



Constructing large-scale high-fidelity fracture networks based on generative AI

Mengmeng Nie^a, Xuhai Tang^{a,*}, Fei Gao^a, Quansheng Liu^a, Jiangmei Qiao^{a,b,**}

^a School of Civil Engineering, Wuhan University, Wuhan, 430072, China

^b Wuhan University Shenzhen Research Institute, Shenzhen, 518108, China

ARTICLE INFO

Keywords:

Fracture network upscaling
High-fidelity generation
Generative artificial intelligence
Generative adversarial network (GAN)
Patch-by-patch generative paradigm

ABSTRACT

Accurate fracture network modeling is crucial for many fields, such as geothermal resources exploitation, underground infrastructure construction and nuclear waste disposal. However, only small-scale fracture network images can generally be obtained in field work, which is insufficient for engineering-scale analysis. Therefore, based on generative artificial intelligence (AI), we introduced a novel algorithm, termed Upscaling-GAN, to generate large-scale high-fidelity fracture networks by directly learning from small-scale fracture network images collected in the field. The algorithm employs a two-stage generation process: (1) Generative Adversarial Network (GAN) model is trained to generate small-scale fracture networks; (2) based on the patch-by-patch generation paradigm, a large-scale fracture network is generated from these small-scale fracture networks, which are obtained by applying the trained model. The achieved AI-generated fracture networks have significant advantages: 1) compared to fractal-geometry-based methods or stochastic discrete fracture network model, the present method can exactly characterize the geometrical characteristics and topological structures of natural fracture systems; 2) compared to previous GAN-based method, our algorithm can rigorously capture the spatial variability of fracture apertures by directly learning from raw fracture network images without laborious preprocessing procedures such as binarization and skeletonization; 3) when the size of the target image increases, the GPU memory consumption remains nearly unchanged. And the effectiveness of the Upscaling-GAN model in fracture network modeling is rigorously validated through qualitative and quantitative evaluations.

1. Introduction

Natural fractures are widely distributed in rock masses, which profoundly alter the mechanical properties, hydraulic conductivity, and mass-transport behavior of rock masses.^{1–3} Therefore, accurate fracture network modeling is indispensable for understanding mechanisms behind these physical processes and predicting corresponding behaviors. The complexity of natural fracture networks is manifested in the fracture geometries, spatial arrangements, orientations, and size distributions.^{4–6} A comprehensive characterization and high-fidelity modeling of the complicated fracture systems are thus of paramount importance for many engineering applications, such as tunnel excavation, enhanced geothermal systems (EGS) and CO₂ geological sequestration.^{7–9} Over the past several decades, researchers have carried out extensive studies in terms of discrete fracture network (DFN) modeling and various methods or techniques have been proposed.^{10–13} The most

widely used strategies or approaches with respect to DFN modeling mainly involve the following categories: (1) stochastic DFN methods¹⁰; (2) AI-based modeling methods.¹⁴

In the context of stochastic DFN methods, the probability distributions of different geometrical attributes, which can be obtained according to outcrop surveys, well logs and related data, are utilized to generate DFNs.^{15,16} Poisson model^{17,18} is a classical and commonly used stochastic models. The process of constructing fracture networks using the Poisson model can generally be summarized in two main steps: firstly, the fracture barycentric coordinates can be determined by employing a spatial Poisson process, which will result in an independent spatial distribution across the domain; secondly, 2D or 3D discrete fracture networks can subsequently be constructed by randomly sampling geometric properties from predefined probability density functions.^{19–22} However, owing to the adoption of linear or planar fracture geometries and the assumption of uniform distribution of

* Corresponding author.

** Corresponding author. School of Civil Engineering, Wuhan University, Wuhan, 430072, China.

E-mail addresses: xuhaitang@whu.edu.cn (X. Tang), jiangmei_qiao@whu.edu.cn (J. Qiao).

fracture locations, fracture networks generated by the Poisson model are overly simplified and often exhibit significant discrepancies from those observed in natural systems.¹⁰ Therefore, many improvements have been employed on the basis of Poisson model. For example, an innovative sequential parent-daughter Poisson point process integrating mechanical interactions between fractures was developed by Bonneau et al.²³ for simulating discrete fracture networks. Hekmatnejad et al.²⁴ proposed a 3D Cox-Boolean DFN model, modeling the fracture center intensity as a spatial random field (potential field) to characterize spatial variability and inferring 3D potential field parameters from 1D/2D observations, which overcame the limitation of constant intensity in the traditional Poisson model. In order to better capture or account for spatial correlations and scale-invariance properties inherent in natural fracture networks, fractal-geometry-based methods were developed.^{25–28} The fractal dimension, which can be computed by box-counting method,²⁹ can effectively characterize the clustering behavior, spatial dispersion, and spatial complexity of fracture networks. Overall, stochastic DFN methods exhibit substantial flexibility and computational efficiency, allowing rigorous uncertainty quantification through repeated stochastic realizations.³⁰ However, their primary limitation lies in insufficient consideration of geological processes and mechanical interactions, thereby restricting their capacity to authentically reproduce real-world geological conditions.^{30,31}

In recent years, the rapid development of generative artificial intelligence (generative AI) has opened up new conceptual and methodological pathways for generating fracture networks. Specially, significant process in the field of texture synthesis^{32–37} can provide a robust foundation for advancing the DFN modeling. Bergmann et al.³⁸ proposed a model called Periodic Spatial Generative Adversarial Network (PSGAN) to learn texture manifolds and generate high-quality texture images. Rodriguez-Pardo and Garces³⁹ introduced SeamlessGAN, which generates seamlessly tileable textures from a single input sample. Abdellatif et al.³² proposed a GAN model to generate infinite-size textures in a patch-by-patch manner when only given small-sized images. Recently, GANs have also been applied in the field of fracture network modeling. Chen et al.⁴⁰ adopted variational auto-encoder and generative adversarial network (VAE-GAN) to characterize complex fracture networks within a stochastic inversion framework, thereby enhancing the efficiency and accuracy of fracture network inversion. On the basis of SinGAN,⁴¹ Zhou et al.¹⁴ developed a framework, termed FracGen, for reconstructing and upscaling natural fracture networks, which has promoted the application of GAN in fracture network modeling. In their method, they applied preprocessing operations such as thresholding, binarization and skeletonization to the original images. While effective for the characterization of topological structures and morphological features such as curvature and bending, these preprocessing steps made it impossible to capture variations in fracture aperture. In addition, GPU memory limitations may be encountered in their framework when generating large-scale fracture network images. Teng et al.⁴² developed the Wasserstein generative adversarial network with gradient penalty (WGAN-GP) to generate high-fidelity discrete fracture networks from low-dimensional latent spaces. Likewise, challenges related to the explicit representation of fracture aperture and the limited image resolution remain unresolved.

In this work, based on the model proposed by Abdellatif et al.,³² a novel framework, termed Upscaling-GAN, is developed to generate large-scale, high-fidelity fracture networks by directly learning from the original small-scale fracture network images. The remainder of the paper is organized as follows. Section 2 mainly introduces the algorithm of the proposed model. The training data, model parameters and training parameters are described in Section 3. Evaluation of the training process based on loss curves and the analysis of model scalability are introduced in Section 4. Qualitative and quantitative evaluations of the fracture networks generated by our Upscaling-GAN model are demonstrated in Section 5. The comparative analysis of our proposed model, vanilla model and traditional fractal-geometry-based modeling

method are meticulously described in Section 6. Lastly, the main contributions of the present model are summarized in Section 7.

2. Methodology

We have developed a platform for constructing digital rocks using generative AI, referred to as the AI-Geo platform. The AI-Geo platform is designed to generate digital rock models from text-format data and then apply numerical simulations to solve key physical parameters (e.g., mechanical and hydraulic properties). The “text-format data” includes engineering information, such as geophysical imaging data and well-logging data. At present, the AI-Geo platform consists of four modules:

AIG-Knowledge Module: Based on knowledge-transfer algorithm, this module fuses prior knowledge from similar engineering projects with the limited on-site monitoring data to fill in missing information and construct the geotechnical model for the target project.

AIG-2D3D: Automatically generates corresponding 3D geotechnical models from high-resolution 2D images.

AIG-Upscaling: Scale small-scale models up to large-scale models that meet engineering requirements.

AIG-Physics: Based on cutting-edge generative AI algorithms, physical constraints are incorporated to calibrate and optimize the AI-generated model, thereby avoiding hallucination issues and improving the accuracy of the generated results.

This work primarily focuses on the algorithmic development of the AIG-Upscaling module. In order to generate fracture networks that satisfy our requirements, some improvements to the vanilla model, which was proposed by Abdellatif et al.,³² are implemented. And our primary improvements focus on the internal architecture of the discriminator and loss function, which will be systematically elaborated in this section. The large-scale, high-fidelity fracture networks can be generated through a two-stage generation process: (1) GAN model is trained to learn features from the original small-scale fracture network image; (2) under the patch-by-patch generative paradigm, the trained GAN model is repeatedly applied to generate a large number of small-scale fracture network images, which are then stitched together to produce the desired large-scale fracture network image.

2.1. Stage 1: training GAN model according to original small-scale fracture network image

The overall architecture of the GAN model for the first stage is shown in Fig. 1, which consists of the generator (G) and discriminator (D). In the first stage, $N * N$ small-scale patches are produced simultaneously. The size of each patch is set to $h \times w$. During the training process, the GAN model progressively learns to generate patches and to seamlessly assemble them into a larger image.

