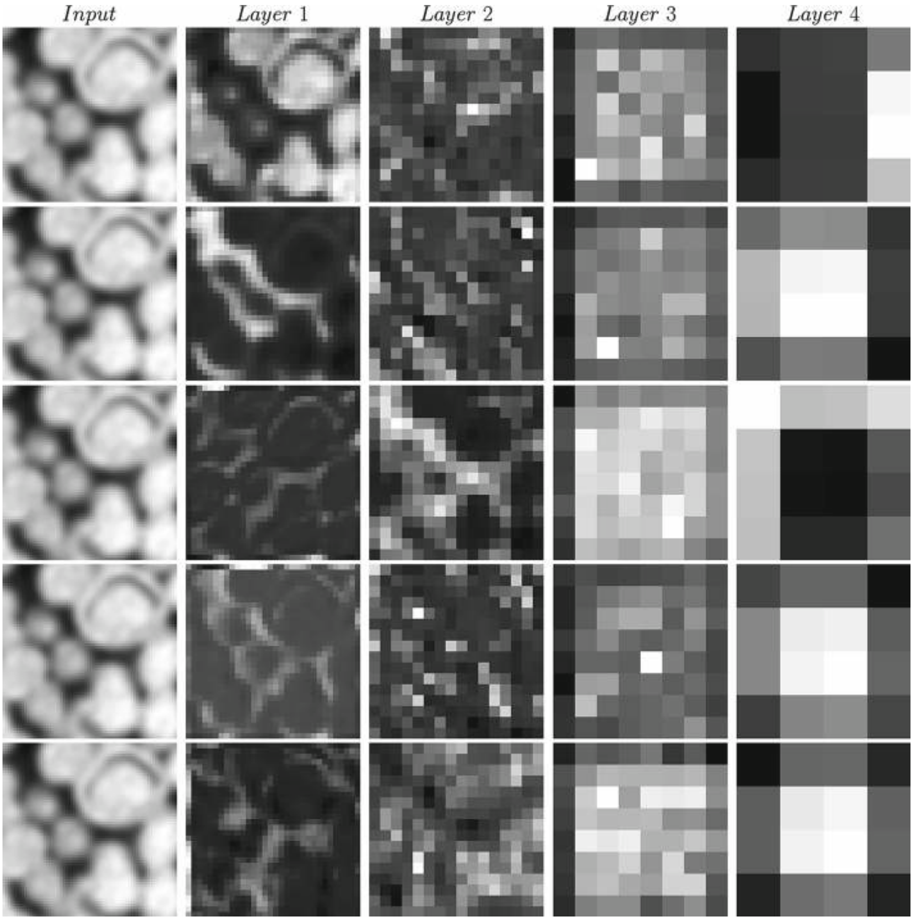

Fig. 14 An inspection of the behavior of the discriminator's learned feature representations for a training sample of the original Ketton training image. Each column represents one layer of the discriminator network. Each row represents one learned filter kernel in each layer applied to the input (leftmost column)

2011; Tahmasebi et al. 2017). While considering the resulting image quality to be equal, one possible differentiation of these methods is computational run time. Reported run times are heavily dependent on a number of criteria such as the simulated image size, software implementation or hardware used. Table 3 presents a summary of measured computational time reported for a number of recent reconstruction methods as well as their respective simulated image sizes.

Most methods reported in Table 3 incur a high computational cost per generated realization, with the exception of the method of Tahmasebi et al. (2017). We refer to these methods as proportional cost methods as the computational cost scales linearly with the number of created realizations. Training-based methods such as the presented GAN-based approach have a high initial computational cost due to the required training phase. Our method, once training is completed, has a very small generation time per realization. It is possible to determine an amortization time, when the use of one approach, considering all other factors equal, becomes beneficial.

Table 3 Comparison of reported computational run times of recent reconstruction methods

Authors	Method	Size [voxels^3]	Run time (× 1) (h)	Run time (× 100) (h)
Computational run time comparison				
Pant (2016)	Simulated annealing	300^3	22–47	2200
Čapek et al. (2009)	Simulated annealing	320^3	160–400	16,000
Tahmasebi et al. (2017)	Patch-based	$1000^2 \times 300$	0.1	11
Okabe and Blunt (2004)	MPS	150^3	12	1200
Current work	GAN	450^3	8	8.1

The computational cost for running 100 realizations is estimated based on the smallest reported run time for each algorithm. An unbiased comparison is difficult as each method has been evaluated on a different dataset and image sizes

