Abstract
Due to the scarcity and vulnerability of physical rock samples, digital rock reconstruction plays an important role in the numerical study of reservoir rock properties and fluid flow behaviors. With the rapid development of deep learning technologies, generative adversarial networks (GANs) have become a promising alternative to reconstruct complex pore structures. Numerous GAN models have been applied in this field, but many of them suffer the unstable training issue. In this study, we apply the Wasserstein GAN with gradient penalty, also known as the WGAN-GP network, to reconstruct Berea sandstone and Ketton limestone. Unlike many other GANs using the Jeunes- Shannon divergence, the WGAN-GP network exhibits a stable training performance by using the Wasserstein distance to measure the difference between generated and real data distributions. Moreover, the generated image quality correlates with the discriminator loss. This provides us an indicator of the training state instead of frequently subjective assessments in the training of deep convolutional GAN (DCGAN) based models. An integrated framework is presented to automate the entire workflow, including training set generation, network setup, model training and synthetic rock validation. Numerical results show that the WGAN-GP network accurately reconstructs both Berea sandstone and Ketton limestone in terms of two-point correlation and morphological properties.

Introduction
Evaluation of rock properties and fluid flow behaviors in rock samples provides valuable information to set up parameters for large-scale reservoir simulation (Geiger et al., 2010; He, Sinan, et al., 2021; Hoteit & Firoozabadi, 2008a, 2008b). However, apart from prohibitive acquisition cost, physical rock samples are risky to be damaged during core flooding experiments. This makes it impractical to use a single rock for different research purposes. In addition, the heterogeneous nature of pore structures brings significant uncertainties through limited rock samples. Thus, it is necessary to have a sufficient number of similar pore structures to fully investigate rock and transport properties in a given rock. Such demands stimulate the development of digital rock reconstruction techniques.

Conventionally, there are three types of methods to create digital pore structures. The most straightforward approach is direct scanning of rock samples using sophisticated imaging tools, such as micro-CT
Recent advances in machine learning have inspired various environmental, geoscience, and energy applications, such as CO₂ leakage forecasting in saline aquifers (He, Zhu, et al., 2021a), history matching in field-scale simulations (Santoso et al., 2021), upscaling of discrete-fracture models (He et al., 2020; He, Santoso, et al., 2021), upscaling of rock fractures (He, Zhu, et al., 2021b), recovery factor prediction in gas-injected reservoirs (He, Qiao, et al., 2021), multi-component flash calculations (Li et al., 2019; Zhang et al., 2020) and recognizing and detecting fractures from real outcrop (Santoso et al., 2019). These successful implementations motivate us to explore the possible employment of machine learning on digital rock reconstructions. Generative Adversarial Networks (GANs), first proposed by Ian Goodfellow et al. (2014), have been extensively applied to pore-scale imaging and processing (Wang et al., 2021), such as image segmentation, resolution enhancement, image reconstruction, etc. Due to the strong feature learning capability, GANs are suitable for synthetic rock reconstruction without losing diversity by implicitly learning the probability distribution from the training dataset. Mosser et al. (2017) firstly used DCGANs to reconstruct 3D porous structures of sandstone and carbonate. After their ground-breaking work, DCGANs rapidly swept the community and became an important approach for digital rock reconstruction (Mosser et al., 2018; Liu et al., 2019; Volkhonskiy et al., 2019; Valsecchi et al., 2020). Inspired by this, various GAN models have also been used, including but not limited to conditional GAN (Feng et al., 2019; Volkhonskiy et al., 2019), style GAN (Fokina et al., 2020), progressively growing GAN (Zheng and Zhang, 2020), Bicycle GAN (Feng, et al., 2020), as well as some hybrid models combining variational autoencoder and DCGAN (Shams et al., 2020; Zhang et al., 2021; Zhang et al., 2021).

