Computational Geosciences (2021) 25:553–573
https://doi.org/10.1007/s10596-020-10027-w

ORIGINAL PAPER

U-net generative adversarial network for subsurface facies modeling

Chengkai Zhang [1] · Xianzhi Song [1] · Leonardo Azevedo [2]

Received: 6 July 2020 / Accepted: 15 December 2020 / Published online: 4 January 2021

Abstract

Subsurface models are central pieces of information in different earth-related disciplines such as groundwater management and hydrocarbon reservoir characterization. These models are normally obtained using geostatistical simulation methods. Recently, methods based on deep learning algorithms have been applied as subsurface model generators. However, there are still challenges on how to include conditioning data and ensure model variability within a set of realizations. We illustrate the potential of Generative Adversarial Networks (GANs) to create unconditional and conditional facies models. Based on a synthetic facies dataset, we first train a Deep Convolution GAN (DCGAN) to produce unconditional facies models. Then, we show how image-to-image translation based on a U-Net GAN framework, including noise-layers, content loss function and diversity loss function, is used to model conditioning geological facies. Results show that GANs are powerful models to capture complex geological facies patterns and to generate facies realizations indistinguishable from the ones comprising the training dataset. The U-Net GAN framework performs well in providing variable models while honoring conditioning data in several scenarios. The results shown herein are expected to spark a new generation of methods for subsurface geological facies with fragmentary measurements.

Keywords Facies modeling · Image-to-image translation · Generative adversarial network · U-Net · Deep learning

1 Introduction

Modeling the subsurface rock properties plays a vital role in different geosciences field such as reservoir modeling and characterization (e.g., [36]), groundwater management (e.g., [39]) and mining engineering (e.g., [43]). A key step within the geo-modeling workflow is predicting the spatial distribution of the subsurface geological facies as the subsurface rock properties depend on the rock-type (i.e., the facies). This step involves building realistic three-dimensional numerical geological facies models honoring direct measurements available at few and sparse direct observations (e.g., borehole data).

Geostatistical simulation methods have been the preferred methods to build subsurface facies models. The most common modeling approaches comprise sequential indicator simulation [13], truncated Gaussian simulation [30] and geostatistical simulation conditioned to multi-point statistics (MPS) [3, 12, 28, 29, 41]. These methods have been successfully implemented in different applications and geological settings. All these geostatistical simulation techniques reproduce the experimental data at their location. However, they do have limitations regarding the reproduction of the spatial distribution and connectivity of complex and non-stationary sedimentary environments such as turbidite channels.

Recently, data-driven deep learning methods based on neural networks have been proposed to generate facies models (e.g., [6, 9, 16]). Under this framework, one of the most attractive methods due to its ability in reproducing complex spatial patterns is a semi-supervised deep learning model named Generative Adversarial Network (GAN). GAN was originally proposed by Goodfellow et al. [18] within the context of image generation and processing. Fed with a random latent vector z, a GAN model is able to map from z into an image that reproduces the probability distribution of the

✉ Xianzhi Song
songxz@cup.edu.cn

✉ Leonardo Azevedo
leonardo.azevedo@tecnico.ulisboa.pt

Chengkai Zhang
2020310167@student.cup.edu.cn

[1] State Key Laboratory of Petroleum Resources and Prospecting, China University of Petroleum (Beijing), Beijing 102249, China

[2] CERENA, DECivil, Instituto Superior Técnico, Universidade de Lisboa, Av. Rovisco Pais, 1049-001 Lisbon, Portugal

training dataset. Nowadays, several GAN variants have been proposed to improve the performance of the original formulation and be implemented to real-world applications [19]. GAN and its variants such as the Deep Convolutional Generative Adversarial network (DCGAN, [37]), Wasserstein Generative Adversarial Network (WGAN, [2]) and conditional Generative Adversarial Network (cGAN, [32]) have been applied to different fields such as image style transfer [23], image translation [21, 49], image super-resolution [26], natural language processing [22], image classification [50] and data synthesis [4, 14].

As for the application of GAN in geoscience, research efforts towards generating unconditional and conditional facies models are growing rapidly in recent years due to its potentially good performance and high efficiency when compared with conventional geostatistical modeling techniques. Chan and Elsheikh [9] studied the performance of WGAN in generating unconditional geological models while preserving the statistical features of the training dataset. This objective was assessed both visually and quantitatively. Laloy et al. [24] introduced GAN to generate unconditional 2-D and 3-D geological realizations of facies. Sun [42] formulated a state-parameter identification GAN for obtaining deep bidirectional representations of geophysical models. Zhong et al. [48] formulated an unconditional GAN (ucGAN) model to learn the dynamic functional mappings in multiphase models. Mosser et al. [33] investigated the use of GAN in stochastically reconstructing structure of porous media [34] as well as conditionally generating subsurface properties constrained to well data [35]. Dupont et al. [16] developed GAN models to generate multiple 2D geological realizations constrained to some physical measurements, followed by Zhang et al. [47], which extended Dupont's model to 3D. Laloy et al. [25] explored the feasibility of gradient-based deterministic inversion of geophysical data with Generative Adversarial Networks. Azevedo et al. [6] explored the application of GAN in generating unconditional and conditional subsurface property models for discrete and continuous properties. These works show that, when compared against geostatistical simulation methods, GANs are able to better reproduce continuities in channelized sedimentary features while allowing a wide exploration of the model parameter space. Recently, the generation of conditioned facies models has also been approached based on the progressive growing of GANs [40].