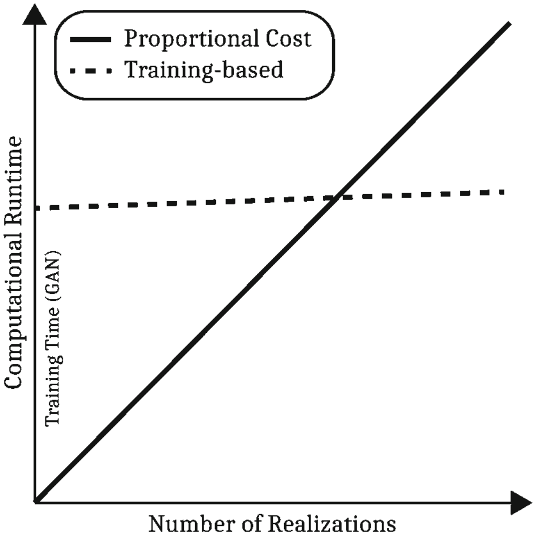

Fig. 15 Comparison of the computational cost for two stochastic reconstruction methods at fixed image size. Proportional cost-based methods are associated with a high run time per realization. Training-based methods, such as the presented GAN method, have a high initial computational cost due to their training phase and a small cost per generated realization afterward

Figure 15 presents a schematic comparison of the computational cost induced by different methods as a function of the number of realizations at a fixed image size. The amortization time, where the two curves intersect, corresponds to the number of realizations at which training-based methods, such as GANs, become faster.

5 Conclusions

We have presented a method to reconstruct microstructures of porous media based on gray-scale image representations of volumetric porous media. By creating a GAN-based model of an oolitic Ketton limestone, we have shown that GANs can learn to represent the statistical and effective properties of segmented representations of the pore space as well as their

Minkowski functionals as a function of the image gray level. In addition to the effective permeability which is associated with a global average of the velocity field, we show that the pore-scale velocity statistical distributions have been recovered by the synthetic GAN-based models. We highlight the roles of the discriminator and generator function of the GAN and show that the GAN learns a multi-scale representation of the pore space based on inference from a latent random prior. Large hyper-parameter searches involved in the deep neural network architectures and learning instabilities make the training of GANs difficult. The high computational cost involved in training GANs is made good use of for applications when very large or many stochastic reconstructions are required. The differentiable nature of the generative network parameterised by the latent random vector provides a powerful framework in the context of gradient-based optimization and inversion techniques. Future work will focus on creating GAN-based methodologies that ensure a valid representation of the underlying data distribution allowing application of GANs for uncertainty quantification and inversion of effective material properties.

Acknowledgements The authors would like to thank Hannah Menke (Imperial College London) for providing the Ketton image dataset. The authors thank H.J. Vogel of UFZ—Helmhotz Center for Environmental Research—for making the software library available for public use. O. Dubrule would like to thank Total S.A. for seconding him as a visiting professor at Imperial College London.

References

Alhashmi, Z., Blunt, M., Bijeljic, B.: The impact of pore structure heterogeneity, transport, and reaction conditions on fluid-fluid reaction rate studied on images of pore space. Transp. Porous Media **115**(2), 215–237 (2016)

Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein GAN. ArXiv e-prints (2017). arXiv:1701.07875

Arora, S., Zhang, Y.: Do GANs actually learn the distribution? An empirical study. ArXiv e-prints (2017). arXiv:1706.08224

Berg, C.F., Lopez, O., Berland, H.: Industrial applications of digital rock technology. J. Pet. Sci. Eng. **157**(Supplement C), 131–147 (2017). https://doi.org/10.1016/j.petrol.2017.06.074. http://www.sciencedirect.com/science/article/pii/S0920410517305600

Berryman, J.G.: Relationship between specific surface area and spatial correlation functions for anisotropic porous media. J. Math. Phys. **28**(1), 244–245 (1987)

Berthelot, D., Schumm, T., Metz, L.: BEGAN: boundary equilibrium generative adversarial networks. ArXiv e-prints (2017). arXiv:1703.10717

Blunt, M.J., Bijeljic, B., Dong, H., Gharbi, O., Iglauer, S., Mostaghimi, P., Paluszny, A., Pentland, C.: Pore-scale imaging and modelling. Adv. Water Resour. **51**, 197–216 (2013)

Caers, J.: Geostatistical reservoir modelling using statistical pattern recognition. J. Pet. Sci. Eng. **29**(3–4), 177–188 (2001)

