applied sciences

MDPI

Article

WGAN-Based Realization Process of Gravel Soil for Hydraulic Property Simulation

Bin Zhu * and Xiang-Gang Hu

Earth Sciences College, Guilin University of Technology, Guilin 541004, China; 1994009@glut.edu.cn
* Correspondence: 1999002@glut.edu.cn

Abstract: Gravel soil faces significant engineering challenges such as leakage erosion and soil flow due to its complex composition and susceptibility to groundwater effects. This study integrates the entire machine learning process, including pre- and post-processing of images, WGAN implementation, and validation of hydraulic and morphological properties. Obtaining intact gravel soil samples is difficult and costly due to their erodible nature in the Li River, China. A μ-CT scanning series is employed to capture detailed images with three microstructural characteristics of gravel soil, forming the basis for training datasets using WGANs. This approach allows the generation of similar 3D realizations that replicate the microstructural characteristics and hydraulic behaviors of a prototype of gravel soils. Through computational fluid dynamics (CFD) simulations, the effectiveness of the realizations in hydraulic behavior within reconstructed porous structures is verified. This process indirectly validates the consistency between the realization's microstructure and the prototype. This integrated methodology not only enhances understanding but also aids in the optimization of engineering designs and applications in geotechnical and materials science disciplines.

Keywords: WGANs (Wasserstein Generative Adversarial Networks); machine learning for porous structure; pore-scale CFD simulation; gravel soil; hydraulic properties imitation

check for updates

Citation: Zhu, B.; Hu, X.-G. WGAN-Based Realization Process of Gravel Soil for Hydraulic Property Simulation. *Appl. Sci.* **2024**, *14*, 9873. https://doi.org/10.3390/app14219873

Academic Editor: Fulvia Chiampo

Received: 4 September 2024
Revised: 16 October 2024
Accepted: 23 October 2024
Published: 29 October 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The reconstruction and the evaluation of the material properties of porous media play a key role across many engineering disciplines. Many physical processes such as the movement of multiple phases of fluids through sedimentary rocks are controlled by individual pores at the micron and sub-micron scales. In Guilin city along the Li River, there is a special soil layer consisting of a unique composition of pebble gravel interspersed with fine sand or silty clay. This gravel soil layer is vulnerable to disturbances due to its high groundwater content, which can result in various engineering challenges such as seepage erosion, piping, soil flow, and even water-inrush incidents. A typical pit water-inrush accident induced by seepage erosion of fine particles was reported in our recently published paper [1]. The erodible stability of this type of gravel soil often depends on the ratio of coarse to fine particle contents, which is a complex issue that has long troubled the field of engineering worldwide. Obtaining intact samples of this gravel soil is difficult and expensive due to the laborious process of acquisition and experimentation with physical specimens.

To address the engineering challenges mentioned above, understanding the internal structure and hydraulic behavior of geotechnical materials is crucial. A proposed solution involves using μ-CT scanning to acquire digital images of gravel soil samples [2]. This method begins with standard soil sampling followed by CT scanning to capture these images. Representative images are then selected to create a training dataset for image generation aimed at reconstructing virtual core samples with microstructures and properties similar to those of the original soil samples. This approach allows the creation of multiple digital 3D virtual samples that replicate the properties of standard samples, maintaining the microstructural characteristics of the original sample collection. It forms the basis for virtual

geological scenarios utilizing the digital-twin technique for material microstructures [3,4]. This method significantly reduces the need for extensive trial-and-error experiments.

Generative adversarial networks (GANs), introduced by Goodfellow et al., 2014 [5], have become a prominent technique in the realm of reconstruction and digital-twin development using unsupervised intelligent deep learning approaches. GANs are notable for their capability to generate realistic data through the interplay of two neural networks: the generator and the discriminator, which compete against each other. This technology is valuable for tasks such as data generation, augmentation, and sample expansion and is particularly beneficial in scenarios with limited annotated data. Currently, GANs have found applications across various domains including soil moisture simulation [6], soil deformation and cracking simulation, and 3D digital core reconstruction of rock masses, as documented in recent studies [7]. These applications leverage GANs′ ability to generate synthetic data that closely mimic real-world phenomena, aiding in modeling complex processes and enhancing understanding. To address challenges like unstable training dynamics inherent in GANs, Gulrajani et al. (2017) [8] introduced an advanced algorithm known as Wasserstein Generative Adversarial Networks with Gradient Penalty (WGAN-GP). This method helps mitigate issues such as gradient vanishing and exploding, thereby improving the stability and effectiveness of GAN training processes. In summary, GANs represent a powerful tool in modern data science and engineering, enabling the creation of synthetic data that enhance model training, simulation accuracy, and understanding of complex systems across diverse fields.

The field of large-scale spatial model reconstruction, particularly in earth modeling applications such as reservoir characterization, often employs MPS (multiple-point statistics) methods. These techniques are essential for reconstructing stochastic random media in two- and three-dimensional scenarios. MPS methods excel in capturing the complex geological structures and patterns that typify the earth′s subsurface, thereby addressing the heterogeneity and uncertainty inherent in such environments. Historically, methods like simulated annealing have been pivotal in stochastic modeling for their ability to incorporate arbitrary cost functions related to statistical and morphological properties, which are crucial for unconditional three-dimensional image reconstruction. These techniques laid the groundwork for more advanced approaches that followed. In recent years, there has been significant innovation with the integration of MPS methods and GANs (Generative Adversarial Networks). Laloy (2018) [9] proposed a method that combines MPS and GANs for geological modeling, specifically for generating features like river channels and reservoirs. This integration leverages GANs′ ability to produce realistic and diverse synthetic data, enhancing the fidelity and variability of geological models beyond what traditional MPS methods alone can achieve. This combined approach allows the creation of highly detailed and varied geological structures, improving the accuracy and usefulness of earth modeling in applications ranging from hydrology to oil and gas reservoir characterization. The synergy between MPS and GANs represents a promising direction in spatial modeling, enabling more robust simulations and predictions in geosciences.

In reconstructing small- or pore-scale media, researchers utilize high-resolution digital images from techniques like μ-CT or medical CT scans. Generative Adversarial Networks (GANs) are then employed to generate virtual samples mimicking these structural characteristics. To enhance fidelity, GANs employ gradient-differentiable Wasserstein distance as a loss function, surpassing limitations of traditional divergences like KL and JS. Additionally, penalty functions ensure stable training, addressing issues such as mode collapse and improving overall sample quality. This approach facilitates accurate reconstruction and modeling of complex materials in fields such as soil science and material science.

This paper introduces a systematic approach to digitally reconstructing gravel soil samples using μ-CT scan slice images and advanced computational techniques, involving image processing, MATLAB (R2022a) coding, PyTorch (3.9) programming, working flow, and so on. In Section 2, a dataset is meticulously constructed from these μ-CT scan images, involving the acquisition, preprocessing, and compilation of data suitable for machine

learning. In Section 3, a generator model based on WGAN-GP is trained with the dataset to meticulously recreate digital core samples, placing strong emphasis on fidelity assessment and evaluating model performance. In Section 4, computational fluid dynamics (CFD) simulations are employed to validate the consistency of the hydraulic properties of these generated models in comparison to the training set (prototype). By simulating fluid behavior within the reconstructed porous structures, engineers can gain insights into how these materials perform under different flow conditions, informing design decisions and optimizing applications in fields such as materials science and engineering. In Section 5, the main hydraulic properties of soils and some statistical functions are then employed to quantitatively measure the similarity between the reconstructed samples and the original data, ensuring robustness and accuracy in the reconstruction process. Section 6 includes a discussion of the processes and indices, such as specific area and Eula characteristics, that facilitate the assessment of hydraulic properties and validate the effectiveness of the generation method for practical engineering applications. Some conclusions are drawn in Section 7. This comprehensive approach underscores the integration of imaging, machine learning, and simulation techniques to advance understanding and analysis in geological and engineering studies.

The paper aims to digitally reconstruct gravel soil samples using μ-CT scanning, machine learning, and computational simulation techniques to replicate their hydraulic properties, advancing understanding and addressing engineering challenges in environments like Guilin city along the Li River.

2. Generation of the Training Datasets

According to the composition characteristics of gravel soils in Guilin city, three groups of granular soil specimens, each characterized by different fine grain content features, were placed in transparent acrylic cylindrical containers measuring 350 mm in height and diameter. These specimens exhibited characteristics typical of gap-graded gravel soils, lacking particles in the ranges of 1–2 mm, 0.5–1 mm, and 0.25–0.5 mm, which basically represent the typical characteristics of gravel soil in Guilin. High-resolution 3D X-ray microimaging using the Zeiss Xradia 510 Versa system produced μ-CT images for each group, resulting in voxel datasets sized $1004 \times 1024 \times 1014$ with a resolution of 35 µm/voxel.

After obtaining the μ-CT raw dataset of 1014 pieces of 2D CT-images, each sized 1004×1024 pixels, the images were processed using ImageJ software (1.54f) with threshold segmentation. Different threshold values based on grayscale intensity were applied to segment gravelly soil into two main components: void or solid skeleton. Using the cropping function in Avizo (2020), regions of interest (ROIs) containing well-developed pores were selected and cropped to 700×700 pixels each, resulting in 700 cropped images per dataset. These images were then overlaid to create 3D representations, which were subsequently sliced into multiple $64 \times 64 \times 64$ voxel-sized 3D images. This entire process generated three distinct datasets, each comprising 1900 training sets, as illustrated in Figure 1.

The generation of training datasets involves tedious processes such as 3D μ-CT image processing, conversion of images to matrices, and the formation of sub-models. Converting 3D pixel images into vector graphics is an important step in the early stage of AI graphics generation. Training set generation and storage based on MATLAB constitute an effective means for image processing and training dataset production in the early stage. The following points are the main steps to produce training datasets in the form of a MATLAB-based matrix.

Formation of Image Matrices: Utilize the MATLAB script to read two-dimensional slice image files from a specified directory and load them into a three-dimensional array (volume = $700 \times 700 \times 700$).