When generating conditioning geological models (i.e., locally constrained to some direct observation), the most common approach in the aforementioned works regard it as semantic image inpainting problem and rely on the framework proposed by Yeh et al. [46] (Fig. 1). This framework turns the constrained model generation task into an optimization problem divided in two steps. The first step consists in training an unconditional GAN model able to provide plausible facies realizations when inputting a vector $\boldsymbol{z}$ and trained using a dataset composed of geological facies models. In the second step, a vector $\boldsymbol{z_0}$ is randomly sampled and the GAN outputs an image mapping from $\boldsymbol{z_0}$. Then, the error between the generated image and the conditioning data is computed and used to update the location of the vector $\boldsymbol{z}$ in the latent space. The optimization is performed using optimization algorithms such as gradient decent method or Newton's method (e.g., [6, 35]). The iterative process is repeated until some user-defined criterion of convergence is reached. The optimized location of $\boldsymbol{z_0}$ in the latent space is used to generate geological models able to reproduce the constraining data, which is regarded as the

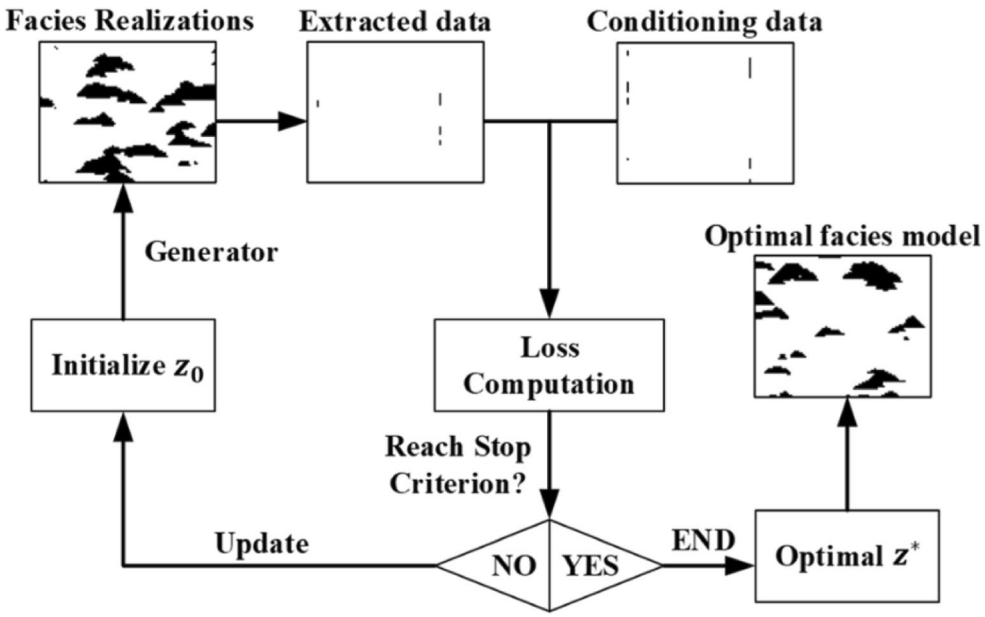

Fig. 1 Schematic representation of the workflow of semantic inpainting problem under the scope of conditional facies model generation

optimal solution. Though this method is able to produce indistinguishable and well-constrained geological models, the variability, or diversity, of the generated facies models is greatly limited because only one single local optimal solution can be obtained. In subsurface earth modeling, the ability to generate variable and diverse models is a critical feature as it is related to the ability to explore the model parameter space, a critical step for decision making.

In this work we develop a generative model able to produce geologically plausible facies models considering two scenarios with and without conditioning data. In the example with direct observations we explored the flexibility of the proposed generative model to handle different configurations of conditioning data. These objectives are attained using a DCGAN model trained for the unconditional scenario, while a U-Net GAN framework, consisting on a U-Net structure with content and diversity loss functions, is proposed and trained for conditioning facies model generation. In this scenario, we propose the conditioning of the models during the training stage of the deep network in contrast to the two-step approach as proposed in the inpainting problem [46].

In both scenarios the generative models are trained using a training dataset where each sample represents a facies model generated with geostatistical simulation conditioned to multi-point statistics. The quality of the generated models by the proposed deep learning methods is assessed based on the visual comparison between real and generated facies models, statistical properties and variability within the generated ensembled. Cases consisting of different data conditioning sets are studied to demonstrate the applicability of the proposed framework. The influence of the U-Net GAN structure and loss functions on the results is discussed. As far as our knowledge goes, this work represents a first approach in establishing a generative model to produce realistic facies models, honoring constraints during training and not as a two-step approach as in the inpainting problem [46].