Even though GANs have achieved great success in digital rock reconstruction, many of them suffer the unstable training issue. This roots in the fact that these networks use the Jesen-Shannon (JS) divergence to evaluate the distance between generated and real data distributions. Unfortunately, the JS divergence remains constant if the real and generated data distributions have no overlap. Under such circumstances, it cannot provide useful gradients to update generator parameters and consequently the training process becomes unstable. We note that a collection of tricks can be used to stabilize GANs training, such as one-sided label smoothing, white noise injection, etc., but they fail to resolve the fundamental issue. Thus, Arjovsky et al. (2017) proposed the Wasserstein GAN (WGAN) by replacing the JS divergence with the distance of Wasserstein to enhance the stable training process. The Lipschitz constraint is conserved by weight clipping in their networks, yet it is prone to cause the gradient vanishing or gradient exploding issue. To overcome this issue, Gulrajani et al. (2017) subsequently improved WGANs by introducing a gradient penalty to conserve the Lipschitz constraint instead of weight clipping. One distinct advantage of WGAN-type models is that the loss of discriminator correlates with the quality of generated images well. Thus, it allows us to track the training state more easily and greatly reduces the subjective assessment via visual check of generated images.

Due to these advantages, in this study, we train WGAN-GP networks to reconstruct Berea sandstone and Ketton limestone from 2D image datasets. The training dataset is subsampled from the original binary images and then augmented by applying rotation, as well as slightly horizontal/vertical shear
transformation. We present an integrated framework to automate the whole process from training set generation, network setup to model training, and synthetic image validation. Synthetic rock images are examined in terms of the two-point correlation and Minkowski functionals, and then they can be collected based on a certain property range, like porosity. Even though we can achieve the same purpose by training conditional GANs with a porosity constraint, here are a couple of concerns. First, the literature review shows that current conditional GANs require the porosity of generated rocks approaching to a reference porosity value, but in practice, it is often expected that the porosity of generated rocks is bounded by a certain range. In addition, the objective function of the generator has to include an additional contribution from the porosity constraint. However, selecting the penalty coefficient for this porosity constraint is heuristic, which will increase the difficulty in training a conditional GAN by trial and error. In some literature, the penalty coefficient of the porosity constraint is regarded as a hyperparameter, and it is assumed constant. However, this may lead to a failure of training in our experiments. Instead, the coefficient should dynamically change with the training process, which requires a sophisticated algorithm to control the adjustment of this coefficient. From the perspective of engineering applications, the proposed workflow is simple and robust to generate synthetic rock images under a given rock property within any range of interest.

The rest of this paper is organized as follows. In the next section, we briefly describe the WGAN-GP network. Then an integrated workflow is presented, which automates the entire process consisting of training image preparation, model training, and synthetic rock validation. We will conduct numerical experiments to reconstruct Berea sandstone and Ketton limestone using the WGAN-GP networks and then evaluate synthetic rocks by comparing the two-point correlation and morphological properties of real and generated images. In the end, we will make concluding remarks and present some future works.

**Wasserstein GAN with Gradient Penalty**

Typically, GANs include two critical components, discriminator and generator. During the training process, the discriminator and generator compete with each other. In particular, the discriminator attempts to discern generated images from real images, while the generator tries to create “real” images as much as possible to fool the discriminator. Let us denote the discriminator and generator by $D_{\phi}$ and $G_{\theta}$, respectively. Suppose that the real dataset $x$ is sampled from $P_{\text{Data}}$. The random noise $z$ from a normal distribution $P_z$ is fed into the generator to create rock images after a series of deep convolutional operations. For GANs using the JS divergence to measure the distance between generated and real data distributions, the discriminator serves as a binary classifier with real images labeled by one and generated images labeled by zero. Thus, the two-player game between discriminator and generator can be formulated as the following min-max optimization problem

$$
\min_{\theta} \max_{\phi} \mathbb{E}_{x \sim P_{\text{Data}}} \left[ \log \left( D_{\phi}(x) \right) \right] + \mathbb{E}_{z \sim P_z} \left[ \log \left( 1 - D_{\phi}(G_{\theta}(z)) \right) \right], \tag{1}
$$