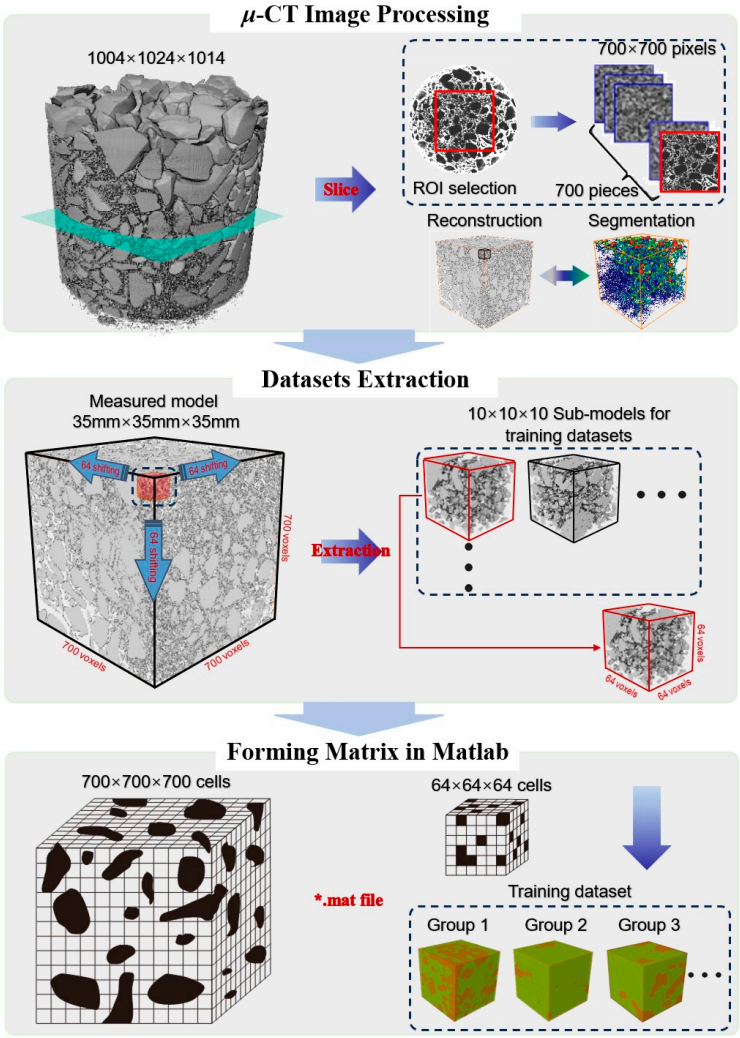


Figure 1. A scheme of image processing and training set acquisition for different groups of soil specimens. Different gravel soil samples with unique characteristics of porous distribution can be formed; *nth* groups of 64 × 64 × 64 in size for the training set were selected randomly from the 700 × 700 × 700 primitive soil prototype. Note: *.mat file is MATLAB's data storage format, used to save variables, arrays, and structures.

Determination of Sub-Model Scope: Define a step size (stepSize = 64) and a block size (blockSize = 64 × 64 × 64) to cut up the three-dimensional array into uniformly-sized blocks along the x, y, and z directions. Calculate the number of blocks (numBlocks = 1000) that can be generated based on the given step size.

Extraction of Sub-Models: Iterate through each block index, extract the corresponding block from the three-dimensional array, and save it as an M file. Ensure each file name includes information about its block index. Save all extracted block files in the specified directory.

Achieved Outcome: The code effectively partitions the 3D μ-CT raw data into consistent block sizes and stores these blocks as M files, which are ready for subsequent processing and analysis.

The whole process of training dataset generation is described in Figure 1. The μ-CT image data for ML learning in the study are freely accessible in Figshare (doi.org/10.6084/m9.figshare.25665360) (CC BY 4.0) [10].

Specific preprocessing steps applied to the μ-CT scan images prior to the above steps was necessary. Preprocessing of micro-computed tomography (μ-CT) images is crucial for improving the quality of the data before further machine learning processing. The specific steps can vary depending on the application, but the following techniques have been adopted for managing noise and artifacts to ensure clean input data: (1) Median filtering is applied to remove salt-and-pepper noise without blurring the edges of structures; (2) image alignment and registration techniques are applied to align the multiple images correctly; and (3) thresholding segmentation techniques are applied to isolate regions of pores and solid materials. The preprocessing of μ-CT images is a critical step to ensure that the data are clean and ready for analysis. The above steps help manage noise and artifacts effectively, leading to improved image quality and more reliable results for subsequent WGAN learning.

3. Model Training

3.1. Generative Adversarial Nets (GANs)

Goodfellow et al. (2014) [5] proposed the framework of the GAN technique to address the generative problem, which aims to train a generative model that can approximate an unknown distribution p_{data} over a high-dimensional space $\boldsymbol{\chi}$ given many observed real samples x_r from this distribution, i.e., $x_r \sim p_{data}$. Figure 2 illustrates the typical workflow of a GAN. A GAN generally consists of two trainable blocks, a generator (G_θ), and a discriminator (D_φ), with θ and φ being trainable parameters. A latent variable Z is defined with a known distribution p_z (e.g., Gaussian) over a low-dimensional space $\boldsymbol{Z}$, and z is considered a sample from $\boldsymbol{Z}$, i.e., $z \sim p_z$. The generator G_θ maps z into x_G, which lies in the space $\boldsymbol{\chi}$, i.e., $x_G = G_\theta(z)$; the distribution of x_G is p_G, i.e., $x_G \sim p_G$. The discriminator D_φ maps the generated samples x_G and the given data samples x_r into two scalar values called scores s_G and s_r, respectively, i.e., $s_G = D_\varphi(x_G)$, $s_r = D_\varphi(x_r)$. These scores evaluate the similarity of the inputs of x_G and x_r. The loss function of GANs is based on some type of distance between s_G and s_r (Equation (1)), which represents the distance between p_G and p_{data}. As shown in Figure 2, the discriminator D_φ and the generator G_θ are alternatively trained by maximizing and minimizing the loss function (Equation (1)) until a certain stopping (Minimax) criterion is reached.

$$J^{(D)} = -\boldsymbol{E}_{x_r \sim p_{data}} log D_\varphi(x_r) - \boldsymbol{E}_{x_G \sim p_G} log\big(1 - D_\varphi(G_\theta(z))\big) \tag{1}$$

$$J^{(G)} = \boldsymbol{E}_{x_G \sim p_G} log\big(1 - D_\varphi(x_G)\big) \tag{2}$$

where $J^{(\cdot)}$ is some type of distance between x_G and x_r, such as Jensen-Shannon divergence, and $\boldsymbol{E}$ is the expectation over $x_G \sim p_G$ and $x_r \sim p_{data}$.

The generator and discriminator antagonize each other until the generated data fully merge with the real data, making the distributions of the model almost identical to reach a Nash equilibrium, i.e., a saddle point of the discriminator loss. The optimization strategy of the discriminator can be derived as follows.

$$D^*_\varphi(x) = \frac{p_{data}(x_r)}{p_{data}(x_r) + p_G(x_G)} \tag{3}$$

Estimating this ratio using supervised learning is the key approximation mechanism used by GANs. As shown in the last picture in Figure 2, when the discriminator (D_φ) is unable to effectively distinguish between the real samples x_r and the generated samples x_G, it indicates that the generator (G_θ) has successfully learned the features that are similar to the real data distribution, i.e., $x_r \sim p_{data}$. This means that the generator has acquired the ability to generate realistic samples.

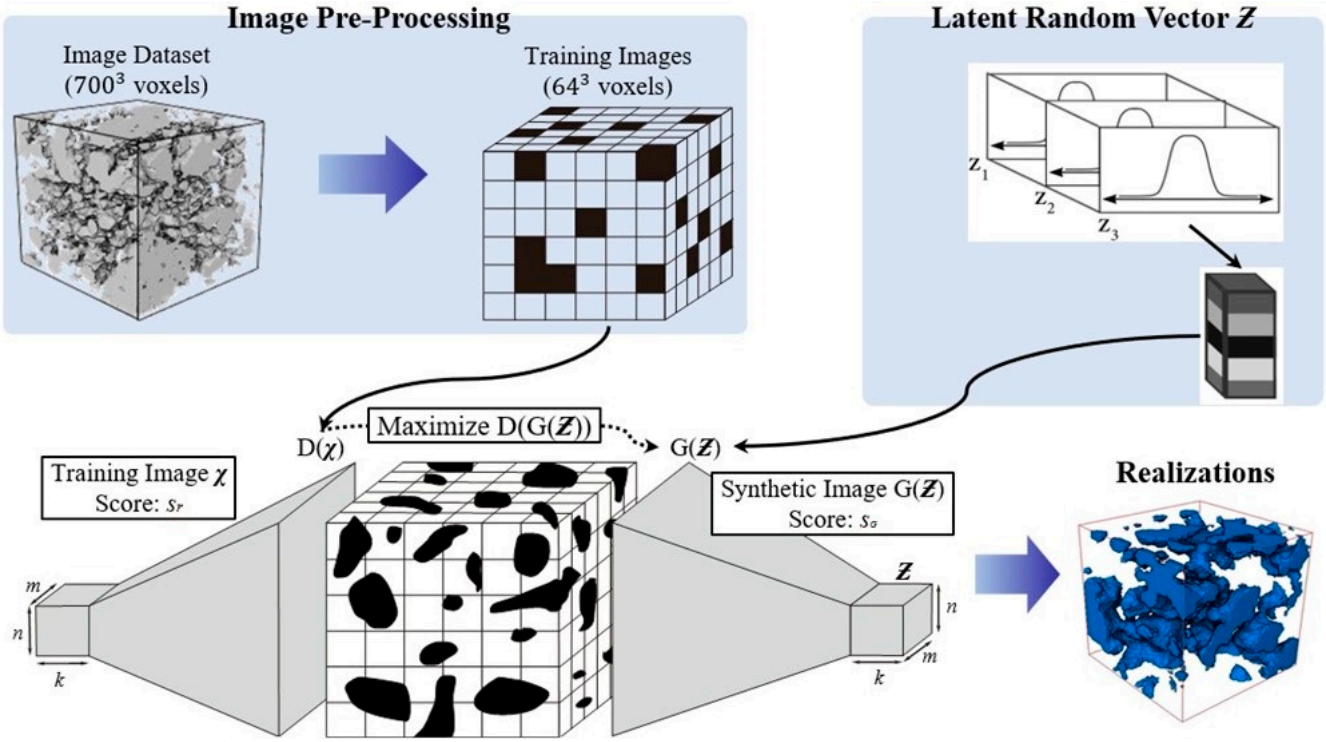


Figure 2. Overview of the GAN training process. In GAN training, segmented volumetric images are divided into smaller voxel training images. The generator (G) creates synthetic images from random latent space ($\boldsymbol{Z}$) samples, which are assumed to follow a uniform distribution ~U (−1,1). The discriminator (D) distinguishes between real training images (scored as s_r) and generated images (scored as s_G) using binary cross-entropy. Through iterative training, the generator improves to produce high-quality images that deceive the discriminator. Training stops when desired image quality is achieved, and the generator can then generate new realizations by inputting larger latent vectors for larger output images.