2 Methodology

2.1 Generative adversarial networks

Assuming one has access to a dataset containing an ensemble of models describing an expected geological spatial pattern (e.g., facies model). It is believed the data included in this dataset follow an implicit probability distribution $\boldsymbol{p}_{\boldsymbol{data}}$. If $\boldsymbol{p}_{\boldsymbol{data}}$ can be explicitly formulated, additional models can be generated for various applications (i.e. one would be able to generate model realizations $\boldsymbol{x}$ with the same statistical distributions as in training dataset). However, in earth-related problems it is usually impossible to find such an explicit expression. Therefore, generative models are proposed to approximate the distribution, among which neural networks are widely used due to their strong fitting capacity. GAN [17, 18] is a deep learning method consisting of two deep neural networks, the Generator (G) and the Discriminator (D). The GAN is fed with a random latent vector $\boldsymbol{z}$ usually sampled from a normal distribution $\boldsymbol{z}\sim p(\boldsymbol{z}) = N(0, 1)$, and is expected to output a real-like image $\boldsymbol{x}$ that reproduces the main properties of $\boldsymbol{p}_{\boldsymbol{data}}$. The generation process is achieved by G, which is trained to map samples from $\boldsymbol{z}$ into $\boldsymbol{x}$. Then, the generated image and the real image are input into D, which is trained to estimate a probability indicating an image being real (from the training dataset) or fake (generated by G). G and D are trained alternatively by optimizing the following min-max objective function to enhance their generative and discriminative ability:

$$\min_G \max_D \left\{ E_{x\sim p_{data}}[\log D(\boldsymbol{x})] + E_{z\sim p_z}\left[\log\left(1-D\left(G\left(\boldsymbol{z}\right)\right)\right)\right]\right\} \quad (1)$$

where G and D are the Generator and the Discriminator, $\boldsymbol{p}_{\boldsymbol{data}}$ and $\boldsymbol{p}_{\boldsymbol{z}}$ are the distribution of training data and latent vectors. $\boldsymbol{x}$ and $\boldsymbol{z}$ represent training models and latent vectors, respectively.

By optimizing the loss function, D is trained to distinguish between samples from $\boldsymbol{p}_{\boldsymbol{data}}$ and from G. G is expected to generate samples like those from $\boldsymbol{p}_{\boldsymbol{data}}$ and fool D to treat them as real ones.

A full mathematical description of the most common GANs applied in geosciences can be found in the following reference works [1, 2, 7, 8, 11, 17, 18, 38, 40].

2.2 Problem statement

In this work we aim at generating realistic facies modes using GANs. A set of facies models representing some probability distribution and the expected sedimentary environment is used as the training dataset. In the application examples shown next we created a training dataset using geostatistical simulation with multi-point statistics [41]. Figures 2 and 3 show part of the training dataset, which consists of 10,000 binary facies models with size of 80 × 100 in $\boldsymbol{i}$- and $\boldsymbol{j}$-directions, respectively. The training dataset aims at reproducing a sedimentary environment mainly dominated by carbonate mounds. The mounds are represented by zero (i.e., black facies), while the background facies are represented by one (i.e., white facies). In the generation of the training dataset two vertical wells placed close to the boarders of the model were considered as experimental data (Fig. 2b). These data, the training dataset and the well data, were used to develop a deep generative model that is able to create diverse unconditional and conditional facies realizations honoring the pre-existing conditioning data (i.e., the well data).

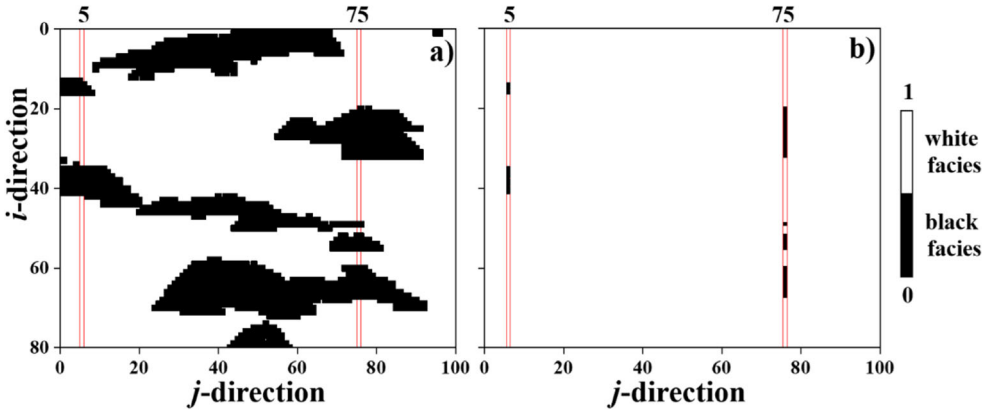

Fig. 2 **a** Example of facies models used as training dataset and **b** experimental data along two wells used as experimental data

2.3 DCGAN model for unconditional facies simulation

To generate unconditional facies models, we employ a DCGAN [37] where G and D are designed in the form of convolutional neural networks (CNN), building upon the powerful image-feature extraction capability of CNN. The G and D architectures in the DCGAN used herein are depicted in Fig. 4. The number of CNN layers and parameters of each convolution layer are calculated according to the size of the models within the training dataset and the size of the desired output. G maps facies models using transposed convolution layers from a random 100-dimension vector $\boldsymbol{z}\sim N(0, 1)$. D outputs a scalar ranging between 0 and 1 indicating whether the image is from the real dataset (i.e., the training dataset) or from G. Data in G and D is in the format of $(\boldsymbol{b_z,N,W,H})$, where $\boldsymbol{b_z}$ is the batch size, $\boldsymbol{N}$ is the channels, $\boldsymbol{W}$ is width and $\boldsymbol{H}$ is length. Except for the first and last layers, each transposed convolution layer and convolution layer is followed by a batch normalization layer and a leaky rectified linear unit (LeakyReLU) function except the last one, which is a Tanh function in G and Sigmoid function in D.