2.1.1. Generator

The input data to the generator consist of three parts, i.e., spatial noise z , local padding p and a stochastic spatial map M . According to our experiments, we found that omitting the stochastic spatial map can significantly accelerate the training process without affecting the quality of the generated outputs. This observation is consistent with the finding reported by Abdellatif et al.³² Therefore, stochastic spatial map is disregarded in this study. The first input, spatial noise z , can be obtained by sampling from a normal distribution. The second input is local padding. The padding strategy used in this study, including local padding and replicate padding, is illustrated in Fig. 2. In convolutional operations, padding refers to adding extra pixels (typically zero-valued pixels) along the edges of the images or feature maps. However, adding zero-valued pixels around the images or feature maps leads to mismatches or visible discontinuities at the seams between adjacent patches. To achieve seamless transitions during the stitching process, local padding uses the values from adjacent patches to fill in the padded regions. As shown in Fig. 2, the central patch's padding regions are populated with pixels

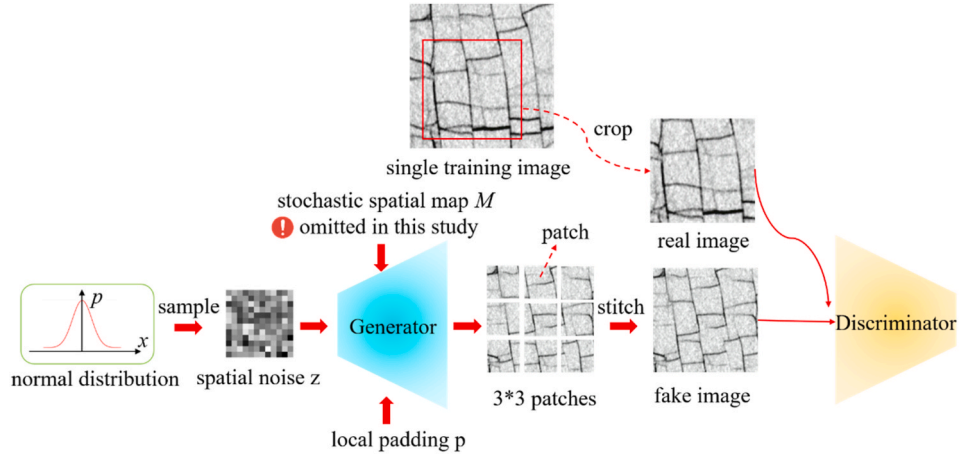


Fig. 1. The overall architecture of the GAN model for the first stage.

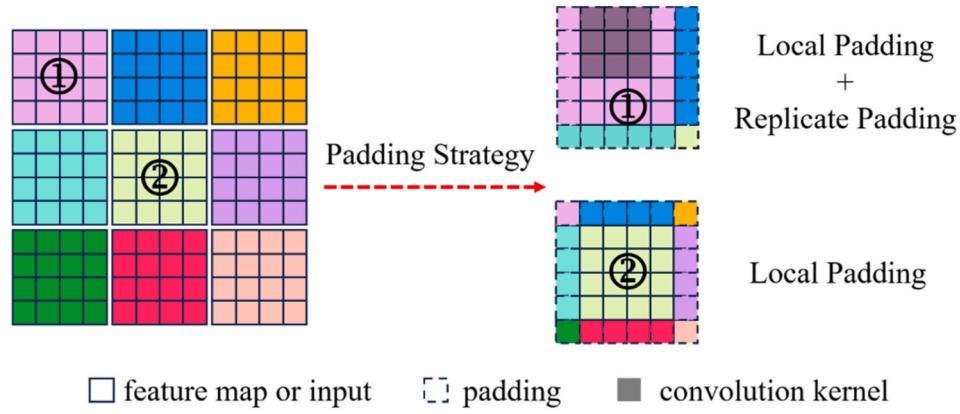


Fig. 2. Padding strategy used in this study.

taken from all eight neighboring patches. This strategy enables each generated patch to incorporate contextual information from

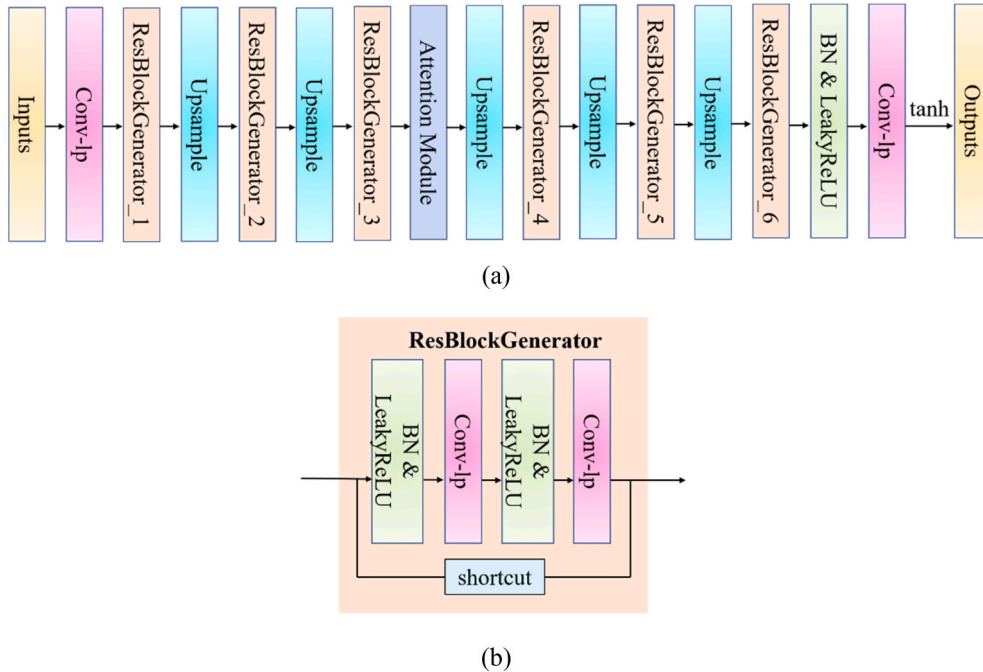


Fig. 3. The detailed architectures of the generator: (a) the architecture of generator; (b) ResBlockGenerator module.

surrounding regions, thus producing seamlessly tiled, continuous large-scale images. For boundary pixels in the outer patches, replicate padding is used as there is no adjacent patch. In this paper, local padding is applied to the input noise z and feature maps of generator layers.

The architecture of the generator is shown in Fig. 3(a). It mainly consists of the following modules: 1) the module of *Conv-lp* is designed to perform convolution operation with local padding; 2) Batch Normalization and LeakyReLU activation function are applied in the *BN & LeakyReLU* module; 3) the *Attention Module* implements the self-attention mechanism on the feature map, enabling each spatial position to refine its representation based on information from all other positions, thereby explicitly modeling long-range spatial dependencies and global structural relationships; 4) the *Upsample* module performs spatial upsampling on feature maps, increasing their resolution to a larger size; 5) *ResBlockGenerator_1*, *ResBlockGenerator_2*, *ResBlockGenerator_3*, *ResBlockGenerator_4*, *ResBlockGenerator_5* and

ResBlockGenerator_6 share the same architecture shown in Fig. 3(b).

2.1.2. Discriminator

The detailed architectures of the discriminator are shown in Fig. 4. In the vanilla model, PatchGAN⁴⁴ is adopted as the discriminator architecture. However, according to our experiments, we observed that the dual-PatchGAN architecture, which is shown in Fig. 4(a), achieves superior performance. Therefore, it is adopted in this paper. This architectural configuration also appears in the work of Sun et al.,⁴⁵ where the authors term it a Hybrid-discriminator (Hybrid D). In this paper, both PatchGANs share exactly the same architecture but use different convolution operations. In the first PatchGAN, all convolutional layers use the standard convolution operation. In the second PatchGAN, dilated convolution operation is applied. Dilated convolution enlarges the receptive field by inserting gaps between kernel elements while keeping the number of parameters unchanged.⁴³ This enables the

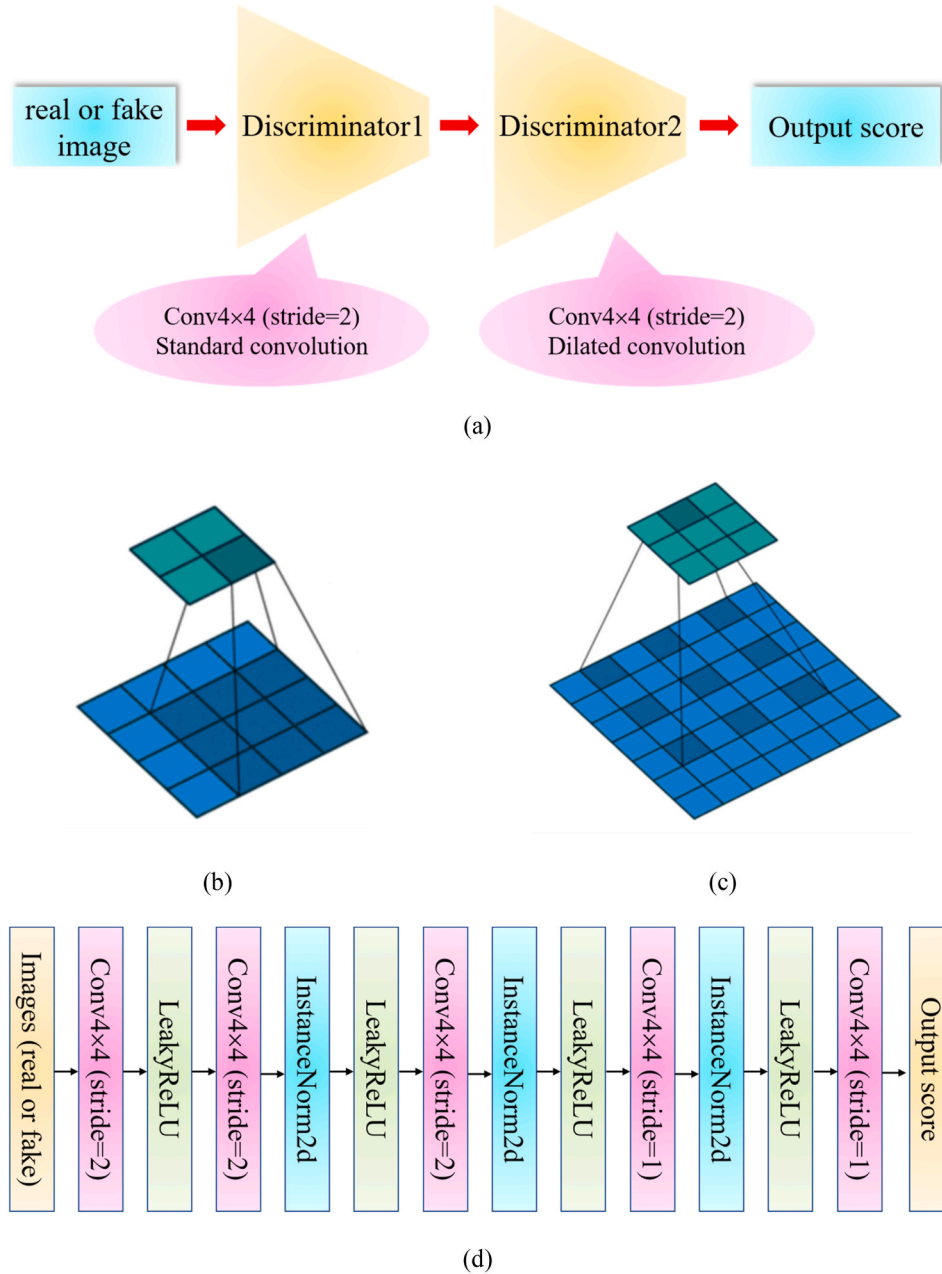


Fig. 4. The detailed architectures of the discriminator: (a) the overall architecture of discriminator (Dual-PatchGAN architecture); (b) standard convolution⁴³; (c) dilated convolution⁴³; (d) the architecture of PatchGAN.

network to capture broader contextual information without reducing spatial resolution, thereby enhancing its ability to represent multi-scale structures. The computational principles of standard convolution and dilated convolution are shown in Fig. 4(b) and (c), respectively. Fig. 4(d) shows the architecture of PatchGAN. The $Conv4 \times 4$ module represents the 2D convolution with the kernel size of 4. And the Instance Normalization is applied in the layer of *InstanceNorm2d*.