In summary, the GAN framework can train a generative model that approximates an unknown data distribution by utilizing observed real samples. A GAN consists of a generator and a discriminator, where the generator creates data from a latent variable, and the discriminator evaluates the authenticity of the generated and real samples. The two components are trained in opposition to minimize their respective loss functions until they reach a Nash equilibrium, signifying that the generator can produce realistic samples indistinguishable from the actual data distribution.

3.2. 3D Deep Convolutional Neural Network for GANs

The integration of Convolutional Neural Network (CNN) layers into Generative Adversarial Networks (GANs), which produces Deep Convolutional GANs (DCGANs), has significantly enhanced the efficiency of network architectures [11]. Unlike traditional GANs, DCGANs specify a structured architecture:

The discriminator replaces pooling layers with convolutional layers, while the generator replaces them with deconvolution layers (transposed convolution layers).

Both the generator and discriminator employ batch normalization, enhancing training stability and convergence.

In deeper architectures, fully connected hidden layers are eliminated.

The generator uses ReLU activation in all layers except the output layer, which employs Tanh to constrain outputs between −1 and 1.

The discriminator uses LeakyReLU activation across all layers to improve non-linearity and address issues like neuron death.

DCGANs have extended into three-dimensional models for tasks such as X-ray CT imaging, MRI scanning, and imaging of optically transparent media. These 3D-DCGANs utilize volumetric convolutional networks to generate $64 \times 64 \times 64$ voxel-sized images from latent vectors sampled from a probabilistic latent space $\boldsymbol{Z}$. Unlike CNNs that extract features from images, 3D-DCGANs amplify features from random signals using deconvolution, reconstructing them into new images. The architectures of the generator and discriminator of the 3D Deep Convolutional GAN in this work is shown as Table 1.

Table 1. The architectures of the generator and the discriminator used in this study.

	Block	Layer	Activation	Output Shape	Strides	Filling	Normal
Generator		Latent vector	-	200×1	-	-	-
	Block 1 ($4 \times 4 \times 4$)	Fully connected	ReLU	$32{,}768 \times 1$	-	-	-
		Reshape	-	$4 \times 4 \times 4 \times 512$	-	-	-
		Convolutional $4 \times 4 \times 4$	ReLU	$4 \times 4 \times 4 \times 512$	1	0	yes
	Block 2 ($8 \times 8 \times 8$)	Upsampling	-	$8 \times 8 \times 8 \times 256$	-	-	-
		Convolutional $4 \times 4 \times 4$	ReLU	$8 \times 8 \times 8 \times 256$	2	1	yes
	Block 3 ($16 \times 16 \times 16$)	Upsampling	-	$16 \times 16 \times 16 \times 128$	-	-	-
		Convolutional $4 \times 4 \times 4$	ReLU	$16 \times 16 \times 16 \times 128$	2	1	yes
	Block 4 ($32 \times 32 \times 32$)	Upsampling	-	$32 \times 32 \times 32 \times 64$	-	-	-
		Convolutional $4 \times 4 \times 4$	ReLU	$32 \times 32 \times 32 \times 64$	2	1	yes
	Block 5 ($64 \times 64 \times 64$)	Upsampling	-	$64 \times 64 \times 64 \times 1$	-	-	-
		Convolutional $1 \times 1 \times 1$	Tanh	$64 \times 64 \times 64$	2	1	no
Discriminator		Input image	-	$64 \times 64 \times 64$	-	-	-
	Block 5 ($64 \times 64 \times 64$)	Convolutional $1 \times 1 \times 1$	LReLU	$64 \times 64 \times 64 \times 1$	1	0	no
		Convolutional $4 \times 4 \times 4$	LReLU	$64 \times 64 \times 64 \times 1$	2	1	no
	Block 4 ($32 \times 32 \times 32$)	Downsampling	-	$32 \times 32 \times 32 \times 64$	-	-	-
		Convolutional $4 \times 4 \times 4$	LReLU	$32 \times 32 \times 32 \times 64$	2	1	yes
	Block 3 ($16 \times 16 \times 16$)	Downsampling	-	$16 \times 16 \times 16 \times 128$	-	-	-
		Convolutional $4 \times 4 \times 4$	LReLU	$16 \times 16 \times 16 \times 128$	2	1	yes
	Block 2 ($8 \times 8 \times 8$)	Downsampling	-	$8 \times 8 \times 8 \times 256$	-	-	-
		Convolutional $4 \times 4 \times 4$	LReLU	$8 \times 8 \times 8 \times 256$	2	1	yes
	Block 1 ($4 \times 4 \times 4$)	Downsampling	-	$4 \times 4 \times 4 \times 512$	-	-	-
		Convolutional $4 \times 4 \times 4$	LReLU	$4 \times 4 \times 4 \times 512$	1	0	no
		Reshape	Flatten	$32{,}768 \times 1$	-	-	-

In this context, the architecture of the 3D Deep Convolutional GAN includes a generator with five blocks, each comprising a transposed deconvolutional layer, batch normalization, and ReLU activation. The final block uses Tanh to output images constrained within -1 to 1. This process involves mapping a random noise vector $z \sim p_z$ to generate an image $x_G \sim p_G$. The discriminator is structured with four convolutional layers and one fully connected layer using LeakyReLU activation for processing binary images.

Training the 3D-DCGAN involves using the RMSprop optimizer across 500 epochs, with a batch size of 64 and a learning rate of 0.0002. The loss function incorporates a gradient penalty term to prevent issues like mode collapse and gradient explosion, ensuring stable and effective training of the network.

3.3. Loss Function of the WGAN with Gradient Penalty (GP)

The improvements proposed by [12] and subsequent researchers address several key challenges in training Generative Adversarial Networks (GANs), aiming to stabilize and enhance the quality of generated images. A detailed explanation of each optimization strategy compared to the original GAN algorithm is provided below:

Removal of the sigmoid activation function: Unlike traditional GANs, WGANs eliminate sigmoid activation at the discriminator's output layer. This change allows the discriminator to output unbounded real numbers, which are used to compute distances like Wasserstein or Earth-Mover's distance.

Log-less loss function: Traditional GANs use logarithmic loss functions, such as binary cross-entropy. WGANs, on the other hand, use loss functions that do not involve logarithms.

They focus on minimizing differences between distributions, which are measured by the Wasserstein distance.

Weight clipping: WGANs enforce a Lipschitz constraint using weight clipping. After each discriminator parameter update, weights are clipped to prevent them from exceeding a fixed constant "c". This constraint stabilizes training and prevents the discriminator from overpowering the generator.

Optimization algorithms: Momentum-based optimizers like Momentum and Adam, which are commonly used in traditional neural networks, are not ideal for WGANs. Instead, RMSProp and SGD are recommended due to their stability during training, especially in conjunction with weight clipping and the absence of sigmoid activation functions.

These changes collectively address the instability issues commonly encountered in training original GANs, such as mode collapse (where the generator fails to explore the entire data distribution) and discriminator overpowering (where the discriminator becomes too effective, making it difficult for the generator to improve). By focusing on the Wasserstein distance and implementing these modifications, WGANs have shown improved stability and generated image quality in various applications of generative modeling.

The proposed Wasserstein loss function based on the Wasserstein distance between p_{data} and p_G is shown in Equation (4):

$$W(G_\theta, D_\varphi) = E_{x_r \sim p_{data}} D_\varphi(x_r) - E_{z \sim p_z} D_\varphi(G_\theta(z)) \quad (4)$$

where D_φ does not have the sigmoid function in the last layer and constrains the backpropagate training parameters in the range of (−c, c) by weight clipping.

Gulrajani et al. (2017) [8] improved weight clipping further using an alternative method to penalize the norm of the gradient of the discriminator with respect to its input. The proposed method performs better than standard WGANs and enables stable training of a wide variety of GAN architectures with almost no hyperparameter tuning. This improved loss function is shown in Equation (5):

$$W(G_\theta, D_\varphi) = E_{x_r \sim p_{data}} D_\varphi(x_r) - E_{z \sim p_z} D_\varphi(G_\theta(z)) - \lambda E_{\hat{x} \sim p_{\hat{x}}} \left[\left(||\nabla_{\hat{x}} D_\varphi(\hat{x})||_2 - 1 \right)^2 \right] \quad (5)$$

where λ is the penalty coefficient and the default value is 10, and $p_{\hat{x}}$ is the distribution of the sampled $\hat{x}$ between $x_G \sim p_G$ and $x_r \sim p_{data}$ by the *t~uniform* (0,1) function, i.e., $\hat{x} = t x_r + (1-t) x_G$.

In our study, we took the loss function of the WGAN with the gradient penalty as the minimizing objective for backward propagation to ensure more stable training performance.

We utilized PyTorch to train and test a model for digital realization reconstruction. It begins with preparing datasets by placing them in a designated folder. Ensuring the PyTorch environment, users configure key variables and hyperparameters such as learning rate and batch size. Training is initiated by executing main.py, while pre-execution checks include verifying dataset formats and adjusting any hardcoded paths in the code. For testing, users modify the path in test.py to load the generator model, facilitating the reconstruction of digital realizations when executed within the PyTorch environment. Throughout training, essential data such as loss functions, MDS data, and model states are automatically saved, facilitating subsequent analysis and utilization. This structured approach ensures systematic setup, execution, and validation of the digital realization reconstruction process using PyTorch. The code and plotting data for WGAN-GP in the study are freely available at GitHub via https://github.com/Zhu-Bi/Gravelsoil_GAN [13], accessed on 1 May 2024.

4. Generation of Gravel Soil Models and Evaluation Criteria

4.1. Method

Figure 3 illustrates three randomly chosen realizations for each of the three groups of real gravel soil models, which were randomly selected from the training set with a size of 64 × 64 × 64 voxels. When evaluating criteria for an AI generative model of porous materi-

als, key considerations include the accurate representation of microstructural features like pore size distribution—two-point correlation function [14,15] and connectivity—Euler characteristic [16]. Quality training data sourced from simulations or experiments ensure that the model learns from realistic examples. The choice of generative methodology—whether GANs or physics-informed models—depends on the desired outcomes, such as realism or adherence to physical laws, i.e., hydraulic behaviors. Validation metrics, such as pore size distribution or permeability, gauge the model's fidelity to real materials. Ensuring physical consistency via constraints or physics-informed approaches is crucial to align generated structures with real-world properties. Lastly, the model's application scope, whether in material design or industrial use, dictates its practical relevance and effectiveness. These considerations collectively guide the development of robust AI generative models tailored for porous material research and applications.