Čapek, P., Hejtmánek, V., Brabec, L., Zikánová, A., Kočiřík, M.: Stochastic reconstruction of particulate media using simulated annealing: improving pore connectivity. Transp. Porous Media **76**(2), 179–198 (2009). https://doi.org/10.1007/s11242-008-9242-8

Che, T., Li, Y., Jacob, A.P., Bengio, Y., Li, W.: Mode regularized generative adversarial networks. ArXiv e-prints (2016). arXiv:1612.02136

Chiu, S.N., Stoyan, D., Kendall, W.S., Mecke, J.: Stochastic Geometry and its Applications. Wiley, Hoboken (2013)

Dumoulin, V., Visin, F.: A guide to convolution arithmetic for deep learning. ArXiv e-prints (2016). arXiv:1603.07285

Efros, A.A., Freeman, W.T.: Image quilting for texture synthesis and transfer. In: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, pp. 341–346. ACM (2001)

Flannery, B.P., Deckman, H.W., Roberge, W.G.: D'AMICO, K.L.: Three-dimensional X-ray microtomography. Science **237**(4821), 1439–1444 (1987)

Goodfellow, I.: NIPS 2016 tutorial: generative adversarial networks. ArXiv e-prints (2017). arXiv:1701.00160

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in Neural Information Processing Systems, pp. 2672–2680 (2014). http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf

Guardiano, F.B., Srivastava, R.M.: Multivariate Geostatistics: Beyond Bivariate Moments, pp. 133–144. Springer, Netherlands (1993). https://doi.org/10.1007/978-94-011-1739-5_12

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.: Improved training of wasserstein GANs. ArXiv e-prints (2017). arXiv:1704.00028

Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. ArXiv e-prints (2015). arXiv:1502.03167

Jetchev, N., Bergmann, U., Vollgraf, R.: Texture synthesis with spatial generative adversarial networks. ArXiv e-prints (2016). arXiv:1611.08207

Jiao, Y., Stillinger, F., Torquato, S.: A superior descriptor of random textures and its predictive capacity. Proc. Nat. Acad. Sci. **106**(42), 17634–17639 (2009)

Jiao, Y., Stillinger, F.H., Torquato, S.: Modeling heterogeneous materials via two-point correlation functions. II. Algorithmic details and applications. Phys. Rev. E **77**, 031,135 (2008). https://doi.org/10.1103/PhysRevE.77.031135

Joshi, M.: A Class of Stochastic Models for Porous Media. University of Kansas, Chemical and Petroleum Engineering (1974). https://books.google.co.uk/books?id=_zu8OwAACAAJ

Kaae Sønderby, C., Caballero, J., Theis, L., Shi, W., Huszár, F.: Amortised MAP inference for image super-resolution. ArXiv e-prints (2016). arXiv:1610.04490

Kodali, N., Abernethy, J., Hays, J., Kira, Z.: On convergence and stability of GANs. ArXiv e-prints (2017). arXiv:1705.07215

LeCun, Y., Bottou, L., Orr, G.B., Müller, K.R.: Efficient backprop. In: Neural Networks: Tricks of the Trade, pp. 9–50. Springer (1998)

Mariethoz, G., Caers, J.: Multiple-Point Geostatistics: Stochastic Modeling with Training Images. Wiley, Hoboken (2014)

Matheron, G.: Random Sets and Integral Geometry. Wiley series in probability and mathematical statistics: probability and mathematical statistics. Wiley (1975). https://books.google.co.uk/books?id=bgzvAAAAMAAJ

Mecke, K.R.: Additivity, convexity, and beyond: applications of Minkowski functionals in statistical physics. In: Statistical Physics and Spatial Statistics, pp. 111–184. Springer (2000)

Menke, H., Bijeljic, B., Blunt, M.: Dynamic reservoir-condition microtomography of reactive transport in complex carbonates: effect of initial pore structure and initial brine pH. Geochim. Cosmochim. Acta **204**, 267–285 (2017)

Mosser, L., Dubrule, O., Blunt, M.J.: Reconstruction of three-dimensional porous media using generative adversarial neural networks. Phys. Rev. E **96**, 043,309 (2017). https://doi.org/10.1103/PhysRevE.96.043309

Mosser, L., Dubrule, O., Blunt, M.J.: Conditioning of three-dimensional generative adversarial networks for pore and reservoir-scale models. ArXiv e-prints (2018). arXiv:1802.05622

Mostaghimi, P., Blunt, M.J., Bijeljic, B.: Computations of absolute permeability on micro-CT images. Math. Geosci. **45**(1), 103–125 (2013). https://doi.org/10.1007/s11004-012-9431-4