2.1.3. Loss function

The GAN model is trained through a two-player minimax game in which the generator G learns to synthesize realistic samples that imitate the real data distribution while the discriminator D distinguishes generated samples from real data.⁴⁶ Through alternating optimization, G and D are improved progressively until the system converges toward an equilibrium where G yields realistic samples and D can no longer successfully distinguish them from real data.

In the work of Abdellatif et al.,³² the sigmoid-based binary cross entropy was used as loss function. However, according to our experiments, we found that the hinge loss with a margin-based objective leads to a better performance. Therefore, it is adopted in this study. The hinge loss provides constant, non-vanishing gradients whenever scores of real images drop below +1 or scores of fake images rise above -1. This margin prevents the discriminator from outputting overly extreme scores and lets the generator receive more stable updates, leading to faster convergence in high-resolution GAN training. The loss functions of discriminator and generator can be described as:

$$L_{D1} = E_{x \sim p_{real}} [\max(0, 1 - D(x))] + E_{x \sim G_{fake}} [\max(0, 1 + D(x))] \quad (1)$$

$$L_G = -E_{x \sim G_{fake}} [D(x)] \quad (2)$$

In addition, the gradient penalty term⁴⁵ is also considered in the loss function of discriminator. In order to approximate the 1-Lipschitz condition, gradient penalty constrains the discriminator's gradient norm at points interpolated between real and generated samples.⁴⁷ This stabilizes adversarial training, suppresses extreme discriminator scores and results in stronger performance in comparison with the method of weight clipping in WGAN.⁴⁸ Therefore, the loss function of discriminator in this paper can be expressed as:

$$L_D = L_{D1} + \lambda \cdot E_{x^* \sim p_{x^*}} [(\|\nabla_{x^*} D(x^*)\|_2 - 1)^2] \quad (3)$$

where x^* can be obtained by interpolating between the fake and real images and λ denotes the weight of gradient penalty term, which is typically set to 10 according to the recommendation of Gulrajani et al.⁴⁷

2.2. Stage 2: generating desired large-scale fracture network

Upon completion of the first stage, a well-trained GAN model is obtained, which has effectively learned and captured the characteristics of the input small-scale fracture networks. And the GPU memory consumption in the first stage is determined mainly by the patch size. Given that the size of each patch in this study is limited to 128×128 pixels, the training process remains memory-efficient and does not require excessive GPU resources.

Then, in the second stage, arbitrary-scale fracture networks are produced by applying the trained model within a patch-by-patch generation paradigm. The generation process of desired large-scale fracture network is illustrated in Fig. 5. However, it is worth noting that when the well-trained GAN model is used to generate two small-scale fracture network images at the same time (i.e., P1 and P2) and forcibly stitch them together, a noticeable inconsistency arises at the junction between the rightmost patches of P1 (i.e., x3) and the leftmost patches of P2 (i.e., y1). This occurs because the two images are generated independently and therefore cannot exploit information from each other during the generation process.

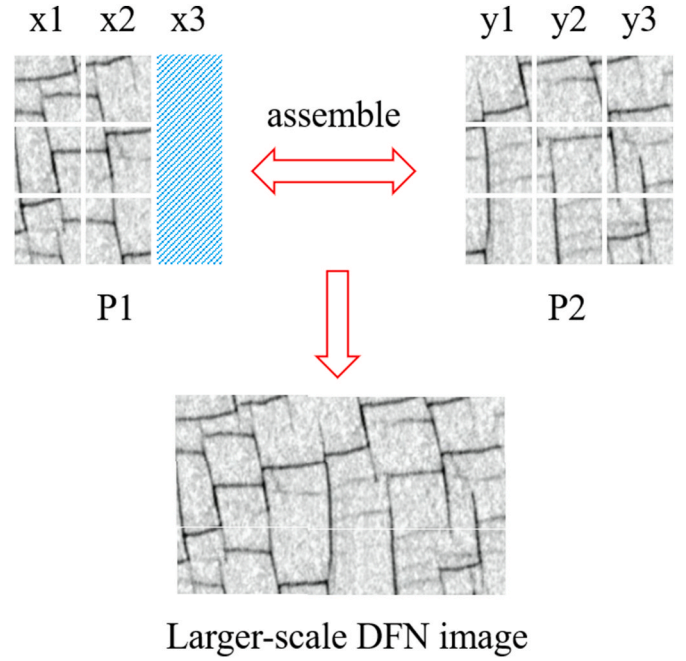


Fig. 5. The generation process of a larger fracture network image.

To address this issue, a new generation process is applied in this stage. The whole process of stage 2 can be summarized as follows. In step 1, the trained GAN model is applied to generate a single small-scale fracture network image (i.e., P1). For the rightmost patches of P1 (i.e., x3), there are no neighboring patches on the right to provide true contextual information. As a result, the right boundaries of $\times 3$ are padded using replicate padding, not local padding. In addition, the pixel values along the right boundary of the intermediate patches (i.e., x2) in P1 are preserved. In step 2, P2 is generated to extend the image horizontally using the trained model. It is worth noting that the leftmost patches of P2 (i.e., y1) now have neighboring patches on the left, i.e., x2. This means that local padding is applied along the left boundary of y1 using the previously saved pixel values from P1. Because the convolutional layers in P2 now are able to “see” the true contextual information from P1, the generated content along the left boundary of P2 becomes consistent with x2. Finally, the rightmost patches of P1 (i.e., x3) are removed, and the remaining patches of P1 are then concatenated with P2. The same workflow is followed when stitching images along the vertical direction. Continuous application of this process along both x and y dimensions enables generation of infinite-size fracture network images.

According to the technical workflow shown in Fig. 5, the assembly of different small-scale images necessitates preserving a few pixel values for local padding, which consumes only a fraction of GPU memory. This design fundamentally eliminates the memory limitations when generating large-scale fracture network images of arbitrary size.

3. Training data and parameter selection

3.1. Training data

In this study, the training data is a small-scale fracture network image with a resolution of 500×500 pixels, sourced from the work of Karimpouli et al.⁴⁹ To facilitate the subsequent quantitative evaluation of both the original and generated fracture network images, the physical size of the training image is assumed to be $5 \text{ m} \times 5 \text{ m}$. The training image is shown in Fig. 6.

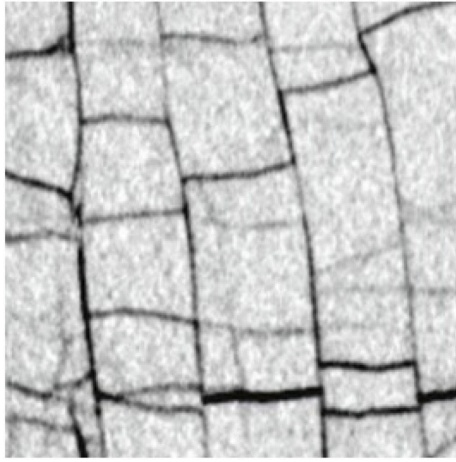


Fig. 6. The small-scale natural fracture network for training.⁴⁹

3.2. Parameter selection

During the training process, at each iteration, a batch of images with the size of 384×384 pixels are randomly cropped from the original image for model optimization. In this work, 3×3 small-scale patches are generated simultaneously, and the size of each patch is set to 128×128 . Subsequently, these patches are merged into a larger image with the size of 384×384 . It is worth noting that the images acquired from the field are directly used as model input, without a series of preprocessing operations, such as binarization, thresholding and skeletonization. Therefore, the cumbersome preprocessing procedures can be avoided and more details can be preserved in our method. The relevant parameters with respect to GAN model and training process are listed in Table 1. In this table, SpectralNorm (Spectral Normalization) is a weight normalization technique applied to neural networks with the objective of enhancing training stability and mitigating issues such as gradient explosion and vanishing gradients.

4. Results: analysis of loss curves and model scalability

4.1. Loss curves

The loss curves of our model during training are shown in Fig. 7. A stable and convergent behavior is observed, indicating that the generator and discriminator have reached a balanced adversarial state. The generator loss initially increases and then stabilizes with moderate oscillations after approximately 100 epochs, indicating adversarial balance and gradual improvement in sample quality. The discriminator losses for both real and fake samples rapidly drop below 0.03. This illustrates that the discriminator has exhibited excellent performance in

Table 1

Relevant parameters used in our model and training process.

Class	Parameters	Value
Training parameters	Number of epochs	780
	Learning rate of generator	0.0002
	Learning rate of discriminator	0.0002
	Batch size	60
	Total number of samples	6000
	Weight of gradient penalty term	10
Model parameters	Dimension of the input noise (number of channels)	128
	Resolution of an input noise patch	4
	SpectralNorm in generator	True
	SpectralNorm in discriminator	True
	$N \times N$ (Number of patches)	3×3
	$h \times w$ (the size of each patch)	128×128

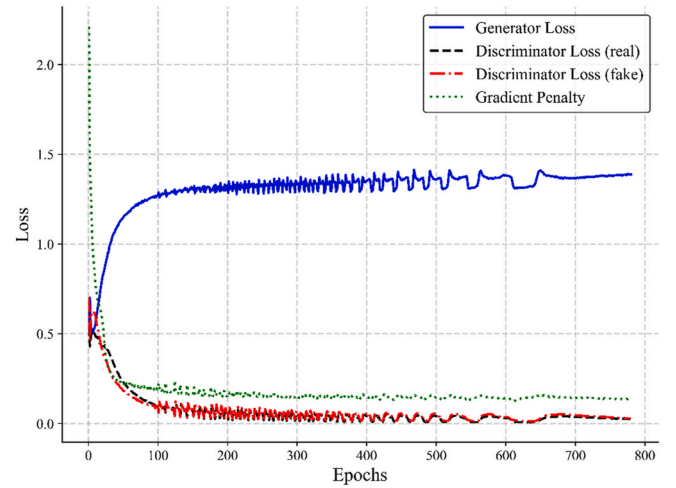


Fig. 7. The loss curves of our model during training.

distinguishing fake samples from real ones. The gradient penalty term remains stable at a low level after the initial decay, demonstrating the effective enforcement of the Lipschitz constraint, which ensures training stability. Therefore, according to the loss curves and visual assessment of the image, the model parameters at the 750th epoch are adopted as the final result for subsequent evaluation and deployment.