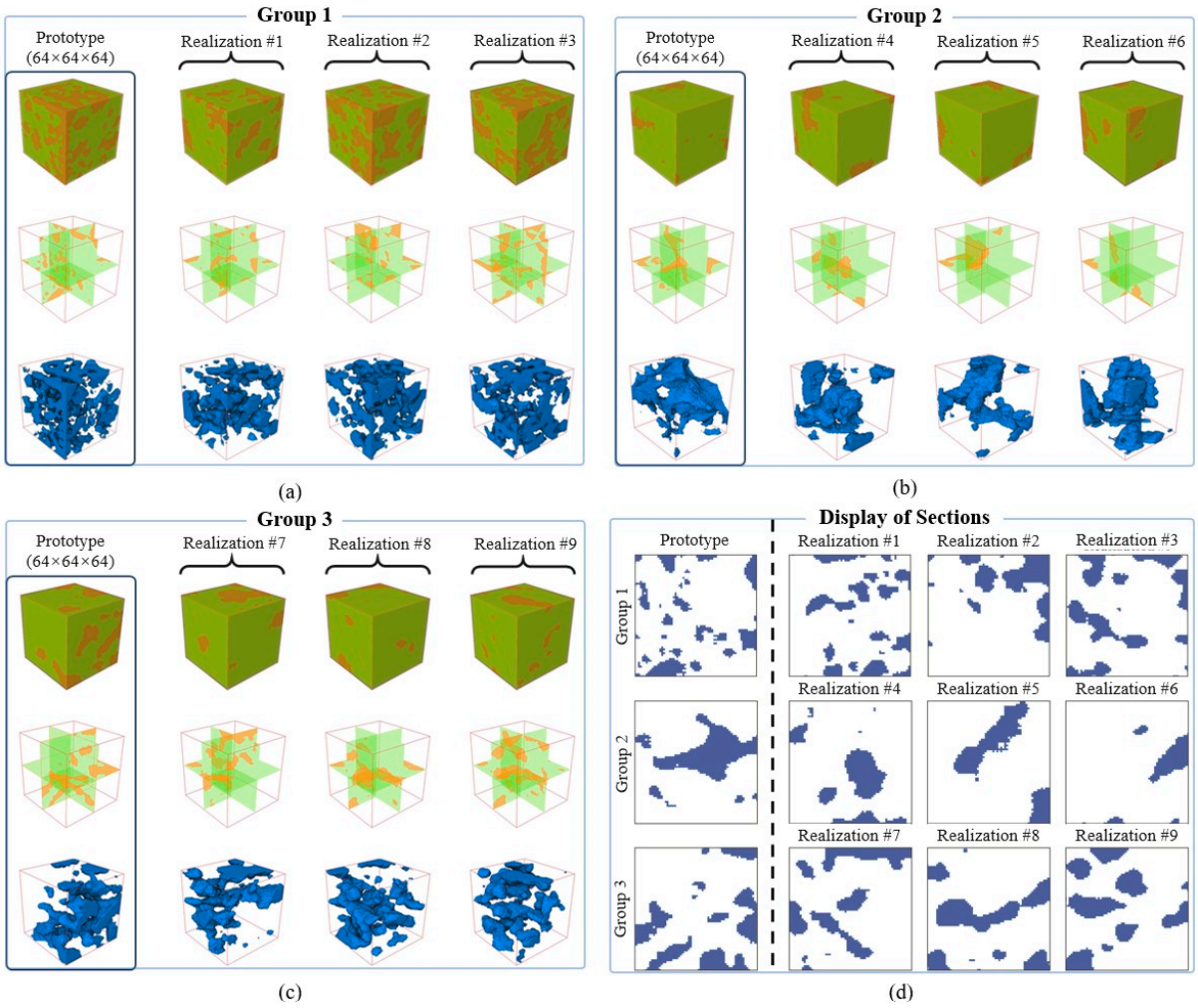


Figure 3. Three groups of size 64 × 64 × 64 of the training images shown in the Figure 1 are taken as the real data, and three randomly chosen realizations reconstructed by the WGAN-GP framework is sourced from sampling 200 random numbers of a Gaussian distribution (0,1). They are illustrated in a 3D formation in (**a**–**c**); (**d**) a set of 2D middle *z*-directional cross-section views of every training image and three corresponding realizations are listed. The pore space in all figures is represented by the dark color.

Here, we suggest that ensuring that a generative model adheres to physical laws, particularly those governing hydraulic behavior, can enable effective encapsulation of various properties such as porosity and connectivity within the model's output. This method implies that by integrating these physical laws into the model's design and training

process, the generated outputs can better simulate real-world characteristics related to fluid, potentially enhancing realism and applicability in scenarios where such properties are crucial. It underscores the importance of incorporating domain-specific knowledge into generative models to achieve more accurate and meaningful results in applications involving fluid-related phenomena. So, the hydraulic behavior of each realization is a significant concern, and verifying the permeability of each realization, akin to training a model, through pore-scale fluid-flow simulations is a key focus of this work. Pore-scale fluid-flow simulation is a process that involves data post-processing, 3D-image model reconstruction, pore space segmentation, and ultimately fluid-flow simulation. It includes the following steps:

Three-dimensional training model to a two-dimensional model: After the WGAN training process, the results of the trained model are initially stored as a three-dimensional matrix that can be read in MATLAB. A MATLAB script M file is used to slice a 3D matrix stored in a mat file into 64 2D slices. Each slice is saved as a TIFF file, and the files are sequentially named to preserve their order.

Matrix conversion to images: These grayscale TIFF images contain slice indices in their file names. The M file script includes a folder path variable specifying where these slices are saved. This preparation facilitates further processing in ImageJ.

Three-dimensional image model processing: In ImageJ, each TIFF grayscale image is imported to reconstruct a cohesive 3D grayscale model. By adjusting the threshold, this model is segmented to distinguish between pore space and solid material. The pore space is then selected for subsequent fluid-flow simulation.

Pore-scale fluid-flow simulation: Using Avizo, specifically employing the volume-average method [17], absolute permeability values are estimated in all directions. This step is critical for understanding how fluids will flow through the porous medium. The theoretical basement of this simulation is elaborated upon in the following subsection.

The whole process is described in Figure 4. The fourteen realizations are selected to carry out the CFD simulation and are compared with the training model.

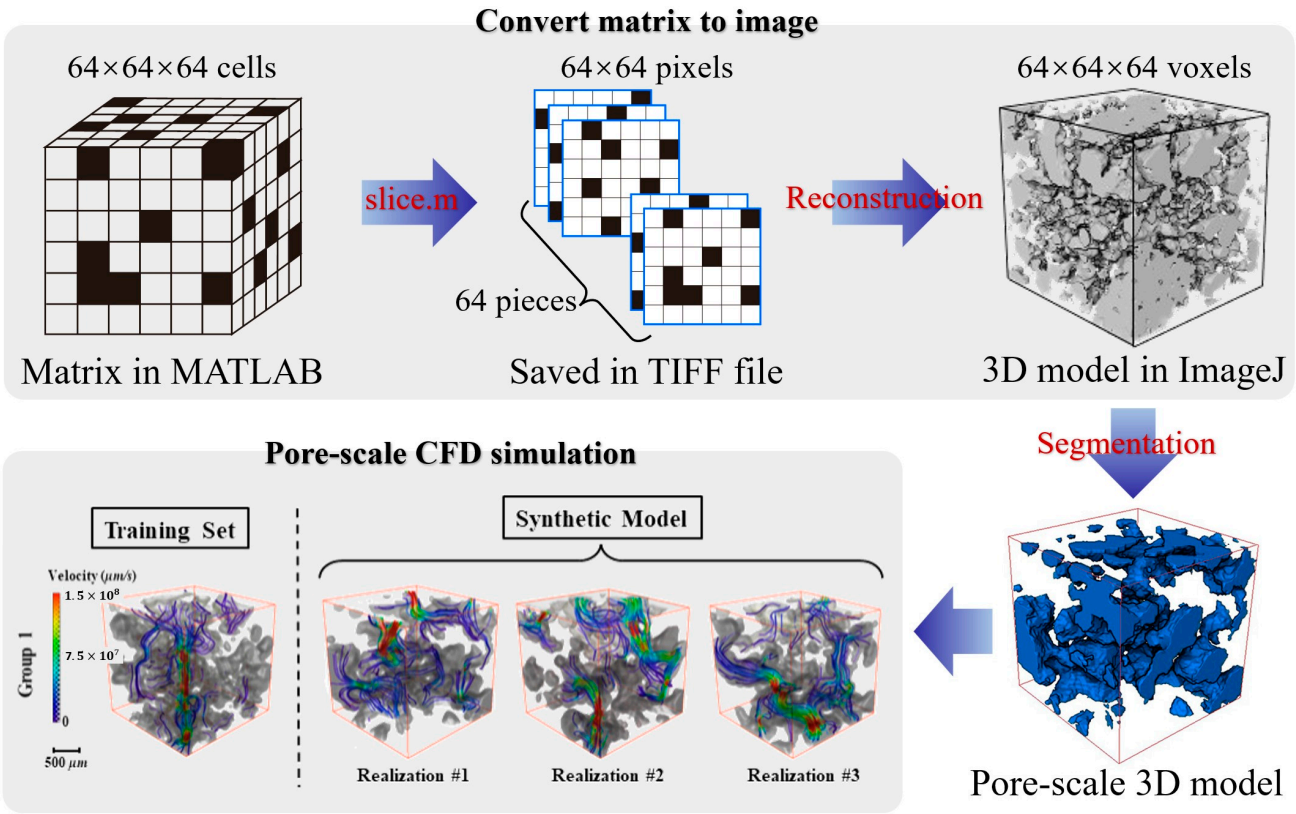


Figure 4. A scheme of pore-scale fluid simulation for determination of permeability.

4.2. Pore-Scale Fluid-Flow Simulation

A pore-scale seepage simulation was implemented to determine the absolute permeability of each gravel soil specimen based on the Avizo (2020.1) software. The controlled equation includes the Stokes flow equation for conservation of momentum and the continuity equation for conservation of mass regardless of the inertia term:

$$\begin{cases} \nabla \cdot \boldsymbol{u} = 0 \\ \mu \nabla^2 \boldsymbol{u} - \nabla P = \mathbf{0} \end{cases} \tag{6}$$

where $\nabla\cdot$ is the divergence operator; ∇ is the gradient operator; $\boldsymbol{u}$ is the velocity of the fluid in the fluid phase of the material; μ is the dynamic viscosity of the flowing fluid; ∇^2 is the Laplacian operator; and P is the pressure of the fluid in the fluid phase of the material.