It is worth to note that all the models are normalized to [−1, 1] before being input into D. The generated models are then transformed into the original domain, in our case [0, 1]. Moreover, as a post-processing step of the unconditional generated models values smaller than 0.5 are set as 0, while values bigger than 0.5 are set as 1 in order to restore the models to binary state.

A set of key parameters that need to be specified and affect the performance of GAN is normally designated as hyperparameter. Hyperparameters are those parameters that cannot be learned by the network (i.e., weights of the neural

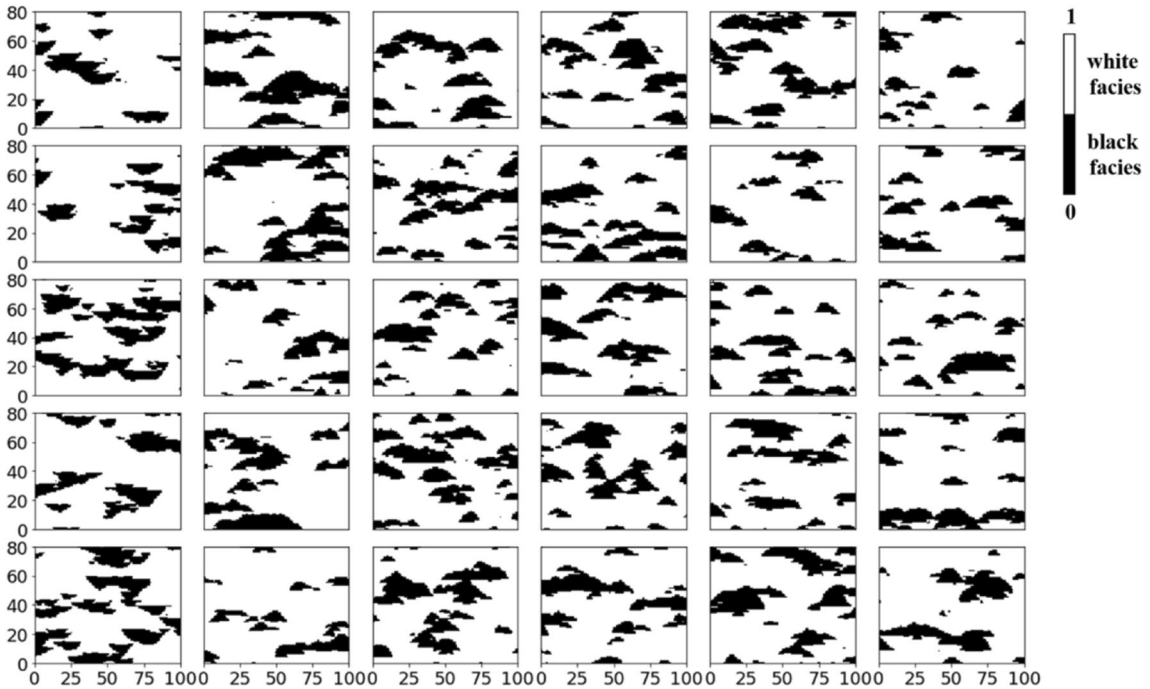

Fig. 3 Examples of facies models used as training dataset

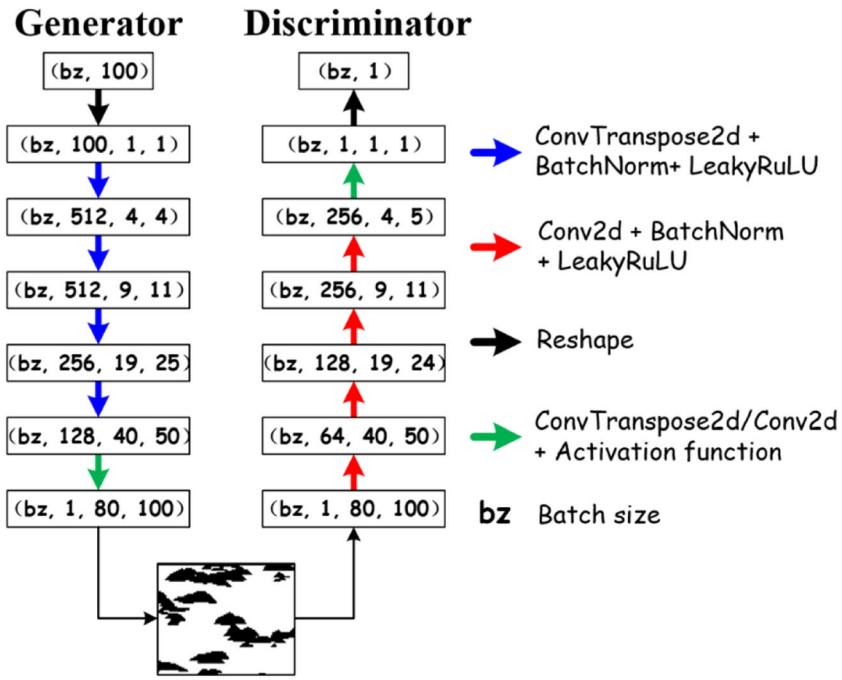

Fig. 4 Architecture of G and D of the unconditional DCGAN

network) such as the learning rate and training epochs. The hyperparameters used in the training of the DCGAN in the examples shown herein are summarized in Table 1.