where $\mathbb{E}$ calculates the expected value, Eq. (1) is solved by seeking for an optimal set of discriminator parameters $\phi$ to maximize the binary cross-entropy (BCE) of discriminator predictions given the generator parameters are fixed. Then the generator parameters $\theta$ are updated by minimizing the BCE of $-\mathbb{E} \left[ \log \left( D_{\phi}(G_{\theta}(z)) \right) \right]$, rather than $\mathbb{E} \left[ \log \left( 1 - D_{\phi}(G_{\theta}(z)) \right) \right]$ for numerical concern, with the updated discriminator parameters. This training process proceeds by training $D_{\phi}$ and $G_{\theta}$ in sequence using the gradient-descent method until training is completed.

In comparison, the objective function of the WGAN-GP network has the following form (Gulrajani et al., 2017).
\[
\max_{\vec{\theta}} \min_{\phi} \mathbb{E}_{z \sim P_z} \left[ D_{\phi} \left( G_{\vec{\theta}}(z) \right) \right] - \mathbb{E}_{x \sim P_{\text{Data}}} \left[ D_{\phi}(x) \right] + \lambda \mathbb{E}_{\vec{x} \sim P_{\vec{x}}} \left[ \left( \| \nabla_{\vec{x}} D(\vec{x}) \|_2 - 1 \right)^2 \right] , \tag{2}
\]

where \( \lambda \) is the penalty coefficient and it is set to 10 in practical applications. Similarly, we will minimize \( L_D \) and \( L_G \) to optimize parameters of discriminator and generator, respectively

\[
L_D = \mathbb{E} \left[ D_{\phi} \left( G_{\vec{\theta}}(z) \right) \right] - \mathbb{E} \left[ D_{\phi}(x) \right] + \lambda \mathbb{E} \left[ \left( \| \nabla_{\vec{x}} D(\vec{x}) \|_2 - 1 \right)^2 \right] , \tag{3}
\]

\[
L_G = -\mathbb{E} \left[ D_{\phi} \left( G_{\vec{\theta}}(z) \right) \right] . \tag{4}
\]

In Eq. (3), the interpolated image \( \vec{x} \) is a linear combination of the real image \( x \) and generated image \( \vec{x} = G_{\vec{\theta}}(z) \)

\[
\vec{x} = \epsilon \ast x + (1 - \epsilon) \ast \vec{x} ,
\]

where \( \epsilon \in (0, 1) \) is a random number sampled from the uniform distribution. Instead of training the discriminator and generator one by one in DCGANs, we train the discriminator in the WGAN-GP network five times and then update the generator once. Moreover, batch normalization is removed from the discriminator to ensure the accuracy of gradient penalty with respect to the inputs, as suggested by Zheng and Zhang (2020).

**Proposed Workflow**

To automate the entire process of digital rock reconstruction, an integrated workflow is presented, which consists of three main steps: training set generation, model training, and synthetic image validation, as shown in Figure 1. We will give a detailed description of each of the steps in the following content.
1. Training set generation
In this study, we used the Berea sandstone and Ketton limestone images to train our WGAN-GP network. For illustration purposes, 2D binary images are used as training input, but this doesn’t affect the extension of the proposed workflow to reconstruct 3D porous structures. To have a sufficient training dataset, we first create the training set by subsampling the original binary image. For the Berea sandstone, the kernel size is set to $64 \times 64$ pixels$^2$ with a stride size of 8 pixels. For the Ketton limestone, the original image is first downsampled from $500 \times 500$ pixels$^2$ to $256 \times 256$ pixels$^2$ in order to capture pores and grains in the Ketton limestone (Mosser et al., 2017). Similarly, we use a kernel size of $64 \times 64$ pixels$^2$ with a smaller moving stride of 4 pixels to extract subvolumes as the initial training dataset. To further augment our dataset, the training image is rotated by an angle, randomly sampled from a uniform distribution between 0 to 360 degrees. In addition, the pore structure is also slightly transformed by applying a horizontal or vertical shear angle which is uniformly distributed within $[-10, 10]$ degree. The augmented dataset will be used to train the WGAN-GP network. We will shuffle the entire dataset once all mini-batches are iterated over.
2. Model Training