Equation (6) is derived from the simplified Navier-Stokes equations, specifically for an incompressible Newtonian fluid under steady laminar flow. To solve these equations, a finite volume method is applied, as described by Whitaker (1999) [17] and Harlow and Welch (1965) [18]. The model uses a staggered grid discretization, which improves accuracy in handling the no-slip boundary condition. In this setup, the solid phase (which adheres to the no-slip condition) is separated from the fluid phase by additional grid faces that are not perpendicular to the main flow direction. Three main conditions—input pressure, output pressure, and flow rate—are specified in the model. Typically, two of these conditions are chosen, with the third is calculated from the selected two. This approach allows a comprehensive analysis of fluid dynamics, which is particularly useful in complex geometries and boundary conditions. Figure 5 illustrates the encapsulated assembly of a direct pore-scale CFD simulation for permeability estimation. It is a constraint container with slip and solid boundaries. More specifically, the Stokes problem is addressed by applying pressure boundary conditions at the inlet and outlet set at 20 kPa and 0 kPa, respectively.

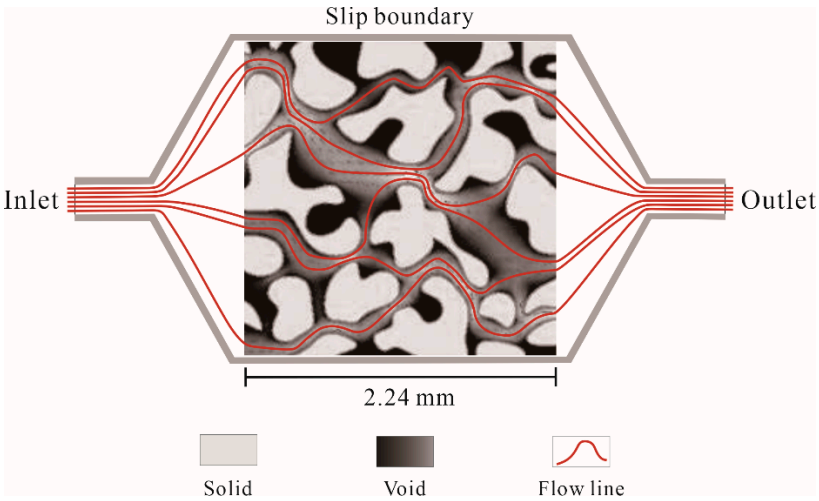


Figure 5. Assembly of a numerical model for the direct pore-scale fluid-flow simulation in our research.

According to the simulated flow rate, the permeability can be calculated using Darcy's law.

$$\kappa = \frac{Q}{S}\mu\frac{L}{\Delta p} \quad or \quad k_h = \kappa\frac{\rho g}{\mu} \tag{7}$$

where Q is the global flow rate passing through the porous medium ($m^3.s^{-1}$); S is the cross-section of the specimen that the fluid travels through (m^2); κ is the permeability (μm^2); ΔP is the pressure difference applied around the sample (Pa); and L is the length of the specimen in the flow direction (m).

The parameters of the pore-scale fluid-flow simulation used are shown in Table 2.

Table 2. Parameters for the simulated model of all realizations.

Property	Value
Fluid Dynamic Viscosity μ [Pa·s]	1.0×10^{-3}
Fluid Density ρ [kg/m^3]	1000
Porosity ε_p of Group 1 [%]	21.8
Porosity ε_p of Group 2 [%]	19.1
Porosity ε_p of Group 3 [%]	18.2
Cross-section area of a Model S [mm^2]	5.0
Length Along a Fluid-Flow Direction L [mm]	2.24
Pressure Difference Applied Δp [kPa]	20.0

4.3. Hydraulic Behaviors

Figure 6 displays the velocity fields from various realizations and their corresponding training sets. These fields reveal distinct areas of high and low velocity, with some regions showing stagnant flow. Notably, the velocity fields across different realizations within the same group exhibit similar characteristics. This consistency suggests that within each specific group, the patterns of fluid-flow and seepage behavior remain uniform across various realizations. The observed similarity in velocity fields implies that the flow through the porous medium follows a consistent pattern irrespective of the individual realization.

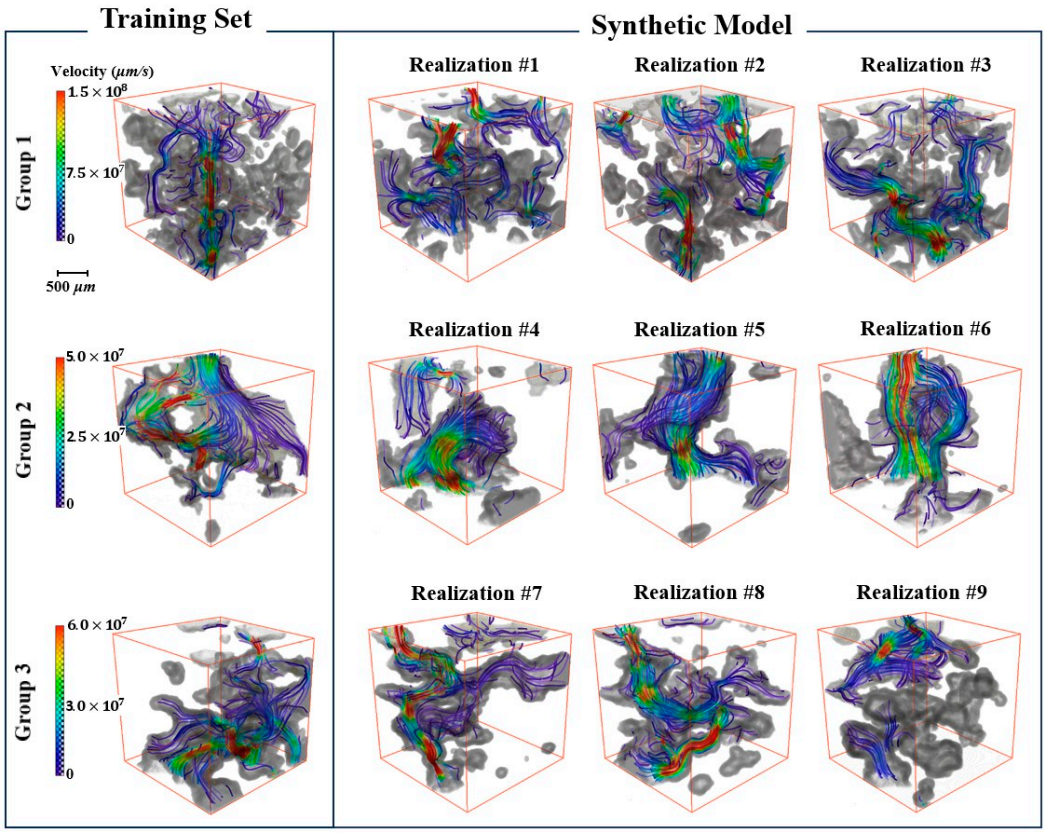


Figure 6. Three-dimensional fluid-flow diagrams in the pore space of the training set and its corresponding generation model. The colored flow lines represent changes in velocity in the pore space.

The hydraulic behaviors are characterized by the permeability, and calculations may be derived to convert intrinsic permeability κ to hydraulic conductivity k_h. Generally, there is the expression for their relationship is:

$$k_h(\text{in cm/s}) = \kappa\left(\text{in } \mu\text{m}^2\right) \times 9.7 \times 10^{-4} \quad (8)$$

Intrinsic permeability is a property of the porous medium itself, while hydraulic conductivity incorporates the properties of both the porous medium and the fluid. Intrinsic permeability quantifies the ease of fluid flow through the porous medium, while hydraulic conductivity represents the actual flow rate of the fluid under a given pressure gradient. Table 3 lists the statistical intrinsic permeability of each group of soil models based on fourteen realizations. The corresponding hydraulic conductivities are converted based on Equation (8).

Table 3. Statistics of hydraulic conductivity for each group of soil model realizations.

No.	k_{hx} [cm/s]		k_{hy} [cm/s]		k_{hz} [cm/s]	
	Prototype	**Avg. Realizations**	**Prototype**	**Avg. Realizations**	**Prototype**	**Avg. Realizations**
Group 1	0.98×10^{-2}	1.32×10^{-2}	0.91×10^{-2}	1.55×10^{-2}	1.15×10^{-2}	1.43×10^{-2}
Group 2	5.44×10^{-2}	4.40×10^{-2}	4.48×10^{-2}	4.44×10^{-2}	2.02×10^{-2}	2.29×10^{-2}
Group 3	2.59×10^{-2}	3.00×10^{-2}	2.71×10^{-2}	2.91×10^{-2}	1.15×10^{-2}	1.30×10^{-2}

Note. The average values were calculated from 14 realization values for each group based on Equation (8).

Table 3 shows the results from this research, where we present data comparing hydraulic conductivity values between the prototype's and realizations' averages (14 same scale-sized models). It indicates that there is consistency or similarity in hydraulic conductivity values between the prototype and the realizations, particularly within the standard range typical of gravel soil, at approximately 10^{-2} cm/s. This agreement is an important finding as it indicates that the measured or calculated values align well with the expected values from the prototype. The exact degree of consistency and similarity is discussed in the next section.

5. Results Analysis

5.1. Validation Analysis of Hydraulic Parameters

Based on Table 2, a series of proposed hypothesis tests aimed to determine which distribution best satisfies the permeability distribution for each group of realizations. A Box-normal plot on porosity and permeability for all realizations was constructed to assess the goodness-of-fit of the distribution for 14 separate realizations. The results of this study indicate that the distribution of permeability values generated by the GAN model shows a closer alignment with a log-normal distribution. In contrast, the distribution of porosity values, as illustrated in Figure 7, more closely resembles a normal distribution. These observations are consistent with established knowledge regarding the relationship between permeability and porosity, as outlined in the Kozney-Carman formula, shown as follows:

$$k = \frac{d_h^2}{180} \frac{{\varepsilon_p}^3}{\left(1 - \varepsilon_p\right)^2} \tag{9}$$

where d_h is the equivalent grain size of porous materials.