4.2. Model scalability

It must be emphasized that, in many models,^{38,41} GPU memory capacity is a key factor limiting the size of generated images. However, in current research, owing to the patch-by-patch generation paradigm and the careful design of the model structure, GPU memory shortage can be completely avoided when generating large-scale images. As shown in Fig. 8, the peak GPU memory consumption during generation process remains nearly constant when the size of the target image increases. And the amount of GPU memory consumption depends mainly on the size of the patches, not on the size of the desired large-scale images. This superior performance makes it easy to conduct fracture network modeling at engineering scales.

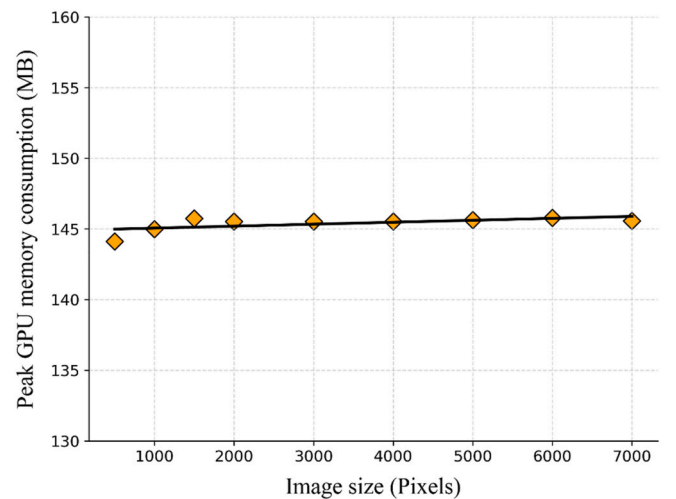


Fig. 8. Peak GPU memory consumption when generating images of different sizes.

5. Results: analysis of model accuracy

5.1. Visual assessment of the AI-generated fracture networks

The Upscaling-GAN algorithm is used to generate fracture network images of various sizes, including 10m * 10m (1000 * 1000 pixels), 20m * 20m (2000 * 2000 pixels) and 30m * 30m (3000 * 3000 pixels). The original small-scale fracture network image and AI-generated upscaled images are shown in Fig. 9.

Several key characteristics can be observed from a purely visual evaluation:

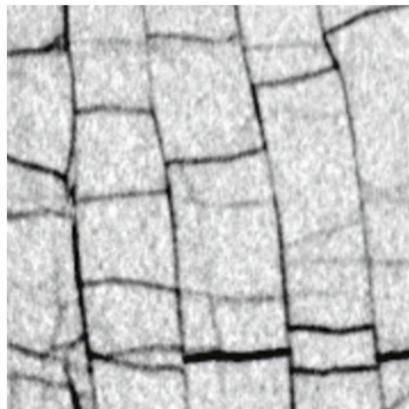
- (1) The AI-generated fracture networks accurately reproduce the topological structure and morphological characteristics of the original fracture network, such as the curved and tortuous trajectories of fractures, the geometric irregularity and the complex interconnected patterns. The images generated by the GAN model exhibit a high level of realism.
- (2) The Upscaling-GAN model successfully captures two dominant fracture orientations present in the original fracture network, i.e., N-S and E-W trends.
- (3) The Upscaling-GAN model is capable of generating fracture networks incorporating visually realistic aperture variations, rather than producing fractures with constant aperture.
- (4) Neither repetitive or identical motifs nor visible seams are observed in the large-sized images. This demonstrates the effectiveness of local padding. In addition, the model accurately learns

the intrinsic features and patterns of the fracture networks rather than merely memorizing the input image.

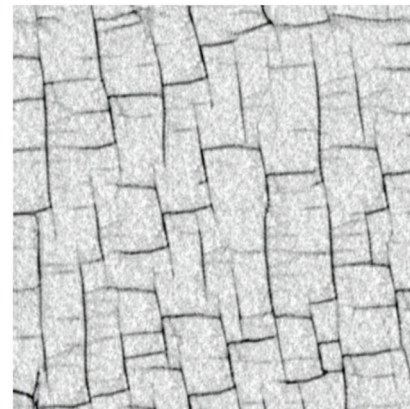
5.2. Quantitative analysis of the AI-generated fracture networks

Before quantitative analysis, it is necessary to conduct a series of preprocessing procedures on these AI-generated fracture network images. We also emphasize that the image preprocessing procedures introduced here, such as binarization, skeletonization, etc., are applied solely to facilitate the subsequent quantitative analysis of the generated fracture networks and to validate the model's effectiveness. They are not required during the entire training process of the model.

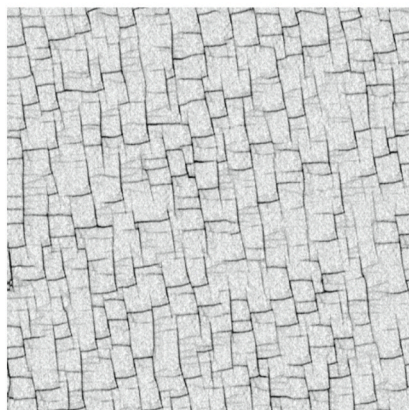
In this paper, the automated fracture trace detection technique proposed by Prabhakaran et al.⁵⁰ is utilized to extract vectorized fracture topologies and geometric attributes from images of fractured rocks. Based on the complex shearlet transform, this automated algorithm is able to capture multi-scale features within the images. There are five key steps in this method: (1) Ridge Detection: extract a highly probable, binarized ridge network by applying complex shearlet transform to the input image under multiple parameter combinations; (2) Segmentation: Otsu thresholding is applied to remove small and isolated pixel clusters; (3) Skeletonization: obtain fracture skeletons with single-pixel width; (4) Polyline Fitting: capture the true orientation and curvature of the fractures through polyline fitting; (5) Polyline Simplification: reduce the number of polyline points to compress the data volume. After these preprocessing steps, a file containing the extracted fracture information can be obtained, which will be used for subsequent quantitative analysis. For a more detailed description of this automated fracture trace



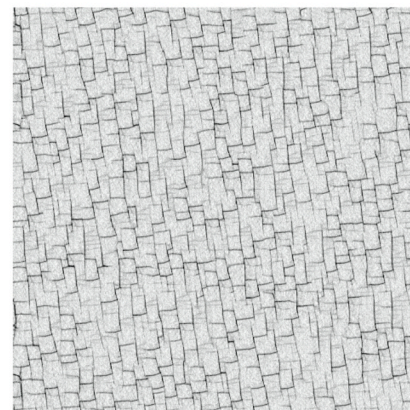
(a) Original image (500 * 500 pixels)



(b) 10m * 10m (1000 * 1000 pixels)



(c) 20m * 20m (2000 * 2000 pixels)



(d) 30m * 30m (3000 * 3000 pixels)

Fig. 9. (a) The original fracture network image and (b–d) images of different sizes generated by our Upscaling-GAN model.

detection technique, interested readers are referred to the works of Prabhakaran et al.^{50,51} and Weismüller et al.⁵²

According to the obtained file, the quantitative analysis and evaluation of the fracture network can be easily performed using FracPaQ,^{53,54} an open-source MATLAB toolbox for the quantification of fracture patterns. This toolbox can be downloaded from the following link: <https://github.com/DaveHealy-github/FracPaQ>.

5.2.1. Fracture orientation

The orientation distributions of fracture segments extracted from the AI-generated images and the real image are shown in Fig. 10. The fracture segment orientations are computed clockwise from the positive Y-axis. Note that a fracture trace is composed of many fracture segments. And the orientation statistics of the fracture networks are derived based on fracture segments. It is apparent from rose diagrams that the fracture orientations in the AI-generated images closely resemble those in the real image. Specially, both the input image and the AI-synthesized fracture networks exhibit two dominant fracture sets oriented near 0°/180° and 90°. This indicates that the present Upscaling-GAN model successfully captures the directional features of the natural fracture system. Although the orientation statistics of the generated and real fracture networks are broadly consistent, subtle discrepancies can still be observed. For example, in AI-generated fracture networks, the fractures that extend in the N-S trend cluster mainly within the 150°–180° range. However, the corresponding fractures in real fracture network are

distributed almost symmetrically between 150°–180° and 0°–30°. This discrepancy may originate from the preprocessing procedures. For example, certain fracture traces might not have been captured during the ridge detection stage.

5.2.2. Fracture intensity

Fracture intensity refers to total fracture length per unit area, which serves as a critical indicator for assessing the degree of fracture development. It can be used to evaluate and predict the hydraulic conductivity of the rock mass, geomechanical behavior and reservoir effectiveness. In this paper, the sampling method of circular scanline, introduced by Mauldon et al.,⁵⁵ is utilized to calculate the fracture intensity. The unbiased estimator for fracture intensity can be expressed as:

$$\hat{l} = \frac{n}{4r} \quad (4)$$

where $\hat{l}$ is the fracture intensity, n denotes the number of intersections between the fracture trace and a circular scanline, and r is the radius of circular scanline. The schematic diagram of this method is shown in Fig. 11.