Figure 2 shows the network architecture of our WGAN-GP model. To synthesize a scalable image, we adopt fully convolutional architectures for the generator. Particularly, the generator has X blocks, the first five of which are composed of the transposed convolution layer, batch normalization layer, and ReLU activation layer. Each transposed convolution layer uses a kernel size of $3 \times 3$ and stride size of 2. As the transposed convolution layer goes deeper and deeper, the size of feature maps is enlarged by a factor of 2, while the number of filters is half of that in the precedent layer. The last transposed convolution layer of the generator is followed by a hyperbolic tangent activation function such that output values are bounded between $-1$ and 1.

On the other hand, the discriminator has Y convolution layers with a kernel size of $3 \times 3$ and a stride size of 2. After each convolution layer, the image size is reduced by half, and accordingly, the number of filters doubles. As mentioned in the last section, we remove the batch normalization layer to preserve the accuracy of the gradient penalty. Therefore, the first four convolution layers are followed by the leaky ReLU activation function immediately, and the last convolution layer directly outputs the result. The detailed layer setup in the generator and discriminator are summarized in Table 1.

Table 1. Network Architecture of the generator and discriminator.

<table>
<thead>
<tr>
<th>Block</th>
<th>Type</th>
<th>Filters</th>
<th>Kernel Size</th>
<th>Stride</th>
<th>Batch Normalization</th>
<th>Activation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TransposedConv</td>
<td>512</td>
<td>$3 \times 3$</td>
<td>4</td>
<td>Yes</td>
<td>ReLU</td>
</tr>
<tr>
<td>2</td>
<td>TransposedConv</td>
<td>256</td>
<td>$3 \times 3$</td>
<td>2</td>
<td>Yes</td>
<td>ReLU</td>
</tr>
</tbody>
</table>
To train our WGAN-GP networks, the adam optimizer is used with the learning rate of $2 \times 10^{-4}$. The dimension of latent input is set to 100, and the mini-batch size is 64. For the leaky ReLU activation function, the negative slope coefficient is 0.2. The WGAN-GP networks are trained in a scheme that the discriminator is updated 5 times after one update of the generator. Totally, we train our model for 10000 generator updates. Images generated by the WGAN-GP network are then post-processed by rescaling grayscale values to $[0, 1]$. Removing single-pixel noise using a $3 \times 3$ median filter and binarizing the resultant image with Otsu’s method (Otsu, 1979).

3. Results Validation

As a successful digital rock reconstruction method, the reconstructed pore structures are supposed to be diversified with similar geometrical and hydraulic properties to the given rock sample. In order to evaluate the generative performance, in this study, we calculate geometrical properties of synthetic rock images, including the two-point correlation and three Minkowski functionals, and compare them with the real dataset. The two-point correlation $S_2$ characterizes the probability that two points $x$ and $x + r$, separated by a distance of $r$, are located in the pore space

$$S_2(r) = P(x \in P, x + r \in P), \text{for } x, r \in \mathbb{R}^d.$$  (5)

On the other hand, Minkowski functionals are often used to describe the morphological characteristics of pore structures, which are correlated with transport properties. For a 2D binary image, we can estimate the area of pore phase, equivalent to porosity in 2D, the circumference, and 2D Euler characteristic in terms of the zeroth-, first- and second-order Minkowski functionals, respectively. A detailed description regarding the calculation of Minkowski functionals on the 2D binary image can be found in the reference (Legland et al., 2007).