This formula describes a power-law relationship where porosity influences permeability, and our findings underscore this relationship through the statistical characteristics observed in the GAN-generated data. As shown in Figure 7, we included scatter points representing the porosity and permeability values of the prototype for comparison with the generated results. Our analysis reveals that these prototype values closely approximate the mean of the generated data and largely fall within the interquartile range (IQR), indicating a confidence interval of approximately 75%. Notably, the porosity value of the prototype from Group 1 stands out as an exception. We also observe that the porosity distribution of the generated models follows a normal distribution, while the permeability adheres to a log-normal distribution. These two parameters exhibit a power-law relationship, as described by the Kozeny-Carman equation.

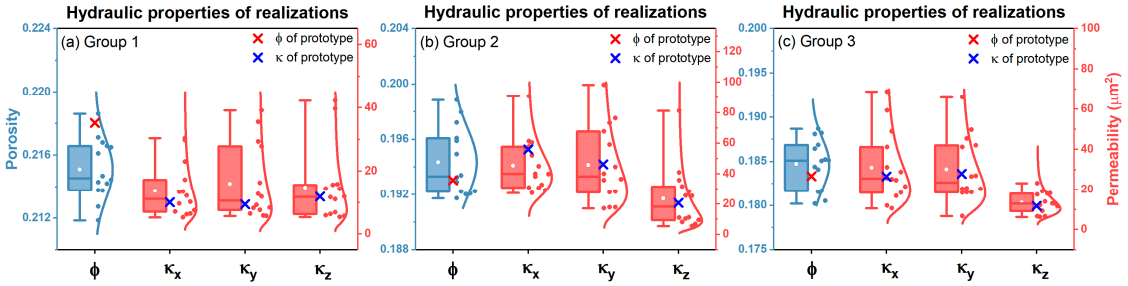


Figure 7. Box-normal plots of porosity and permeability distributions by color across realization groups. The GAN-generated permeability data exhibit a closer alignment with a log-normal distribution (depicted in light red). In contrast, porosity data show a tendency towards a normal distribution (depicted in light blue).

The hydraulic permeability of each group, as detailed in Table 3, indicates a flow rate on the order of $\times 10^{-2}$ cm/s, which is consistent with classical criteria and engineering norms. Figure 8 introduces an alternative visualization method using a skeleton model to illustrate the relationship between connected pores and the resulting flow field. While porosity indicates a medium's capacity to store fluids, the Euler characteristic assesses its connectivity, which is crucial for fluid flow. Therefore, reconstructions of porous media should accurately reflect these functional aspects to model relevant physical processes at the pore-scale effectively. Additionally, Euler numbers corresponding to these structures are provided, offering a more intuitive and practically valuable approach compared to solely relying on Euler characteristics for assessing connectivity.

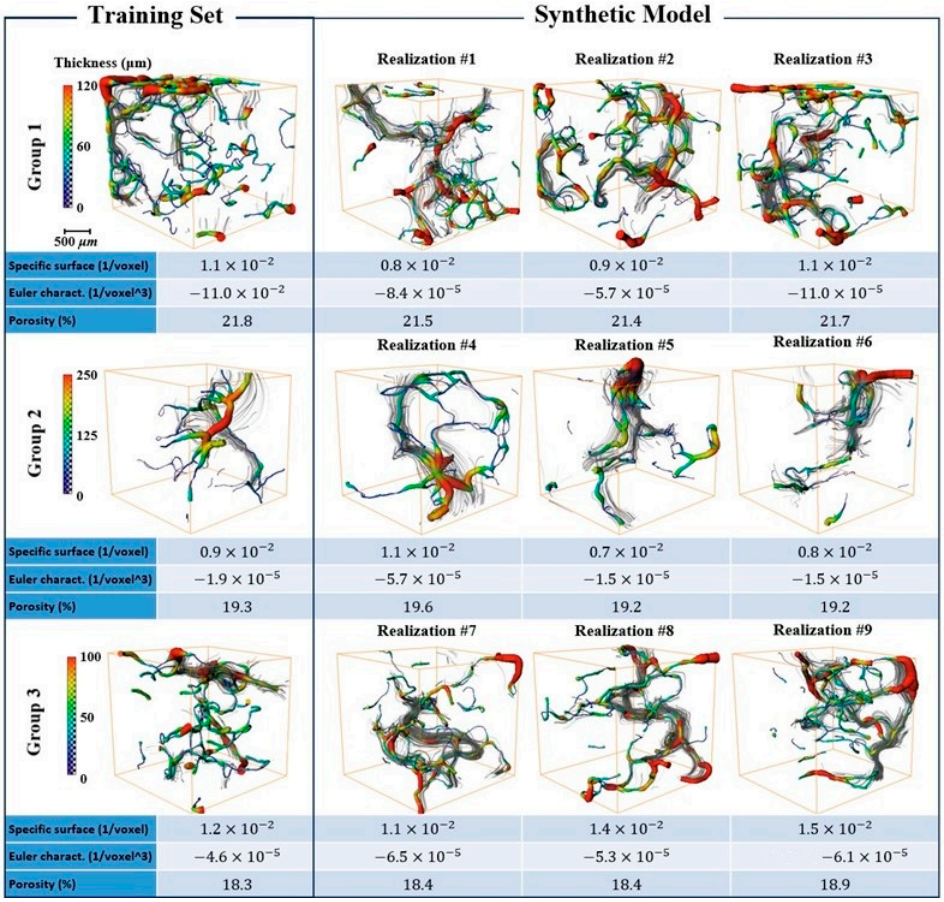


Figure 8. Diagram of fluid-flow lines of the connecting-pore skeleton structure and morphological parameters (such as specific surface area, Euler characteristic, and porosity). Fluid-flow lines flow perfectly along the paths of the connecting-pore skeleton structure.

5.2. Morphological Measures

The specific surface area of a convex body in d-dimensional space refers to the total surface area per unit volume. In porous-material constructions, the specific surface area is crucial for understanding properties such as adsorption, reaction rates, and permeability [16]. It is defined as:

$$S_V = \frac{1}{V} \int dS \tag{10}$$

where integration occurs over the void-solid interface S. The specific surface area S_V has dimensions of $\frac{1}{length}$, and its inverse allows us to define a characteristic—pore size.

The Euler characteristic provides insight into the topology of the convex body and can be used to classify different shapes and structures. It is defined as:

$$\chi = V - E + F \tag{11}$$

where χ is the Euler characteristic, V is the number of vertices, E is the number of edges, and F is the number of faces. The specific Euler characteristic refers to the Euler characteristic per unit volume or per unit area. It is a more commonly used parameter than the Euler characteristic and is defined as:

$$\chi_V = \frac{1}{4\pi V} \int \frac{1}{r_1 r_2} dS \tag{12}$$

where r_1 and r_2 are the principal radii of the curvature of the void-solid interface. We do not directly calculate the integral in Equation (12) but instead make use of a relationship for the Euler characteristic χ listed in Equation (11). We can acquire the specific surface area and Euler characteristic from the open-source image morphological software library MorphoLibJ 1.6.3 [19].

The Euler characteristic is another intuitive parameter representing the pore connectivity. The smaller it is, the better the connectivity of pores is. From the value listed in Figure 8, it can be found that the magnitude of the Euler characteristic is in the range of 10^{-5}, similar to the results in the literature provided by Mosser et al. (2017) [20] in which a slightly different type of GAN called a deep convolutional generative adversarial network (DCGAN) [11] is used for 3D generation of binary porous systems. After careful comparison, we found that our three samples are similar to the Berea sample mentioned in the literature [20] regarding specific surface area, Euler characteristics, and porosity. However, there is a significant difference in permeability, as the Berea sample measures approximately 10^{-13} m^2, while our samples fall within the range of 10^{-10} m^2, resulting in a difference of three orders of magnitude. This discrepancy arises because our sample is highly uneven, with the presence of large particles contributing to its greater permeability. Our generative model effectively captures this characteristic, distinguishing it from the more uniform Berea samples.

5.3. Two-Point Correlation Functions

The statistical correlation functions serve as vital criteria for assessing the quality of generation in machine learning, such as with WGAN-GP. These functions include the two-point correlation function and the lineal-path correlation function [21–23]. The former describes the probability that two points separated by a distance $\boldsymbol{r}$ are both in the same phase; hence, it is also referred to as the probability function. The latter provides the probability of finding a line segment with a length wholly contained within the same phase, thereby containing connectedness information along a linear path. These criteria are widely applied by many researchers [15,24,25] as they provide a good assessment of the similarity between the generative model and the training set.

The two-point probability function ${S_2}^{(i)}(\boldsymbol{r})$ is the probability that two points separated by a distance $\boldsymbol{r}$ are both in the same phase i, such as the pore space (a similar function can

be defined for the solid phase). A two-point correlation function for phase i of a multiphase system is defined as:

$$S_2^{(i)}(\boldsymbol{r_1}, \boldsymbol{r_2}) = \left\langle I^{(i)}(\boldsymbol{r_1}) I^{(i)}(\boldsymbol{r_2}) \right\rangle \tag{13}$$

where $S_2{}^{(i)}(\boldsymbol{r_1}, \boldsymbol{r_2})$ is a statistical descriptor of random media; the vectors of $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ represent the positions of two randomly selected points in phase i; the indicator function $I^{(i)}(\boldsymbol{r}) = 1$ if $\boldsymbol{r}$ belongs to phase i and is zero otherwise; and $\langle \cdot \rangle$ represents the volume average.

Lu and Torquato (1992) [21] introduced the so-called lineal-path function $L^{(i)}(\boldsymbol{r})$, which gives the probability of finding a line segment with a length wholly contained in phase i. This function is similar to $S_2{}^{(i)}(\boldsymbol{r})$ in that it requires two points at a distance $\boldsymbol{r}$ to fall in the same phase, but it also requires all points that lie on a line between them to fall in the same phase. So, it contains connectedness information along a lineal path. It is defined as:

$$L^{(i)}(\boldsymbol{r}) = \langle P(\boldsymbol{r_1}, \boldsymbol{r_2}) \rangle \tag{14}$$

Equation (14) is the area fraction of phase i measured from the projected image of a three-dimensional slab of thickness z onto a plane.

Here, we present the two-point correlation functions and the lineal-path functions along the diagonal direction, represented by d_{XY}, formed by the 45° angle between the x and y axes in Figures 9 and 10, respectively. Two MATLAB programs have been developed: one for the two-point probability function and another for the lineal-path correlation function. Each program is designed to read a specified M file containing a three-dimensional array variable that represents volumetric image data, such as prototypes or realizations of materials. These codes for MATLAB in this study also are available by open access at GitHub via https://github.com/Zhu-Bi/Gravelsoil_GAN [13], accessed on 1 May 2024.