The fracture intensity calculation results of the original fracture network and AI-generated fracture networks are shown in Fig. 12. Based on the size of these areas, the number of scan circles used in Fig. 12(a)–

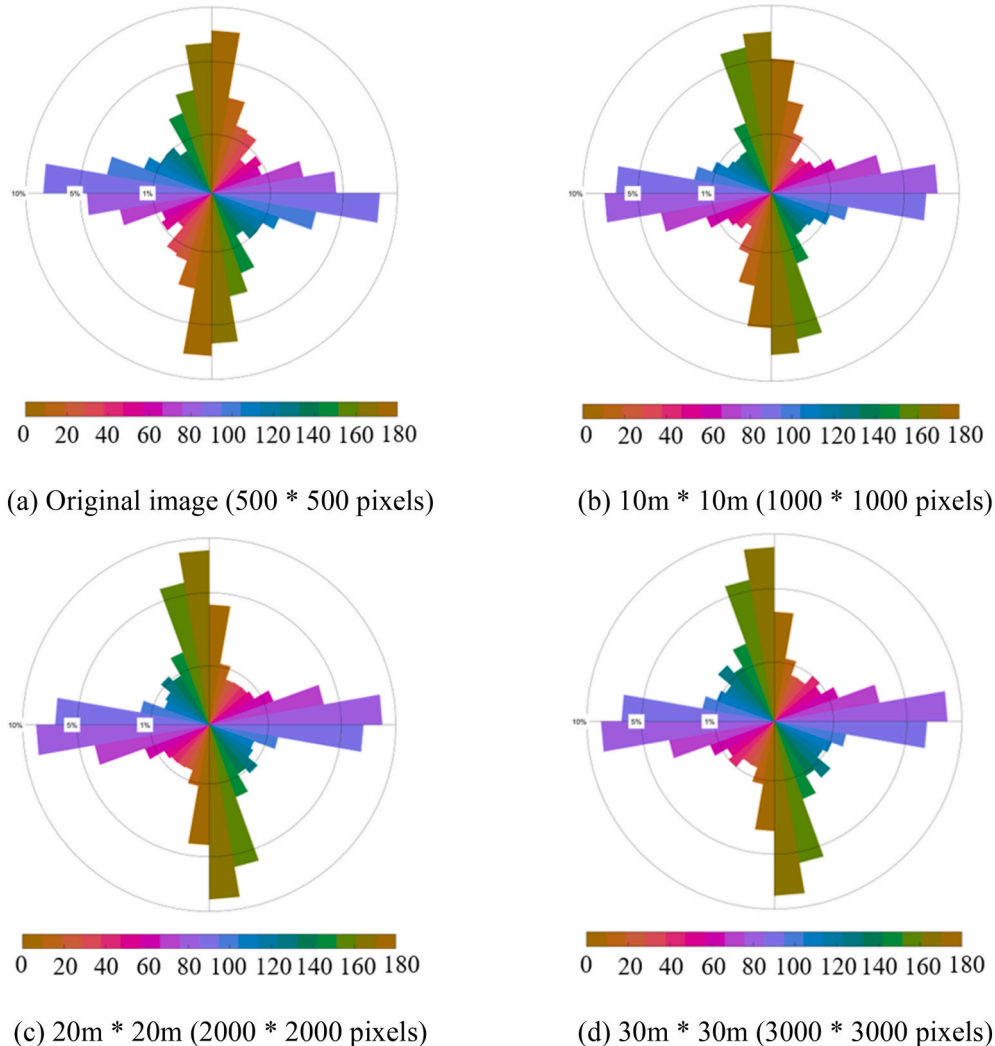


Fig. 10. Rose diagrams of fracture orientations (a) Real fracture network and (b–d) AI-generated fracture networks.

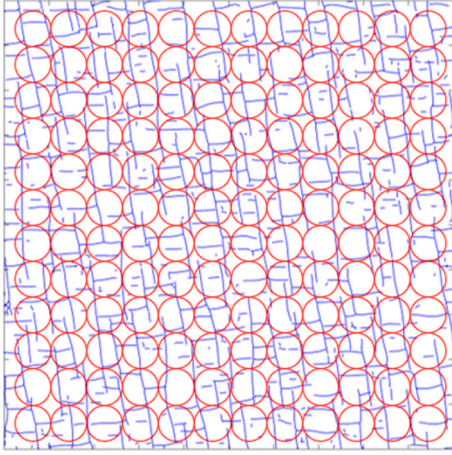


Fig. 11. The schematic diagram of circular scanlines used in the estimation of fracture intensity.

12(d) are set to 9, 36, 144, 324, respectively. A similar distribution can be observed. Specifically, the fracture intensities of whole map are 1.9207 m^{-1} , 1.476 m^{-1} , 1.748 m^{-1} and 1.7549 m^{-1} for real fracture networks and three generated fracture networks, respectively. These results demonstrate that the trained GAN model not only accurately captures the overall spatial density of the real fracture network, but also effectively learns the spatial distribution patterns of fractures across different regions.

5.2.3. Fracture length

In order to analyze the fracture length, the Maximum Likelihood Estimators (MLE) analysis is conducted to determine optimal fracture length distribution function. The probability density functions (PDF) of power law, exponential and log-normal distribution are tested. Finally, the power law distribution is found to be the most suitable for the description of fracture length distribution, which has been verified and widely applied in many practical applications and studies.^{10,56} The specific fitting results are shown in Fig. 13. And the probability density function of power law can be expressed as⁵⁴:

$$p(x|\alpha) = \frac{\alpha - 1}{x_{\min}} \left(\frac{x}{x_{\min}} \right)^{-\alpha} \quad (5)$$

The key parameters in power-law distribution, i.e., α and $x_{\min}$, are listed in Fig. 13. On one hand, the probabilities of using power-law form to characterize the fracture length distribution of the four fracture networks are 99.56 %, 99.28 %, 97.32 %, 97.08 %, respectively, which indicates a high degree of statistical agreement with the power-law behavior. On the other hand, these two key fitting parameters for fracture length in real and AI-generated fracture networks exhibit a certain degree of similarity, which demonstrates that the Upscaling-GAN model have a powerful ability to characterize fracture length distribution. In addition, the maximum fracture length in synthetic fracture networks (approximately 5 m) is slightly larger than that in real fracture network (3.24m). Since several persistent vertical fractures, which have a length of $\sim 5\text{m}$, are easily seen in the real fracture networks (Fig. 9(a)), this discrepancy mainly stems from incomplete extraction of some long fractures during the preprocessing procedures, which leads to the

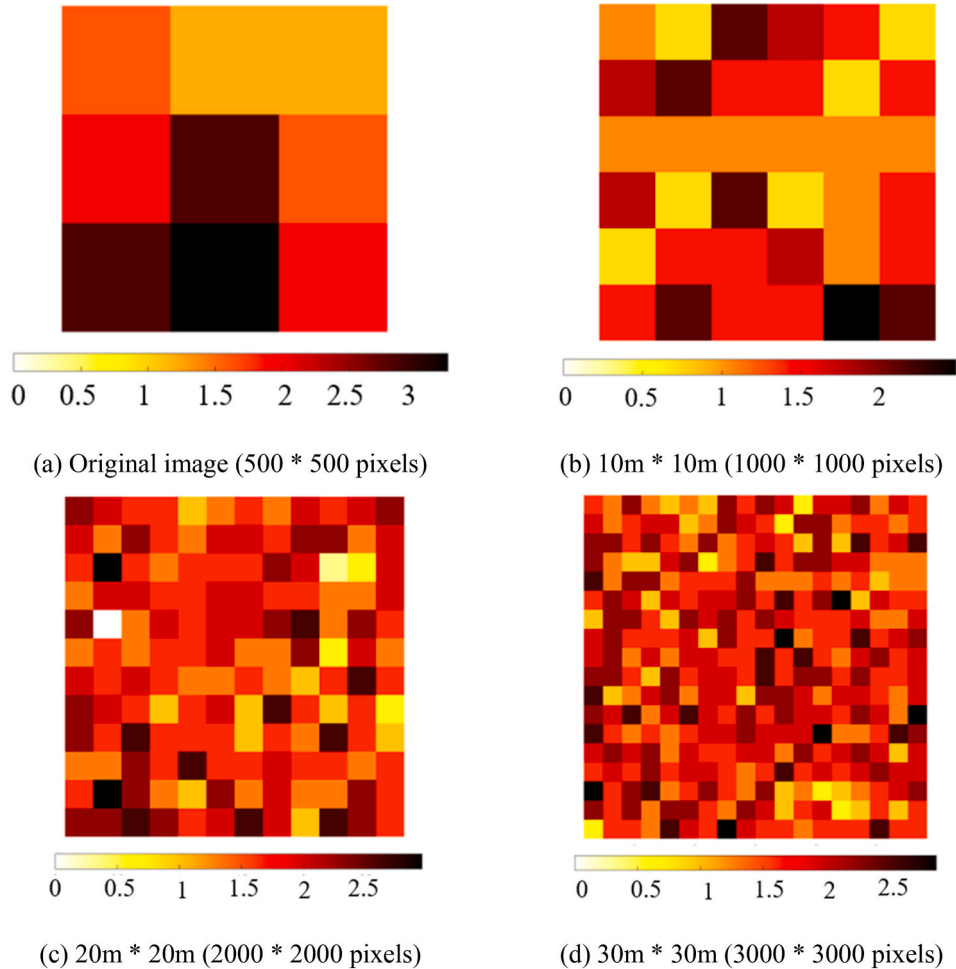


Fig. 12. The fracture intensity calculation results of the original fracture network and AI-generated fracture networks.