**Numerical Results**

We first trained a WGAN-GP network to reconstruct the pore structures of Berea sandstone. To visualize the generator performance, we compare 25 images randomly sampled from the training dataset to synthetic images in Figure 3, with the size of $64 \times 64$ pixels$^2$. Overall, the generated samples capture the pore structures of the training samples. Figure 4 shows two-point correlation functions of 100 training and reconstructed Berea sandstone images, both of which are not scaled by porosity. The shaded area is plotted at one standard deviation with the center of mean values. It can be seen the two-point correlation functions of the generated images agree with those of the original images very well. Figure 5 shows the box plots for three Minkowski functionals, namely (a) area, (b) circumference, and (c) 2D Euler characteristic of 100 training and synthetic images. As we can see, three Minkowski functionals exhibit good agreement between the training and reconstructed samples. On the other hand, the generated images show a wide
range of Minkowski functionals, implying the trained WGAN-GP network enables the generation of diverse pore structures yet with similar morphological features to the original images. We also use the trained WGAN-GP network to generate synthetic Berea sandstone images with the size of $128 \times 128$ pixels, which is shown in Figure 6.

![Figure 3](image1.png)

Figure 3. Twenty-five images of Berea sandstone for training (a) and synthetic images given by the generator (b).

![Figure 4](image2.png)

Figure 4. Comparison of two-point correlation functions for 100 training and synthetic Berea sandstone images, respectively.
Figure 5. Comparison of Minkowski functionals for 100 training and synthetic Berea sandstone images, (a) area (porosity), (b) circumference, and (c) 2D Euler characteristic.

Figure 6. Twenty-five reconstructed Berea sandstone samples with a size of $128 \times 128$ pixels$^2$. 
Another WGAN-GP network is trained to reconstruct Ketton limestone. Similarly, the generative performance is examined in Figure 7 by comparison of the synthetic images, with the size of $64 \times 64$ pixels$^2$, to the training images. Then we calculate the two-point correlation functions, as shown in Figure 8. Figure 9 compares three Minkowski functionals of 100 training and synthetic Ketton limestone samples. In comparison to Berea sandstone, the median porosity of reconstructed samples has a distinct decline. This may be attributed to that the shear transformation reduces porosities of training images for Ketton limestone more significantly than Berea sandstone. In general, the results show that the synthetic images show good agreement with the original images without losing diversity. Figure 10 displays the generated Ketton limestone images with the size of $128 \times 128$ pixels$^2$, which pronouncedly shows the hole effect as the result of large grains.

![Figure 7](image_url)

**Figure 7.** Twenty-five images of Ketton limestone for training (a) and synthetic images given by the generator (b).
Figure 8. Comparison of two-point correlation values for 100 training and synthetic Ketton limestone images.

Figure 9. Comparison of Minkowski functionals for 100 training and synthetic Ketton limestone images, (a) area (porosity), (b) circumference, and (c) 2D Euler characteristic.
Figure 10. Twenty-five reconstructed Ketton limestone samples with a size of $128 \times 128$ pixels.

**Conclusion and Future Works**

In this study, WGAN-GP networks are trained to reconstruct Berea sandstone and Ketton limestone. The training dataset is augmented by random rotation and shear transformation of the initial training samples. However, unlike random rotation of training images, the shear transformation could change pore structures and consequently affect the training process. An integrated framework is presented to automate the entire workflow, including training set generation, Network setup, model training, and synthetic rock validation. We evaluated the generated Berea sandstone and Ketton limestone images in terms of the two-point correlation and Minkowski functionals. It is found that WGAN-GP networks could reproduce pore structures without losing diversity, and more importantly, the training process is stable in comparison to GANs using the JS divergence to measure the distance between the real and generated data distributions. In the future work, we will extend to reconstruct 3D porous structures using the WGAN-GP network and evaluate the synthetic rocks from both geometrical and hydraulic aspects. In particular, we will calculate the permeability of the generated rock samples to analyze the connectivity of pore structures. In addition, to reconstruct multi-scale carbonate rocks, the variational autoencoder will be used to extract microscopic features from SEM images to the synthetic rocks from micro-CT images.
Acknowledgments
We would like to thank Saudi Aramco for funding this research. We would also like to thank King Abdullah University of Science and Technology (KAUST) for providing a license of MATLAB.

References
OnePetro.