2.4 U-Net GAN framework for conditional facies model generation

In order to ensure geological plausibility, subsurface models should be able to reproduce exactly available direct measurements of the property of interest at their location. Under a deep learning framework this has been achieved following a two-step approach, adapting the semantic inpainting method (e.g., [6, 16, 35, 47]). Semantic inpainting refers to the task of inferring missing contents in images based on image semantics. Suppose there is a corrupted image, $\boldsymbol{x}_r$, the objective of semantic inpainting is to fill-in the corrupted part. If G is trained with uncorrupted data, $\boldsymbol{x}\sim\boldsymbol{p}_{data}$, it learns the map from the low

Table 1 Hyperparameters for training unconditional DCGAN

Parameter	Values
Epochs	200
Batch size	64
Optimizer	Adam
Learning rate	0.00005
Betas of Adam	(0.5, 0.999)
Dimension of z	100
Loss function	Binary Cross-Entropy Loss

the dimension latent vector space $\boldsymbol{p}_z$ to the high dimension distribution $\boldsymbol{p}_{data}$. It is believed that there is a "closest" image $\boldsymbol{x}^* = G(\boldsymbol{z}^*)$ to the corrupted image $\boldsymbol{x}_r$.

The match between $\boldsymbol{x}_r$ and $\boldsymbol{x}^*$ is obtained by solving an optimization problem (Fig. 1). Specifically, two differentiable loss functions are designed; a prior loss to penalize unrealistic models and a content loss to make the generated models closer to the conditioning data locally at the locations of the direct observations. A vector $\boldsymbol{z}_0$ is randomly sampled from the latent space after the GAN training process. Then, a model is generated through the trained G to compute both losses and its derivative to vector $\boldsymbol{z}$, which is used to update the location of $\boldsymbol{z}$ in latent space. The optimization process is iterated until it reaches some stopping criterion. Finally, an optimal vector, $\boldsymbol{z}^*$, from which the generated model is close enough to the corrupted one is found.

This approach has been successfully applied in facies modeling and allows the reproduction of constraining data in generated models but with limitations in the accuracy of the models (e.g., [6]) and in generating ensembles of models with variability (e.g., [16]). Moreover, the computational burden of the additional optimization step may be considerable.

To tackle these problems, we propose herein the idea of image-to-image translation [21] for model conditioning using GANs. Image-to-image translation refers to learning a mapping relationship from the latent vector and one image to another instead only from latent vector as in the semantic inpainting problem.

GANs used for image-to-image translation belong to conditional Generative Adversarial Network (cGAN, [32]). GAN models learn the data distribution after training and are then

able to generate new models. However, the model generation process cannot be controlled, and the samples are randomly generated without any local constraint. To make the generation process controllable, a U-Net GAN framework is proposed. This model is constructed by adding additional constraining information to G and D during training. This additional information can be a category label or other auxiliary information such a text, a vector or even an image. In our example we use the data available at the borehole locations. Then, the GAN is capable of outputting samples based on the complementary auxiliary information.

In the image-to-image translation framework, a dataset containing paired images $\{\boldsymbol{X}, \boldsymbol{Y}\} = \{\boldsymbol{x_i}, \boldsymbol{y_i}\}_{i=1}^{N}$ is prepared firstly [49]. $\boldsymbol{x_i}$ is the image we want to translate, while $\boldsymbol{y_i}$ is the translation objective. For example, $\boldsymbol{x_i}$ is a scenic photo taken during the day and $\boldsymbol{y_i}$ is the picture of the same place but taken at night. Then, we can do a scenery transfer task where we translate the landscape photo from day-to-night. For the image-to-image task, we want to train a model able to learn a map from $\boldsymbol{X}$ to $\boldsymbol{Y}$. The G is fed with an input $\boldsymbol{x_i}$ and outputs an image $\boldsymbol{y_i}$. The D is fed with two images $\boldsymbol{x_i}$ and $\boldsymbol{y_i}$ then outputs a probability indicating how real $\boldsymbol{x_i}$ is and whether $\boldsymbol{x_i}$ to $\boldsymbol{y_i}$ are paired. This approach has been proved to be effective at generating images from labeled maps, reconstructing objects from edge maps and colorizing images [32, 49].

We extended this concept to generate conditional facies models. In our framework, $\boldsymbol{x_i}$ indicates a facies model from the training dataset, while $\boldsymbol{y_i}$ is model containing exclusively the conditioning data (e.g., well-data as in the application examples show below) (Fig. 5). Then, the dataset containing the facies model and the corresponding extracted conditioning data is used for the training process. The G is fed with a conditioning model $\boldsymbol{y_i}$ as well as a random latent vector $\boldsymbol{z}\sim\boldsymbol{N}(\boldsymbol{0}, \boldsymbol{1})$ from Gaussian distribution and outputs facies realizations $\boldsymbol{x_i}$. It is expected to learn not only to map a realistic sample from vector $\boldsymbol{z}$ but map to accurate facies data from the conditional image.