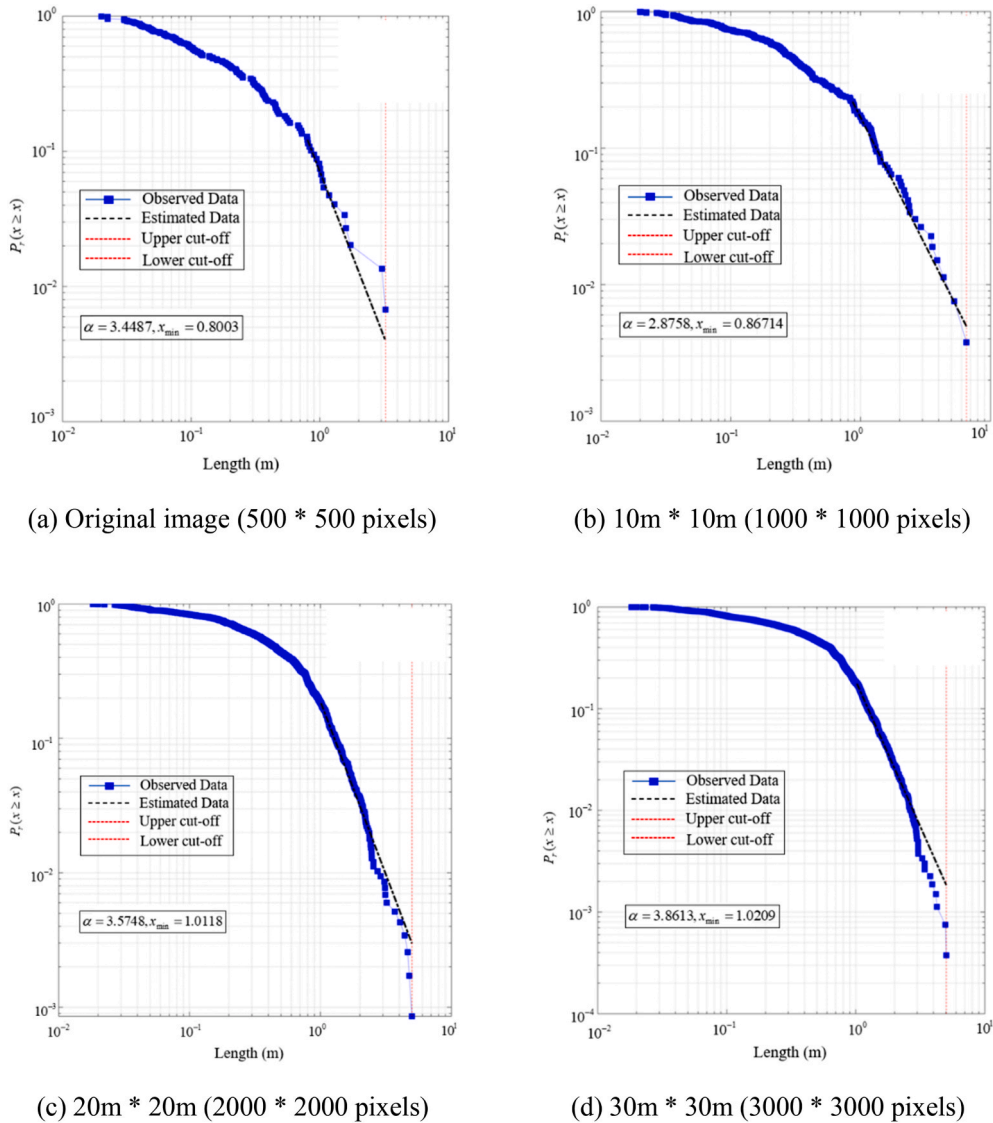


Fig. 13. Power-law scaling estimate for fracture length.

interruption of fractures. This again shows that our dual-scale discriminator architecture results in a wider range of receptive fields, enabling the model to capture long-range dependencies in fracture networks.

5.2.4. Fracture aperture

Fracture aperture is a critical parameter governing fluid flow and reservoir productivity. However, in the realms of fracture network modeling, analysis of fluid flow and numerical simulation of fracture propagation, the initial fracture aperture is usually assigned a constant value.^{14,57,58} Then, the parallel plate model is employed to calculate fracture permeability.⁵⁸ This neglect of the spatial variability of fracture aperture will inevitably lead to the distortion in both seepage predictions and analysis of the rock mass's mechanical behavior. Therefore, it is imperative to accurately characterize the spatial distribution of fracture apertures.

Without the need for preprocessing procedures like skeletonization, the Upscaling-GAN model proposed in this paper learns directly from the raw fracture network images. Hence our method naturally possesses the ability to capture spatial variations in fracture aperture. Instead of producing fractures with a single-pixel width, the results shown in Fig. 9 clearly demonstrate that AI-generated fracture networks are capable of reproducing the spatial distribution characteristics of fracture apertures

observed in the real image. Next, based on techniques such as skeletonization and distance transform, the fracture skeleton is extracted from the binary fracture network image, and the fracture aperture at each skeleton point is computed. Finally, the effectiveness of the AI-generated fracture network images can be further validated by using several quantitative indicators. The calculation results are shown in Table 2.

It can be found that the parameter values with respect to fracture aperture in AI-generated images are consistent with those in original fracture network, especially in terms of the coefficient of variation (CV). The parameter of CV, defined as the ratio of the standard deviation to the

Table 2
Calculation results of fracture apertures.

Image	Mean value (pixel)	SD ^a (pixel)	CV ^a	Maximum value (pixel)
Original image	2.65	1.12	0.42	10
1000 * 1000 pixels	2.50	0.80	0.32	7.2
2000 * 2000 pixels	3.21	1.02	0.32	9.0
3000 * 3000 pixels	3.90	1.46	0.37	10.77

a SD and CV denote standard variation and coefficient of variation, respectively.

mean value, is a dimensionless metric that quantifies the relative data variability or dispersion with respect to its mean value. The consistency of CV shown in Table 2 indicates that the pattern of aperture fluctuations in generated fracture networks is similar to the real one.

6. Discussions

In this part, we present a comprehensive comparative analysis of the performance of our proposed Upscaling-GAN model, vanilla model and

traditional fractal-geometry-based modeling method in generating fracture network images.

6.1. Comparison between the vanilla model and our Upscaling-GAN model

As shown in Fig. 14(a), a fracture network image with a size of 20m * 20m is generated using the vanilla model. And several quantitative indicators are provided to conduct a thorough comparative analysis

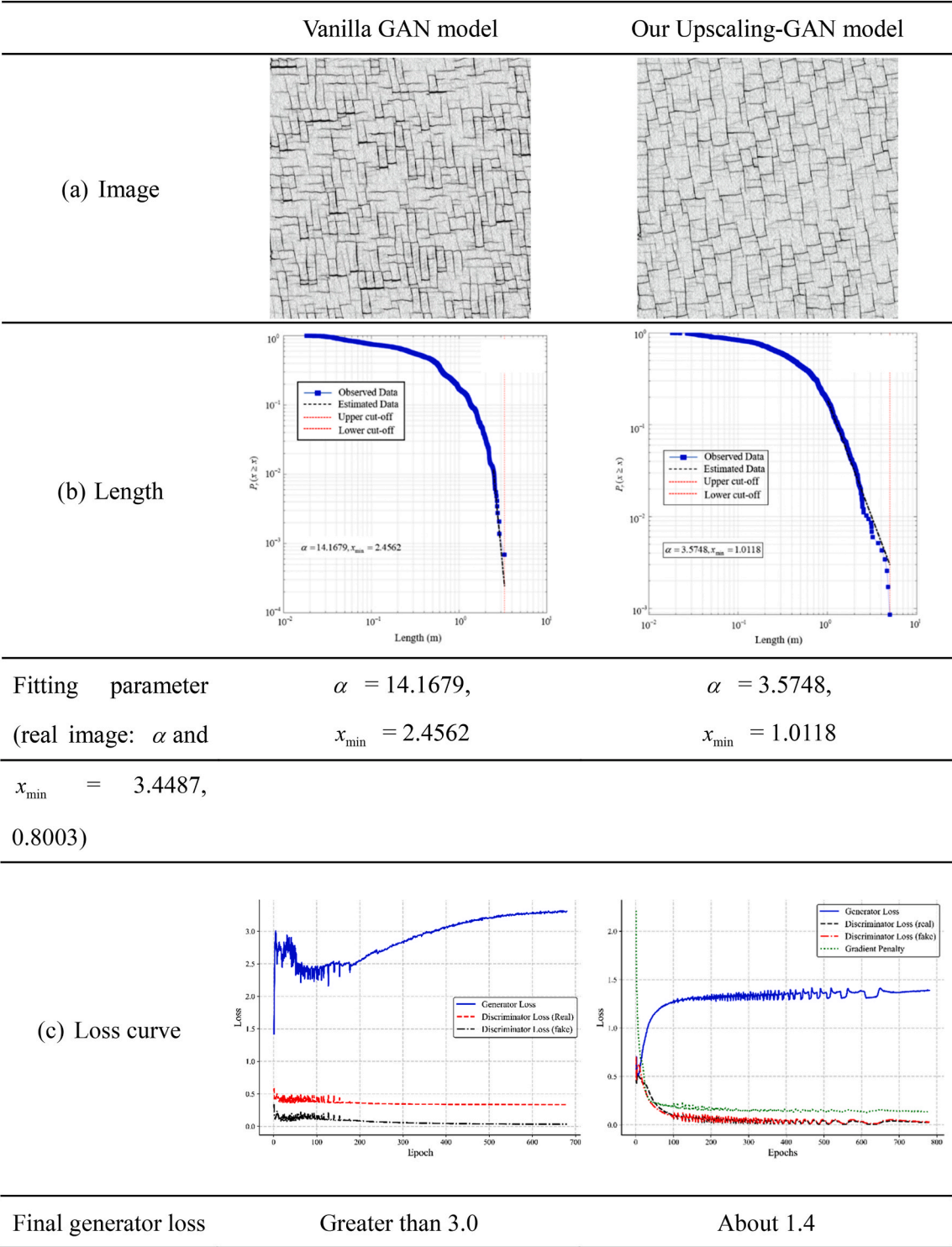


Fig. 14. A comprehensive comparative analysis of the performance between our proposed Upscaling-GAN model and vanilla model.

between the vanilla model and our Upscaling-GAN model. Although the vanilla model can produce visually realistic fracture network image that may satisfy requirements in fields such as computer vision, its outputs still exhibit some unreasonable characteristics when examined from the standpoint of geoscience or geotechnical engineering. For example, it is clearly observed that the fracture network generated by the vanilla model exhibits overly small fracture spacing and higher fracture density in some local regions. Moreover, a closer inspection of the power-law model parameters in Fig. 14(b) indicates that the vanilla model does not accurately characterize the fracture length distribution. Specifically, for the Upscaling-GAN model, the two parameter values are 3.5748 and 1.0118, respectively. And the parameters for the original fracture network are 3.4487 and 0.8003, respectively. In contrast, in the vanilla model, these two parameter values are as high as 14.1679 and 2.4562, respectively. In addition, the loss curves of vanilla model during training are shown in Fig. 14(c). It can be easily found that when the training reaches a relatively stable stage, the generator loss exceeds 3, which is significantly higher than that of Upscaling-GAN model. This further confirms that our modifications to the vanilla model play an important role in enhancing the model's performance and improving the training process.

6.2. Comparison between our Upscaling-GAN model and fractal-geometry-based modeling method

Over the past few decades, fractal geometry has been widely employed in fields such as DFN modeling.^{26,59,60} In order to make a comparison between the fracture networks generated by Upscaling-GAN model and traditional fractal-geometry-based modeling method, the fractal-geometry-based modeling procedure for subsurface fracture networks, which is proposed by Watanabe and Takahashi,²⁵ is utilized in this paper to generate DFNs. Specifically, the mathematical model that describes the relationship between fracture length r and the number of fractures N whose lengths are equal to or larger than r can be expressed as:

$$N = Cr^{-D} \quad (6)$$

where C represents the fracture density parameter, D denotes the fractal dimension.

In this section, the fracture network with a size of 20m * 20m is taken as an example for comparative analysis. Firstly, the fracture network generated by Upscaling-GAN model is quantitatively characterized. The analysis results are summarized in Table 3. And the proportions of fractures within different orientation intervals are presented in Table 4. Then, according to these statistics, a fracture network is generated using the traditional fractal-geometry-based modeling method, which is shown in Fig. 15. It is evident that significant limitations persist in the DFN generated by traditional modeling method, even if the statistical parameters, i.e., fractal dimension, number of fractures, maximum fracture length, minimum fracture length and fracture orientation, derived from the Upscaling-GAN model-generated fracture network are utilized to constrain the entire modeling process. The DFN shown in Fig. 15(a) does not explicitly represent fracture aperture as an intrinsic, spatially varying geometric feature. Instead, in many stochastic or fractal-geometry-based DFN modeling approaches, the aperture information is typically introduced as an additional a posteriori attribute

Table 3
Analysis results of the fracture network generated by our Upscaling-GAN model.