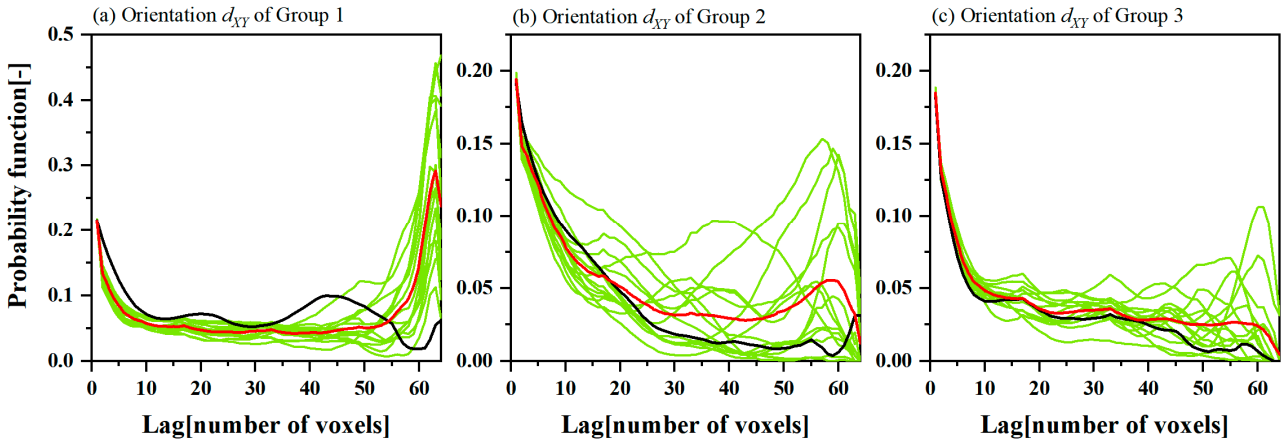


Figure 9. Probability function (PF) of finding a porous space at different 3D distances within three groups of gravel soil models. The PF is calculated for each faces along the d_{XY} of the diagonal direction. The black curves represent the PF of the three groups of the training set, which is 64 × 64 × 64 in size and derived from primitive soil prototypes. Meanwhile, the red curves show the mean PF obtained from 14 WGAN-GP realizations of size 64 × 64 × 64. The light green curve is the PF of each realization.

The two-point correlation function is a statistical tool used in physics to quantify the spatial clustering of objects within a system. It measures the excess probability of finding two objects at a given distance compared to a random distribution. Figure 9 depicts the probability of encountering a porous space at various vertical distances, quantified by the number of voxels, across three groups of generated soil realizations. These probabilities are contrasted with those of three groups of primitive soil prototypes of identical dimensions (64 × 64 × 64). The figure offers insights into the porosity and connectivity characteristics of the generated porous media, facilitating comparisons with the primitive soil prototypes to identify any discernible differences.

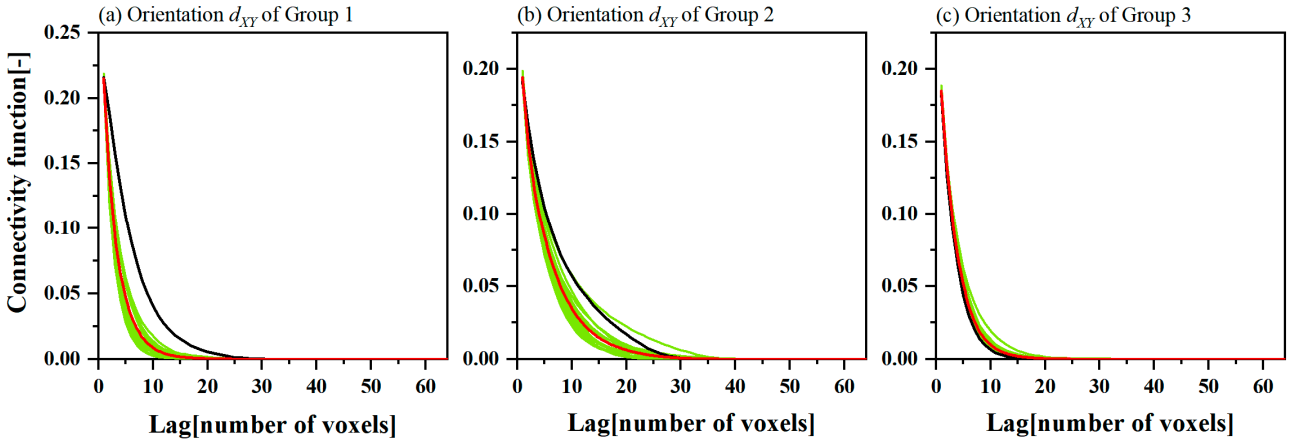


Figure 10. Lineal-path function (LF) of locating the same phase (pore in here) at different 3D distances within three groups of gravel soil models. The LF is calculated for each faces along the d_{XY} of the diagonal direction. The black curves represent the CF of the three groups of the training set, which is 64 × 64 × 64 in size and derived from primitive soil prototypes. Meanwhile, the red curves show the mean CF obtained from 14 WGAN-GP realizations of size 64 × 64 × 64. The light green curve is the LF of each realization.

Visually, the realizations appear consistent with the primitive soil prototypes. There is a best match between the average statistics of the training sets and those of the realizations (comparing the solid red and blue curves) along the x direction, with a maximum absolute deviation of 0.05. Meanwhile, a slightly poor fit can also be observed along the z direction, with a maximum absolute deviation of 0.22.

The lineal-path function, also known as the connectivity function, reveals strong agreement between the average statistics of the training sets and the realizations (solid red and blue curves) along the x, y, and z directions, with a maximum absolute deviation of approximately 0.05. Comparatively, the soil prototype exhibited slightly higher connectivity. This comparison suggests that while the WGAN model realizations may not be flawless, they appear to surpass those generated by recent algorithms such as simulated annealing (SA) and multiple-point statistics (MPS) in terms of quality. Moreover, the WGAN model demonstrates superior efficiency when generating a large number of model realizations.

6. Discussion

Our study demonstrates the effectiveness of our proposed WGAN-GP approach. Figure 3 provides a visual comparison between the real soil prototype and the generated realizations. Upon examination of both 3D models, it is challenging to distinguish between the generated microstructure and the real microstructure, indicating a high degree of similarity. Through fluid-flow simulations, we quantitatively assessed the hydraulic conductivity of the generated model and found them to closely resemble those of the original soil prototype. The pore distribution characteristics of the generated model follow a normal distribution, while the permeability distribution characteristics adhere to a log-normal distribution. This observation suggests a power-law relationship between pore and permeability distributions, which is consistent with the general principles of porous media (as referenced in Figure 7). This relationship was further confirmed by the literature of Nguyen et al. (2022) [26] in which a slightly different type of GAN and actor-critic (AC) reinforcement learning are used for 3D generation of porous systems. These findings indicate that our generator effectively captures the essential morphological features of natural soil. It is capable of reproducing the intricate micro-geometry of soil, encompassing a variety of realistic shapes of voids and matrix structures in the generated microstructures (as depicted in Figures 8–10).

An important limitation of our approach is that the 64 × 64 × 64-voxel patches from the soil model are too small to capture the geometric characteristics of coarse grains.

Despite fine grains dominating the hydraulic properties of the gap-grading gravel soil, the hydraulic effect of coarse particles cannot be overlooked. To address this, we propose gradually expanding the 3D patch size, such as using 128 × 128 × 128-voxel, 192 × 192 × 192-voxel patches, creating a series of pyramid representations, which is similar to the progressive training workflow [15,27]. This approach aims to overcome the issue of small-scale patches lacking sufficient detail in modeling gravel soil prototypes. However, this will increase the demand on training hardware and software. For instance, on a GPU like the GeForce RTX 3090, training could take between 9 to 12 h, while training on CPUs would likely take substantially longer. A feasible strategy is to initially learn the basic soil prototype using 2D faces along one direction with larger patches, such as 700 × 700 pixels, and then estimate voxel values in the other directions based on geostatistical variogram functions. This method could optimize training time, akin to techniques used in existing literature [28–30].

Laloy et al. (2017, 2018) [9,25] introduced an intriguing method for rapidly generating a series of 2D or 3D unconditional realizations from a channelized aquifer training image using the Spatial Generative Adversarial Network (SGAN) they proposed. A key characteristic of the proposed SGAN is its definition of a (very) low-dimensional parameterization, which facilitates efficient probabilistic inversion using state-of-the-art Markov chain Monte Carlo (MCMC) methods. Essentially, it represents a machine learning strategy rooted in a multi-point geostatistical (MPS) method for inversion parameters. In the early stages of training, it does not begin with a uniform distribution latent variable $\boldsymbol{Z}$ but rather a prior probability variable $\boldsymbol{\theta}$, where $\boldsymbol{\theta} = \boldsymbol{Z}$ [24]. Consequently, the time required for training is significantly reduced, and training a SGAN takes 3–12 h. Our approach aimed at porous media generation indeed takes more time in the training phase than probabilistic retrieval based on the MPS method. If we adopt this method and focus solely on training and learning the random function parameters of the realistic model before embedding the adversarial network, we can significantly reduce the generation time. This approach offers a new perspective on improving the efficiency of synthetic image generation. However, the challenge lies in the applicability of MPS to porous media, even though it has been successfully implemented in channel-like models.

Song et al. (2021) [15] trained GANs for geological faces modeling using both progressive and conventional training workflows and then evaluated the performance of the two trained generators. This approach provides a novel solution to the challenges of size and scale in geological modeling. In a subsequent paper, Song et al. (2022) [31] identified the limitations of the MPS method, noting that its geological model representation is restricted to multi-point statistical information derived from a narrow range of training images. Consequently, it lacks a comprehensive abstraction and summary of geological model features across a broader area. When well point data are dense or when a geological structure is complex, the performance of the multi-point geostatistical method can be suboptimal, especially in cases involving intermittent channel faces reservoirs. Conditional GANs may offer a promising solution to address the challenges of sophisticated micro-structural soil modeling. A similar study by Zheng and Zhang (2020) [32] compared results generated by conditional GANs with those produced by MPS, highlighting the potential advantages of GANs for small-scale micro-porous modeling, which contrasts with the applicability of MPS for large-scale geological modeling.