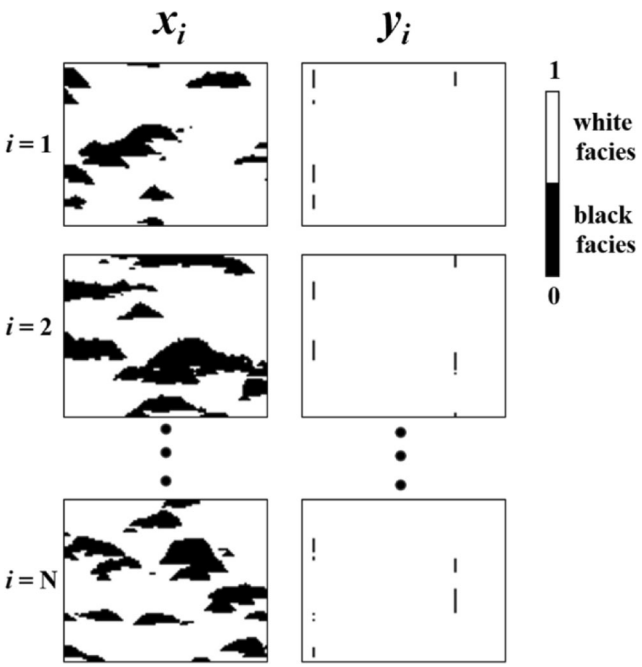

Fig. 5 The paired dataset consisting of N facies models and corresponding extracted conditioning well data

To tackle the image-to-image problem, we developed a U-Net GAN [21]. U-Net was firstly designed for biomedical image segmentation by Ronneberger et al. [38], which consists of a contracting and an expansive path. Isola et al. [21] demonstrate the ability of U-Net to perform image-to-image tasks due to its capacity for image feature extraction and image reconstruction. The architecture of the U-Net GAN used in our application examples is shown in Fig. 6 and consists of two branches.

The downscaling process represented in the left branch of the U-Net structure (Fig. 6) extracts implicit features of the input conditioning image in different level by convolution layers (encoder layers). Then, the upscaling process in the right branch (Fig. 6) rebuilds the image from the extracted feature maps by transposed convolution layers (decoder layers). The skip connections [15] between the two parts of the U-Net are able to pass the extracted features in each downscaling layer to the upscaling layers to enhance the capability of the convolution layers to learn the pattern of the conditional image.

Isola et al. [21] have demonstrated that adding a vector $\boldsymbol{z}$ to the G has little effect on the stochasticity of the output samples. Instead, they provided noise only in the form of dropout layers. However, they found that the generated samples using either latent vector or dropout layers fail to generate ensembles of models with high variability. Therefore, to train a G that can produce samples showing high variability, the architecture of G has to be improved.

On the basis of the U-Net structure, the latent vector $\boldsymbol{z}$ is upscaled four times using transposed convolution layers (named noise-layers). The feature maps of each layer are concatenated with those in U-Net. The left encoder part of U-Net is intended to extract features of the conditioning data and the noise-layers help to increase output sample diversity by adding noise to feature maps, while the right decoder part aims to generate real-like and conditioned facies models using the feature maps from the encoder layers, the lower decoder layers as well as the noise layers.

The loss function of GAN exerts significant influence on the update of model parameters and the quality of generated samples. Normally, by optimizing the loss function given Eq. (1) alternatively, G should learn to generate plausible samples and D is expected to output high values for real images or small values for fake ones, which forces G to produce realistic samples. Equation (1) may be helpful to generate plausible models but cannot guarantee the sample condition and model variability. Therefore, to

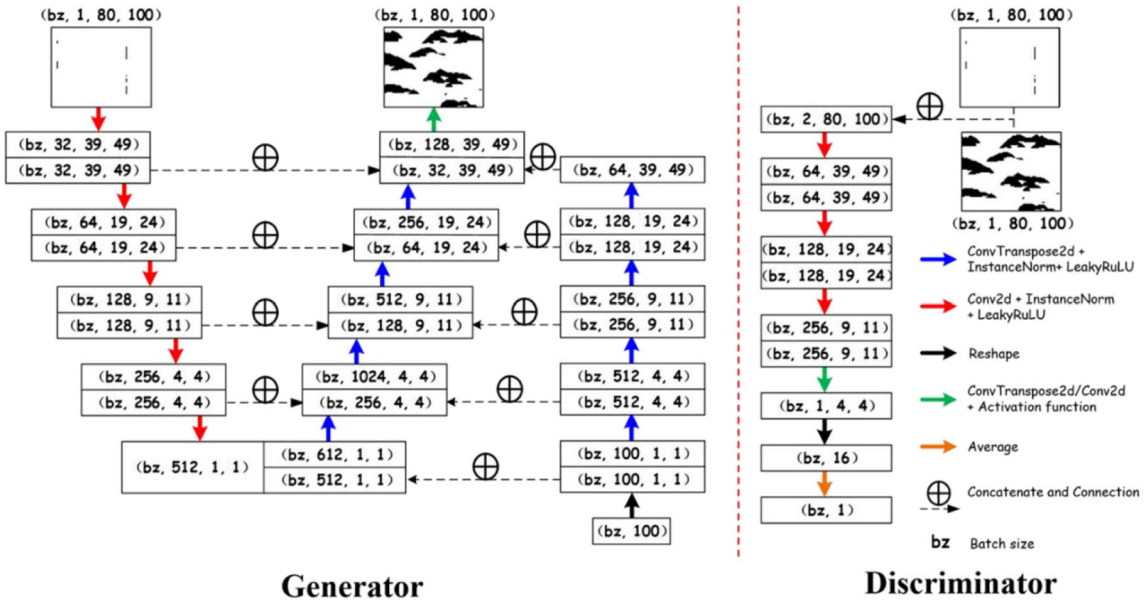

Fig. 6 Schematic representation of the architecture of the ***G*** and the ***D*** of the improved U-Net GAN used in this work

make sure the generated models are constrained at experimental data locations, a content loss $\boldsymbol{L_c}$ is introduced to compute the L1 distance (Eq. (2)) between generated model and conditioning data at specific locations.

$$L_c = \left\| y - G(\boldsymbol{y}, \boldsymbol{z})_y \right\|_1 \tag{2}$$

where $\boldsymbol{L_c}$ is the content loss, y represents the conditional image, $G(\boldsymbol{y}, \boldsymbol{z})_y$ denotes the extracted facies data from the generated sample. Optimizing the content loss would help G to give constrained images.