Okabe, H., Blunt, M.J.: Prediction of permeability for porous media reconstructed using multiple-point statistics. Phys. Rev. E **70**, 066,135 (2004). https://doi.org/10.1103/PhysRevE.70.066135

Okabe, H., Blunt, M.J.: Pore space reconstruction using multiple-point statistics. J. Pet. Sci. Eng. **46**(1–2), 121–137 (2005). https://doi.org/10.1016/j.petrol.2004.08.002

Okabe, H., Blunt, M.J.: Pore space reconstruction of vuggy carbonates using microtomography and multiple-point statistics. Water Resour. Res. **43**(12), 3–7 (2007). https://doi.org/10.1029/2006WR005680

Øren, P.E., Bakke, S.: Reconstruction of berea sandstone and pore-scale modelling of wettability effects. J. Pet. Sci. Eng. **39**(3), 177–199 (2003). https://doi.org/10.1016/S0920-4105(03)00062-7

Otsu, N.: A threshold selection method from gray-level histograms. Automatica **11**(285–296), 23–27 (1975)

Pant, L.M.: Stochastic characterization and reconstruction of porous media. Ph.D. thesis, University of Alberta (2016)

Quiblier, J.A.: A new three-dimensional modeling technique for studying porous media. J. Colloid Interface Sci. **98**(1), 84–102 (1984)

Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. ArXiv e-prints (2015). arXiv:1511.06434

Schmähling, J.: Statistical characterization of technical surface microstructure. Ph.D. thesis, Ruprecht-Karls-Universität, Heidelberg (2006)

Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A.P., Bishop, R., Rueckert, D., Wang, Z.: Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1874–1883 (2016)

Siddique, N., Salehi, A., Liu, F.: Stochastic reconstruction and electrical transport studies of porous cathode of li-ion batteries. J. Power Sources **217**(Supplement C), 437–443 (2012). https://doi.org/10.1016/j.jpowsour.2012.05.121. http://www.sciencedirect.com/science/article/pii/S0378775312010166

Singh, K., Menke, H., Andrew, M., Lin, Q., Rau, C., Blunt, M.J., Bijeljic, B.: Dynamics of snap-off and pore-filling events during two-phase fluid flow in permeable media. Sci. Rep. **7**(1), 5192 (2017)

Tahmasebi, P., Hezarkhani, A., Sahimi, M.: Multiple-point geostatistical modeling based on the cross-correlation functions. Comput. Geosci. **16**(3), 779–797 (2012)

Tahmasebi, P., Sahimi, M.: Reconstruction of three-dimensional porous media using a single thin section. Phys. Rev. E **85**(6), 066,709 (2012)

Tahmasebi, P., Sahimi, M.: Cross-correlation function for accurate reconstruction of heterogeneous media. Phys. Rev. Lett. **110**(7), 078,002 (2013)

Tahmasebi, P., Sahimi, M., Andrade, J.E.: Image-based modeling of granular porous media. Geophys. Res. Lett. **44**(10), 4738–4746 (2017). https://doi.org/10.1002/2017GL073938

Torquato, S.: Random Heterogeneous Materials: Microstructure and Macroscopic Properties, vol. 16. Springer, New York (2013). https://doi.org/10.1007/978-1-4757-6355-3

Torquato, S., Lado, F.: Characterisation of the microstructure of distributions of rigid rods and discs in a matrix. J. Phys. A Math. Gen. **18**(1), 141 (1985). https://doi.org/10.1088/0305-4470/18/1/025

Vogel, H.J.: Quantim (2008). http://www.quantim.ufz.de

Vogel, H.J., Weller, U., Schlüter, S.: Quantification of soil structure based on Minkowski functions. Comput. Geosci. **36**(10), 1236–1245 (2010). https://doi.org/10.1016/j.cageo.2010.03.007

Yeh, R.A., Chen, C., Yian Lim, T., Schwing, A.G., Hasegawa-Johnson, M., Do, M.N.: Semantic image inpainting with deep generative models. ArXiv e-prints (2016). arXiv:1607.07539

Yeong, C.L.Y., Torquato, S.: Reconstructing random media. Phys. Rev. E **57**, 495–506 (1998). https://doi.org/10.1103/PhysRevE.57.495

Zachary, C.E., Torquato, S.: Improved reconstructions of random media using dilation and erosion processes. Phys. Rev. E **84**(5), 056,102 (2011)

Zhao, J., Mathieu, M., LeCun, Y.: Energy-based generative adversarial network. ArXiv e-prints (2016). arXiv:1609.03126