By integrating empirical testing, field trials, benchmarking, and user studies, the proposed WGAN approach can be effectively validated in real-world scenarios. Additionally, its applicability across various domains showcases the versatility of WGANs in solving complex problems while ensuring that the outputs remain relevant and practical. To aid engineers in practicing the proposed approach, this work will be carried out in collaboration with partner companies and through our website to validate and implement our proposed model in an industrial setting. To establish real-world applicability, we can consider additional validation methods: (1) conducting physical experiments that replicate simulation conditions and modeling more experimental and in-situ sample data

to directly demonstrate the approach's efficacy, assess its performance, and gather data on its robustness and accuracy; and (2) comparing our WGAN approach with established methods through cross-validation to highlight its advantages and limitations.

7. Conclusions

In conclusion, leveraging advanced imaging techniques and GAN technology with gradient-differentiable Wasserstein distance and penalty functions enables precise reconstruction of small- or pore-scale media. This approach not only overcomes limitations of traditional methods but also ensures stable training and high fidelity in generating virtual samples. Such advancements are pivotal for enhancing scientific understanding and modeling capabilities across various disciplines, including soil science and material science. Some conclusions are as follows:

The machine learning image generation model based on WGAN-GP utilizing MATLAB's capability for matrix-based image processing combined with PyTorch programming functionalities demonstrates efficiency, reproducibility, and robustness. It shows promising practical applications in reconstructing gravel rock core samples in this research context.

The generated model has been validated through Computational Fluid Dynamics (CFD) simulations. It accurately reproduces the hydraulic properties, specifically the permeability in the range of 10^{-2} cm/s typical of gravel soil, similar to the original samples. Additionally, it closely aligns with their morphological characteristics and indicators. This ability allows it to reflect the original permeability characteristics of gravel soil, showcasing the fine-scale features of microstructures and micropores in a very intuitive manner.

An average training time of 9 h on a basic laptop configuration is commendable. Future research efforts should focus on enhancing the granularity of gravel soil grading. This can be achieved by progressively expanding the region of 3D patches in the training set or transforming a large region of 2D training images into 3D synthetic representations. These approaches aim to capture finer details and improve the accuracy of the soil grading process.

Currently, the main limitations of this study are reflected in the following aspects: (1) generalizability, (2) data dependency, (3) computational complexity, and (4) evaluation metrics. Future developments will focus on the following directions: (1) enhanced validation, (2) data augmentation, (3) optimization techniques, and (4) broader applications.

Author Contributions: Conceptualization, B.Z.; Methodology, X.-G.H.; Software, X.-G.H.; Validation, B.Z. and X.-G.H.; Formal analysis, B.Z.; Investigation, X.-G.H.; Resources, B.Z.; Data curation, X.-G.H.; Writing—review & editing, B.Z.; Visualization, X.-G.H.; Project administration, B.Z.; Funding acquisition, B.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (NSFC41462014).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Acknowledgments: We acknowledge the μ-CT image scanning assistance provided by personnel of the Key Lab of New Processing Technology for Nonferrous Metals and Materials Ministry of Education and Guangxi Key Laboratory of Geomechanics and Geotechnical Engineering. Guilin University of Technology, during field instrumentation, troubleshooting and data collection in this study. The authors also appreciate Xi Yang, Yunnan Aerospace Engineering Geophysical Detecting Co., Ltd., Kunming 650217, China, for his strong support and supervision of this paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zhu, B.; Zhang, J.; Zeng, X.-B. Analysis of internal erosion in granular soil during deep excavation with a water-inrush incident in a covered karst area. *Tunn. Undergr. Space Technol.* **2023**, *132*, 104932. [CrossRef]
2. Gupta, R.; Salager, S.; Wang, K.; Sun, W. Open-source support toward validating and falsifying discrete mechanics models using synthetic granular materials-Part I: Experimental tests with particles manufactured by a 3D printer. *Acta Geotech.* **2019**, *14*, 923–937. [CrossRef]
3. Glaessgen, E.; Stargel, D. The digital twin paradigm for future nasa and us air force vehicles. In Proceedings of the 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference 20th AIAA/ASME/AHS Adaptive Structures Conference 14th AIAA, 1818, Honolulu, HI, USA, 23–26 April 2012.
4. Zohdi, T. A machine-learning framework for rapid adaptive digital-twin based fre-propagation simulation in complex environments. *Comput. Methods Appl. Mech. Eng.* **2020**, *363*, 112907. [CrossRef]
5. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial networks. In Proceedings of the Twenty-Eight Annual Conference on Neural Information Processing Systems (NIPS), Montreal, QC, Canada, 7–12 December 2014.
6. Nourani, M.H.; Kisi, O.; Zounemat-Kermani, M. Evaluating the performance of a generative adversarial network for simulating soil moisture. *Catena* **2020**, *186*, 104382.
7. Wang, Z.; Simoncelli, E.P.; Bovik, A.C. Multiscale structural similarity for image quality assessment. In Proceedings of the Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, Pacific Grove, CA, USA, 9–12 November 2003; pp. 1398–1402.
8. Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; Courville, A.C. Improved Training of Wasserstein GANs. *Adv. Neural Inf. Process. Syst.* **2017**, 5767–5777. [CrossRef]
9. Laloy, E.; Herault, R.; Jacques, D.; Linde, N. Training-image based geostatistical inversion using a spatial generative adversarial neural network. *Water Resour. Res.* **2018**, *54*, 381–406. [CrossRef]
10. Zhu, B.; Hu, X.G. CT Image for Gravel Soil: 22 April 2024 Release [Dataset]. Figshare. 2024. Available online: https://figshare.com/articles/figure/CT_image_for_gravel_soil/25665360 (accessed on 1 May 2024).
11. Radford, A.; Metz, L.; Chintala, S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *arXiv* **2015**, arXiv:1511.06434.
12. Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein generative adversarial networks. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 298–321.
13. Zhu, B.; Hu, X.G. Zhu-Bi/Gravelsoil_GAN: WGAN-Enabled Machine Learning for Gravel-Soil Generation, Version 1.0.0 [Softwate]. Zenodo. 2024. Available online: https://zenodo.org/doi/10.5281/zenodo.11114029 (accessed on 1 May 2024).
14. Torquato, S.; Stell, G. Microstructure of two-phase random media. I. The n-point probability functions. *J. Chem. Phys.* **1982**, *77*, 2071–2077. [CrossRef]
15. Song, S.; Mukerji, T.; Hou, J. Geological faces modeling based on progressive growing of Generative Adversarial Networks (GANs). *Comput. Geosci.* **2021**, *25*, 1251–1273. [CrossRef]
16. Scholz, C.; Wirner, F.; Klatt, M.A.; Hirneise, D.; Schröder-Turk, G.E.; Mecke, K.; Bechinger, C. Direct relations between morphology and transport in Boolean models. *Phys. Rev. E* **2015**, *92*, 043023. [CrossRef]
17. Whitaker, S. *The Method of Volume Averaging*; Kluver Academic Publishers: Dordrecht, The Netherlands, 1999; pp. 154–196.
18. Harlow, F.H.; Welch, J.E. Numerical calculation of time-dependent viscous incompressible flow of fluid with a free surface. *Phys. Fluid.* **1965**, *8*, 2182–2189. [CrossRef]
19. Legland, D.; Arganda-Carreras, I.; Andrey, P. MorphoLibJ: Integrated library and plugins for mathematical morphology with ImageJ. *Bioinformatics* **2016**, *32*, 3532–3534. [CrossRef] [PubMed]
20. Mosser, L.; Dubrule, O.; Blunt, M.J. Reconstruction of three-dimensional porous media using generative adversarial neural networks. *Phys. Rev. E* **2017**, *96*, 043309. [CrossRef] [PubMed]
21. Torquato, S.; Beasley, J.D.; Chiew, Y.C. Two-point cluster function for continuum percolation. *J. Chem. Phys.* **1988**, *88*, 6540–6547. [CrossRef]
22. Lu, B.; Torquato, S. Lineal-path function for random heterogeneous materials. *Phys. Rev. A* **1992**, *45*, 922–929. [CrossRef]
23. Torquato, S. *Random Heterogeneous Materials: Microstructure and Macro-Scopic Properties*; Springer: New York, NY, USA, 2002.
24. Lemmens, L.; Rogiers, B.; De Craen, M.; Laloy, E.; Jacques, D.; Huysmans, M.; Swennen, R.; Urai, J.L.; Desbois, G. *Effective Structural Descriptors for Natural and Engineered Radioactive Waste Confinement Barrier*; [EGU 2017-Poster 14094]; The European Geophysical Union (EGU) General Assembly: Vienna, Austria, 2017.
25. Laloy, E.; Herault, R.; Lee, J.; Jacques, D.; Linde, N. Inversion using a new low-dimensional representation of complex binary geological media based on a deep neural network. *Adv. Water Resour.* **2017**, *110*, 387405. [CrossRef]
26. Nguyen, P.C.H.; Vlassis, N.N.; Bahmani, B.; Sun, W.; Udaykumar, H.S.; Baek, S.S. Synthesizing controlled microstructures of porous media using generative adversarial networks and reinforcement learning. *Sci. Rep.* **2022**, *12*, 9034. [CrossRef] [PubMed]
27. Karras, T.; Laine, S.; Aila, T. A Style-Based Generator Architecture for Generative Adversarial Networks. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–17 June 2019; IEEE: New York, NY, USA, 2019.

28. Tahmasebi, P.; Sahimi, M.; Caers, J. Ms-ccsim: Accelerating pattern-based geostatistical simulation of categorical variables using a multi-scale search in fourier space. *Comput. Geosci.* **2014**, *67*, 75–88. [CrossRef]
29. Tahmasebi, P. HYPPS: A hybrid geostatistical modeling algorithm for subsurface modeling. *Water Resour. Res.* **2017**, *53*, 5980–5997. [CrossRef]
30. Mariethoz, G.; Renard, P.; Straubhaar, J. The Direct Sampling method to perform multiple-point geostatistical simulations. *Water Resour. Res.* **2010**, *46*, W11536. [CrossRef]
31. Song, S.; Shi, Y.Q.; Hou, J. Review of a Generative Adversarial Networks (GANs)-based geomodelling method. *Pet. Sci. Bull.* **2022**, *1*, 34–49. [CrossRef]
32. Zheng, Q.; Zhang, D. Digital rock reconstruction with user-defined properties using conditional generative adversarial networks. *Transp. Porous Media* **2022**, *144*, 255–281. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.