The last problem to solve is how to increase the variability of the generated models. During the training process of GAN, model collapse is a very common problem [1, 2, 17, 31]. G just ignores the variability of input noise vector z and keeps generating similar samples, which are able to fool D. As the generated models are realistic, D fails to distinguish them from the real ones and is not concerned about the variability within the ensemble of generated models. In these cases, the training of the GAN is considered to be trapped into a local minimum. This is because the training of GAN using the min-max loss function (Eq. (1)) is inherently unstable problem [1, 2, 17, 31]. Many improvements have been implemented to deal with this problem. Yang et al. [45] and Mao et al. [27] proposed a simple yet effective approach to address this issue. They formulated a loss term to maximize the ratio of the distance between generated images with respect to the corresponding latent vectors. This loss term encourages G to explore the latent space and produce variable and informative outputs depending on latent vectors. Otherwise, it will penalize G for its model collapse behavior. The diverse loss $\boldsymbol{L_d}$ is defined by:

$$L_d = \frac{d\left(G(\boldsymbol{y}, \boldsymbol{z_1}), G\left(\boldsymbol{y}, \boldsymbol{z_2}\right)\right)}{d(\boldsymbol{z_1}, \boldsymbol{z_2})} \tag{3}$$

where $\boldsymbol{d}(\cdot)$ denotes the distance metric of two distribution. In this paper, $\boldsymbol{d}(\cdot)$ is set as L1 norm distance (Eq. (2)). $\boldsymbol{z_1}$ and $\boldsymbol{z_2}$ represent two latent vectors. $\boldsymbol{G}(\boldsymbol{y}, \boldsymbol{z_1})$ and $\boldsymbol{G}(\boldsymbol{y}, \boldsymbol{z_2})$are two generated samples from $\boldsymbol{z_1}$ and $\boldsymbol{z_2}$, respectively.

Therefore, the training loss function for G and D are defined as follows:

$$L_D = -\log D(\boldsymbol{y}, \boldsymbol{x}) - \log[1 - D(y, G(\boldsymbol{y}, \boldsymbol{z}))] \tag{4}$$

$$L_G = -\log[D(\boldsymbol{y}, G(\boldsymbol{y}, \boldsymbol{z}))] - \lambda_d L_d + \lambda_c L_c \tag{5}$$

where $\boldsymbol{\lambda_c}$ and $\boldsymbol{\lambda_d}$ indicate the weights of content loss and diversity loss.

The contribution of our work is the combination of the U-Net structure with the noise layers, the content loss (Eq. (2)) and the diversity loss (Eq. (3)). A trained G is expected to immediately generate realistic facies models, constrained and variable samples given some conditioning data.

For the application examples shown next, hyperparameters used for training the U-Net GAN are summarized in Table 2. As for the training of unconditional GAN, the training models are normalized in the range [−1, 1] and the generated models are transformed to 0 and 1. Furthermore, some noise sampled from normal distribution are added to both real samples and generated before they are input into the D to make it more robust.

Table 2 Hyperparameters used to train the conditional GAN

Parameter	Values
Epochs	1000
Batch size	64
Optimizer	Adam
Learning rate	0.0005
Betas	(0.5, 0.999)
Dimension of z	100
λ_d	0.05
λ_c	100
Loss function	Binary Cross-Entropy Loss

2.5 Quality assessment metrics

In this section we describe the quantitative metrics applied to evaluate the models generated by the two GAN models. We focus on the comparison of visual features and statistical properties between the models in the training dataset and the generated ones. The statistical properties are assessed by the pixel ratio, defined as the ratio of facies 0 to the total number of grid cells in the model.

The variability within the ensemble of generated models is measured by image similarity metrics, which is computed by the Mean Squared Error (MSE, Eq. (6)) and the Structural Similarity Index (SSIM, Eq. (7); [20, 44]). The lower the MSE score is, the more similar the two images are with a value 0 being identical, while situation for SSIM is on the contrary, a value of 1 represents a total match between two models. The MSE is defined as:

$$MSE(\boldsymbol{x},\boldsymbol{y}) = \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{x}_i - \boldsymbol{y}_i)^2 \tag{6}$$

where $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$ represent two models, $\boldsymbol{N}$ is the number of grid cells, which in our application examples is 8000.

The SSIM is defined as:

$$SSIM(\boldsymbol{x},\boldsymbol{y}) = \frac{\left(2\mu_x\mu_y + c_1\right)\left(2\sigma_{xy} + c_2\right)}{\left(\mu_x^2 + \mu_y^2 + c_1\right)\left(\sigma_x^2 + \sigma_y^2 + c_2\right)} \tag{7}$$

where $\boldsymbol{\mu_x}$ and $\boldsymbol{\mu_y}$, $\boldsymbol{\sigma_x^2}$ and $\boldsymbol{\sigma_y^2}$ are mean values and variance values of $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively. $\boldsymbol{\sigma_{xy}}$ is the covariance of $\boldsymbol{x}$ and y. $\boldsymbol{c_1}$ and $\boldsymbol{c_2}$ are two constants, and for gray image the values are $0.01^2$and 0.03^2.