Parameter	Value
Fractal dimension	1.5303
Number of fractures ($\geq 0.3m$)	710
Maximum fracture length	5.0m
Minimum fracture length	0.3m

Table 4
Orientation information of the fracture network.

Angle interval	Percentage (%)	Angle interval	Percentage (%)
0 – 10°	8.300929839	90 – 100°	13.6339814
10 – 20°	2.204564666	100 – 110°	2.797125951
20 – 30°	1.334742181	110 – 120°	1.302620456
30 – 40°	1.378698225	120 – 130°	1.46238377
40 – 50°	1.401521555	130 – 140°	2.268808115
50 – 60°	1.551141167	140 – 150°	1.846999155
60 – 70°	2.691462384	150 – 160°	3.401521555
70 – 80°	7.771766695	160 – 170°	11.95604396
80 – 90°	17.20033812	170 – 180°	17.4953508

through predefined statistical distributions or empirical rules. In addition, traditional fractal-geometry-based method can't reproduce the actual geometric characteristics and topological structure of original fracture network, including the spatial distribution, relative positional relationships among fractures and fracture tortuosity. However, the framework proposed in this paper not only accurately learns these complicated fracture patterns but also faithfully reproduces these features at any scale.

7. Conclusions

Based on the theory of GAN, we have introduced a novel and high-fidelity scale-up method for fracture network modeling, termed Upscaling-GAN, which provides a solid foundation for more accurate analysis of fractured rock masses. The present Upscaling-GAN model exhibits the following key advantages in fracture network modeling:

- (1) By qualitative evaluation and quantitative analysis, fracture networks of different scales generated by our Upscaling-GAN model can accurately reproduce the geometrical characteristics and spatial topological structures inherent in the original small-scale fracture networks, including fracture orientation, fracture intensity and fracture length. And the superior capability of our approach to generate realistic fracture networks is further demonstrated and validated through a detailed comparison with the vanilla model and traditional fractal-geometry-based modeling method.
- (2) Benefiting greatly from the direct learning of raw fracture network images without any preprocessing, our model has the native ability to characterize fracture apertures. This ability, usually overlooked or unavailable in many models, is crucial for accurately simulating fluid flow and fracture propagation in fractured rock masses.
- (3) Unlike many other GAN models that suffer from severe GPU memory limitations when the size of the target image increases, the peak GPU memory consumption in our Upscaling-GAN model remains nearly unchanged.

CRediT authorship contribution statement

Mengmeng Nie: Writing – original draft, Software, Methodology, Conceptualization. **Xuhai Tang:** Writing – review & editing, Supervision, Funding acquisition, Conceptualization. **Fei Gao:** Visualization, Formal analysis, Data curation. **Quansheng Liu:** Writing – review & editing, Resources, Data curation, Conceptualization. **Jiangmei Qiao:** Validation, Project administration, Investigation.

Consent to publish

All co-authors consented to publish this work.

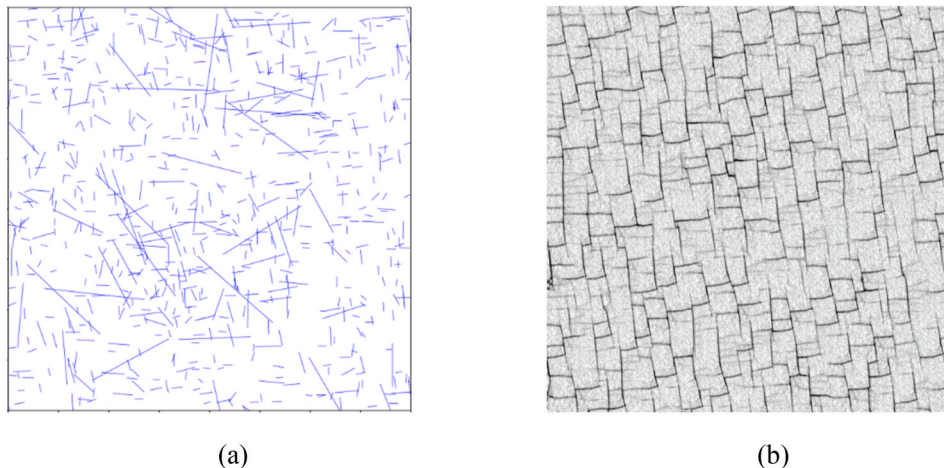


Fig. 15. (a) Fracture network generated by fractal-geometry-based modeling method; (b) fracture network generated by the Upscaling-GAN model.

Ethics approval

The authors declared that:

- this work is original and has not been submitted to more than one journal for simultaneous consideration;
- the manuscript has not been published previously (partly or in full);
- a single study is not split up into several parts to increase the number of submissions;
- no data have been fabricated or manipulated (including images) to support our conclusions;
- proper acknowledgements to other works have been given.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant No. 12502471, 42577189), Shenzhen Science and Technology Program (Grant No. JCYJ20220530140602005, JCYJ20230807090102006, JCYJ20240813111201003) and Guangdong Basic and Applied Basic Research Foundation (Grant No. 2024A1515011772). The model training in this paper has been done on the supercomputing system in the Supercomputing Center of Wuhan University.

Data availability

Data will be made available on request.

References

1. Li XX, Liu JS, Gong WP, Xu Y, Bowa VM. A discrete fracture network based modeling scheme for analyzing the stability of highly fractured rock slope. *Comput Geotech.* 2022;141, 104558. <https://doi.org/10.1016/j.compgeo.2021.104558>.
2. Luo S, Zhao ZH, Peng H, Pu H. The role of fracture surface roughness in macroscopic fluid flow and heat transfer in fractured rocks. *Int J Rock Mech Min Sci.* 2016;87: 29–38. <https://doi.org/10.1016/j.ijrmms.2016.05.006>.
3. Wang PT, Ma C, Zhang B, Gou Q, Tan WH, Cai MF. Development of an improved three-dimensional rough discrete fracture network model: method and application. *Int J Min Sci Technol.* 2023;33(12):1469–1485. <https://doi.org/10.1016/j.ijmst.2023.10.004>.
4. Chen JY, Zhou ML, Huang HW, Zhang DM, Peng ZC. Automated extraction and evaluation of fracture trace maps from rock tunnel face images via deep learning. *Int J Rock Mech Min Sci.* 2021;142, 104745. <https://doi.org/10.1016/j.ijrmms.2021.104745>.
5. Palamakumbura R, Krabbendam M, Whitbread K, Arnhardt C. Data acquisition by digitizing 2-D fracture networks and topographic lineaments in geographic information systems: further development and applications. *Solid Earth.* 2020;11(5): 1731–1746. <https://doi.org/10.5194/se-11-1731-2020>.
6. Fang JL, Zhou FD, Tang ZH. Discrete fracture network modelling in a naturally fractured carbonate reservoir in the jingbei oilfield, China. *Energies.* 2017;10. <https://doi.org/10.3390/en10020183>.
7. Yao C, Shao YL, Yang JH, et al. Effects of fracture density, roughness, and percolation of fracture network on heat-flow coupling in hot rock masses with embedded three-dimensional fracture network. *Geothermics.* 2020;87, 101846. <https://doi.org/10.1016/j.geothermics.2020.101846>.
8. Xiong F, Jiang QH, Xu CS, Zhang XB, Zhang QH. Influences of connectivity and conductivity on nonlinear flow behaviours through three-dimension discrete fracture networks. *Comput Geotech.* 2019;107:128–141. <https://doi.org/10.1016/j.compgeo.2018.11.014>.
9. Cacciari PP, Futai MM. Modeling a shallow rock tunnel using terrestrial laser scanning and discrete fracture networks. *Rock Mech Rock Eng.* 2017;50(5): 1217–1242. <https://doi.org/10.1007/s00603-017-1166-6>.
10. Lei QH, Latham JP, Tsang CF. The use of discrete fracture networks for modelling coupled geomechanical and hydrological behaviour of fractured rocks. *Comput Geotech.* 2017;85:151–176. <https://doi.org/10.1016/j.compgeo.2016.12.024>.
11. Hyman JD, Karra S, Makedonska N, Gable CW, Painter SL, Viswanathan HS. DFNWORKS: a discrete fracture network framework for modeling subsurface flow and transport. *Comput Geosci.* 2015;84:10–19. <https://doi.org/10.1016/j.cageo.2015.08.001>.
12. Huang N, Liu RC, Jiang YJ, Cheng YF. Development and application of three-dimensional discrete fracture network modeling approach for fluid flow in fractured rock masses. *J Nat Gas Sci Eng.* 2021;91. <https://doi.org/10.1016/j.jngse.2021.103957>.
13. Lei QH. *Characterisation and Modelling of Natural Fracture Networks: Geometry, Geomechanics and Fluid Flow*. Dissertation: Imperial College London; 2016. <https://doi.org/10.25560/42358>.
14. Zhou CT, Lyu B, Wang Y. FracGen: natural fracture networks reconstruction and upscaling using generative adversarial networks. *Int J Rock Mech Min Sci.* 2025;191, 106116. <https://doi.org/10.1016/j.ijrmms.2025.106116>.
15. Medici G, Ling FL, Shang JL. Review of discrete fracture network characterization for geothermal energy extraction. *Front Earth Sci.* 2023;11. <https://doi.org/10.3389/feart.2023.1328397>.
16. Maillot J, Davy P, Le Goc R, Darcel C, de Dreuzy JR. Connectivity, permeability, and channeling in randomly distributed and kinematically defined discrete fracture network models. *Water Resour Res.* 2016;52(11):8526–8545. <https://doi.org/10.1002/2016WR018973>.
17. Baecher GB. Statistical analysis of rock mass fracturing. *Math Geol.* 1983;15(2): 329–348. <https://doi.org/10.1007/BF01036074>.
18. Dershowitz WS, Einstein HH. Characterizing rock joint geometry with joint system models. *Rock Mech Rock Eng.* 1988;21(1):21–51. <https://doi.org/10.1007/BF01019674>.
19. Xu CS, Dowd P. A new computer code for discrete fracture network modelling. *Comput Geosci.* 2010;36(3):292–301. <https://doi.org/10.1016/j.cageo.2009.05.012>.
20. Klimczak C, Schultz RA, Parashar R, Reeves DM. Cubic law with aperture-length correlation: implications for network scale fluid flow. *Hydrogeol J.* 2010;18(4): 851–862. <https://doi.org/10.1007/s10040-009-0572-6>.
21. Leung CTO, Zimmerman RW. Estimating the hydraulic conductivity of two-dimensional fracture networks using network geometric properties. *Transport Porous Media.* 2012;93(3):777–797. <https://doi.org/10.1007/s11242-012-9982-3>.
22. Lang PS, Paluszny A, Zimmerman RW. Permeability tensor of three-dimensional fractured porous rock and a comparison to trace map predictions. *J Geophys Res Solid Earth.* 2014;119(8):6288–6307. <https://doi.org/10.1002/2014JB011027>.

23. Bonneau F, Caumon G, Renard P. Impact of a stochastic sequential initiation of fractures on the spatial correlations and connectivity of discrete fracture networks. *J Geophys Res Solid Earth*. 2016;121(8):5641–5658. <https://doi.org/10.1002/2015JB012451>.
24. Hekmatnejad A, Emery X, Elmo D. A geostatistical approach to estimating the parameters of a 3D cox-boolean discrete fracture network from 1D and 2D sampling observations. *Int J Rock Mech Min Sci*. 2019;113:183–190. <https://doi.org/10.1016/j.ijrmm.2018.11.003>.
25. Watanabe K, Takahashi H. Fractal geometry characterization of geothermal reservoir fracture networks. *J Geophys Res Solid Earth*. 1995;100(B1):521–528. <https://doi.org/10.1029/94JB02167>.
26. Davy P, Le Goc R, Darcel C, Bour O, de Dreuzy JR, Munier R. A likely universal model of fracture scaling and its consequence for crustal hydromechanics. *J Geophys Res Solid Earth*. 2010;115(B10). <https://doi.org/10.1029/2009JB007043>.
27. Li LC, Wu WB, El Naggar MH, Mei GX, Liang RZ. Characterization of a jointed rock mass based on fractal geometry theory. *Bull Eng Geol Environ*. 2019;78(8): 6101–6110. <https://doi.org/10.1007/s10064-019-01526-x>.
28. Davy P, Le Goc R, Darcel C. A model of fracture nucleation, growth and arrest, and consequences for fracture density and scaling. *J Geophys Res Solid Earth*. 2013;118 (4):1393–1407. <https://doi.org/10.1002/jgrb.50120>.
29. La Pointe PR. A method to characterize fracture density and connectivity through fractal geometry. *Int J Rock Mech Min Sci*. 1988;25(6):421–429. [https://doi.org/10.1016/0148-9062\(88\)90982-5](https://doi.org/10.1016/0148-9062(88)90982-5).
30. Lavoine E, Davy P, Darcel C, Munier R. A discrete fracture network model with stress-driven nucleation: impact on clustering, connectivity, and topology. *Front Phys*. 2020;8. <https://doi.org/10.3389/fphy.2020.00009>.
31. Kolapo P, Ogunsola NO, Munemo P, Alewi D, Komolafe K, Giwa-Bioku A. DFN: an emerging tool for stochastic modelling and geomechanical design. *Eng*. 2023;4: 174–205. <https://doi.org/10.3390/eng4010011>.
32. Abdellatif A, Elsheikh AH, Menke HP. Local padding in patch-based GANs for seamless infinite-sized texture synthesis. <https://doi.org/10.48550/arXiv.2309.02340>; 2024.
33. Jetchev N, Bergmann U, Vollgraf R. Texture synthesis with spatial generative adversarial networks. <https://doi.org/10.48550/arXiv.1611.08207>; 2016.
34. Li CL, Jiang YM, Cheslyar M. Embedding image through generated intermediate medium using deep convolutional generative adversarial network. *CMC-Comput Mat Contin*. 2018;56(2):313–324. <https://doi.org/10.3970/cmc.2018.03950>.
35. Phillips A, Lang J, Mould D. Diverse non-homogeneous texture synthesis from a single exemplar. *Comput Graph-UK*. 2024;124. <https://doi.org/10.1016/j.cag.2024.104099>.
36. Fruhstuck A, Alhashim I, Wonka P. TileGAN synthesis of large-scale non-homogeneous textures. *ACM Trans Graph*. 2019;38(4). <https://doi.org/10.1145/3306346.3322993>.
37. Zhou Y, Zhu Z, Bai X, Lischinski D, Cohen-Or D, Huang H. Non-stationary texture synthesis by adversarial expansion. *ACM Trans Graph*. 2018;37(4). <https://doi.org/10.1145/3197517.3201285>.
38. Bergmann U, Jetchev N, Vollgraf R. Learning texture manifolds with the periodic spatial GAN. In: *34th International Conference on Machine Learning*; 2017. Sydney, Australia <https://dl.acm.org/doi/10.5555/3305381.3305430>.
39. Rodriguez-Pardo C, Garces E. SeamlessGAN: self-supervised synthesis of tileable texture maps. *IEEE Trans Vis Comput Graph*. 2023;29(6):2914–2925. <https://doi.org/10.1109/TVCG.2022.3143615>.
40. Chen GD, Luo X, Jiao JJ, Jiang CY. Fracture network characterization with deep generative model based stochastic inversion. *Energy*. 2023;273, 127302. <https://doi.org/10.1016/j.energy.2023.127302>.
41. Shaham TR, Dekel T, Michaeli T. SinGAN: learning a generative model from a single natural image. In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019:4569–4579. <https://doi.org/10.1109/ICCV.2019.00467>. Seoul, South Korea.
42. Teng Z, Wu H, Zhang JZ, Ju X, Qi SW. Generating high-fidelity discrete fracture networks from low-dimensional latent spaces using generative adversarial network. *Int J Rock Mech Min Sci*. 2025;196, 106301. <https://doi.org/10.1016/j.ijrmm.2025.106301>.
43. Dumoulin V, Visin F. A guide to convolution arithmetic for deep learning. <https://doi.org/10.48550/arXiv.1603.07285>; 2018.
44. Isola P, Zhu JY, Zhou TH, Efros AA. Image-to-image translation with conditional adversarial networks. In: *30th IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017. <https://doi.org/10.1109/CVPR.2017.632>. Honolulu, HI.
45. Sun C, Demianov V, Arnold D. Geological realism in fluvial facies modelling with GAN under variable depositional conditions. *Comput Geosci*. 2023;27(2):203–221. <https://doi.org/10.1007/s10596-023-10190-w>.
46. Goodfellow IJ, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets. *28th Conference on Neural Information Processing Systems (NIPS)*. 2014;27:2672–2680. Montreal, Canada <https://dl.acm.org/doi/10.5555/2969033.2969125>.
47. Gulrajani I, Ahmed F, Arjovsky M, Dumoulin V, Courville A. Improved training of wasserstein GANs. In: *31st Annual Conference on Neural Information Processing Systems*. Long Beach, CA: NIPS; 2017. <https://dl.acm.org/doi/10.5555/3295222.3295327>.
48. Arjovsky M, Chintala S, Bottou L. Wasserstein GAN. <https://doi.org/10.48550/arXiv.1701.07875>; 2017.
49. Karimipouli S, Tahmasebi P, Saenger EH. Coal cleat/fracture segmentation using convolutional neural networks. *Nat Resour Res*. 2020;29(3):1675–1685. <https://doi.org/10.1007/s11053-019-09536-y>.
50. Prabhakaran R, Bruna PO, Bertotti G, Smeulders D. An automated fracture trace detection technique using the complex shearlet transform. *Solid Earth*. 2019;10(6): 2137–2166. <https://doi.org/10.5194/se-10-2137-2019>.
51. Prabhakaran R, Urai JL, Bertotti G, Weismüller C, Smeulders DMJ. Large-scale natural fracture network patterns: insights from automated mapping in the lillstock (Bristol Channel) limestone outcrops. *J Struct Geol*. 2021;150. <https://doi.org/10.1016/j.jsg.2021.104405>.
52. Weismüller C, Prabhakaran R, Passchier M, Urai JL, Bertotti G, Reicherter K. Mapping the fracture network in the Lillstock pavement, Bristol Channel, UK: manual versus automatic. *Solid Earth*. 2020;11(5):1773–1802. <https://doi.org/10.5194/se-11-1773-2020>.
53. Healy D, Rizzo RE, Cornwell DG, et al. FracPaQ: a MATLAB™ toolbox for the quantification of fracture patterns. *J Struct Geol*. 2017;95:1–16. <https://doi.org/10.1016/j.jsg.2016.12.003>.
54. Rizzo RE, Healy D, De Siena L. Benefits of maximum likelihood estimators for fracture attribute analysis: implications for permeability and up-scaling. *J Struct Geol*. 2017;95:17–31. <https://doi.org/10.1016/j.jsg.2016.12.005>.
55. Mauldon M, Dunne WM, Rohrbaugh Jr MB. Circular scanlines and circular windows: new tools for characterizing the geometry of fracture traces. *J Struct Geol*. 2001;23 (2):247–258. [https://doi.org/10.1016/S0191-8141\(00\)00094-8](https://doi.org/10.1016/S0191-8141(00)00094-8).
56. Bonnet E, Bour O, Odling NE, et al. Scaling of fracture systems in geological media. *Rev Geophys*. 2001;39(3):347–383. <https://doi.org/10.1029/1999RG000074>.
57. Yao WL, Sharifzadeh M, Yang Z, Xu G, Fang ZG. Assessment of fracture characteristics controlling fluid flow performance in discrete fracture networks (DFN). *J Pet Sci Eng*. 2019;178:1104–1111. <https://doi.org/10.1016/j.petrol.2019.04.011>.
58. Tang XH, Rutqvist J, Hu MS, Rayudu NM. Modeling three-dimensional fluid-driven propagation of multiple fractures using TOUGH-FEMM. *Rock Mech Rock Eng*. 2019; 52(2):611–627. <https://doi.org/10.1007/s00603-018-1715-7>.
59. Bour O, Davy P, Darcel C, Odling N. A statistical scaling model for fracture network geometry, with validation on a multiscale mapping of a joint network (Hornelen Basin, Norway). *J Geophys Res Solid Earth*. 2002;107(B6). <https://doi.org/10.1029/2001JB000176>. ETG 4-1-ETG 4-12.
60. Darcel C, Bour O, Davy P, de Dreuzy JR. Connectivity properties of two-dimensional fracture networks with stochastic fractal correlation. *Water Resour Res*. 2003;39(10). <https://doi.org/10.1029/2002WR001628>.