Furthermore, the multidimensional scaling (MDS) technique [10] is used to evaluate the variability and similarity of the generated models [5]. With MDS, high-dimensional data are scaled down to a low-dimensional representation that can be plotted in a cartesian space while keeping their similarity, which is measured by their distance in the metric space. If data from two dataset or distribution are projected in the same coordinate system, the range of the point-cluster can be used to evaluate their similarity and variability. The cluster range should be comparable if the two datasets are similar, otherwise it should be distinguishable. Also, the projected points should be sparsely distributed rather than concentrated on a small cluster if the original data have diverse patterns. Being regarded as an 80×100 matrix, each generated facies model is reshaped to a vector with 8000 samples and then projected to a 2-D point considering an Euclidean distance when computing the MDS. By checking the distance between these points, the similarity and the diversity of the samples can be easily observed. Additionally, if the GAN model learns the real distribution, the point-cluster of the real facies models and the reproduced ones should be close, which can be used to measure the reality of the GAN models.

Finally, the content loss $\boldsymbol{L_c}$ and the conditioning accuracy are used to measure the conditionality of the generated model from the U-Net GAN. For a single image, the content loss measures the total pixel error of the pixels at the well locations. The conditioning accuracy is computed through dividing the content loss by the total pixels in the conditioning well locations.

3 Results and discussion

In this section we show the results obtained by applying the DCGAN to generate unconditional facies models and U-Net GAN to generate facies models conditioned to experimental sample.

3.1 Generating unconditional facies models

A set of facies models generated with the unconditional GAN is shown in Fig. 7. A visual comparison does not allow to distinguish whether samples are from the training dataset or from the unconditional GAN. The unconditional GAN reproduces the general shape of the facies and their spatial continuity. In fact, the area of the model filled-in by the carbonate facies in the generated models resembles the one in the training dataset. The pixel ratios of the 10,000 models of the training dataset and the 10,000 generated ones are shown in Fig. 8. It can be observed that the distribution of facies data of the two datasets is nearly identical. They have the same mean and similar standard deviation (relative error < 0.4%), meaning the two datasets are statistically indistinguishable.

Additionally, to assess the similarity between each model, three sets of 1000 models were randomly retrieved from the training dataset, the latent vector $\boldsymbol{z}$ and the ensemble of generated models. These models were plotted in the MDS space (Fig. 8c). The MDS plot shows that the models sampled from

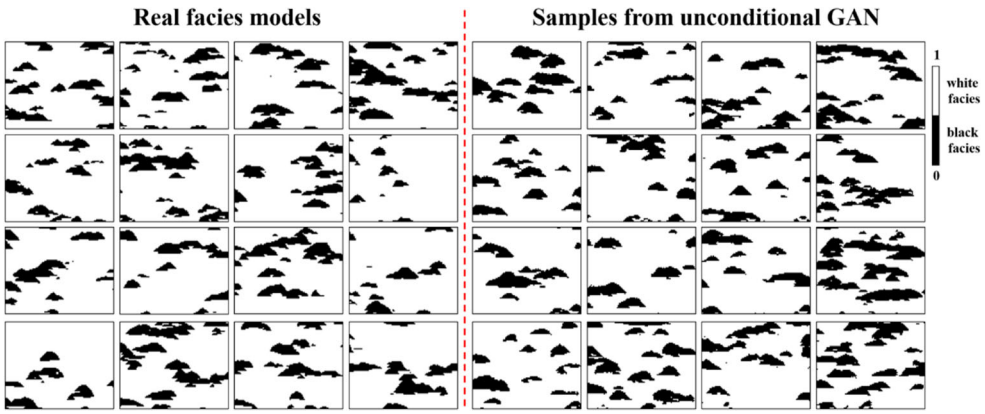

Fig. 7 Real facies models and fake realizations from unconditional GAN

the training and the generated sets are similar, as they are scattered within the same region of the MDS space, assessing quantitatively that facies models from both sets do have similar spatial distribution.

In other word, the GAN model can not only reproduce the visual pattern of the training facies models but can learn the intrinsic statistical features such as the spatial continuity pattern. However, it is worth to note that the DCGAN model still has not captured the full variability of the training dataset, as the generated models are not able to fully cover the space sampled by the training dataset. This limitation can be addressed by optimizing the hyperparameters of the GAN to obtain a better model. Finally, the cluster size of GAN samples compared with that of latent vectors reveals that GAN model can map the random low-dimensional normal distribution vector to diversified high-dimensional facies models. These results indicate that the unconditional DCGAN can generate realistic facies models both visually and statistically.

To assess the variability of the samples within an ensemble of generated models, the MSE and SSIM values between each pair of models in the training dataset are computed and then averaged to two single indexes, which are listed in Table 3. The mean MSE value and mean SSIM value of original facies models are 0.32 and 0.26, respectively. It suggests that images of the training dataset are very different from each other. The mean MSE value and mean SSIM value of samples generated from the unconditional GAN are 0.32 and 0.7 respectively. They are very close to the real ones (within a relative error of 0.77%), suggesting the facies models generated by the unconditional DCGAN are variable. Additionally, from the MDS plot it can be interpreted that the low-dimensional points are

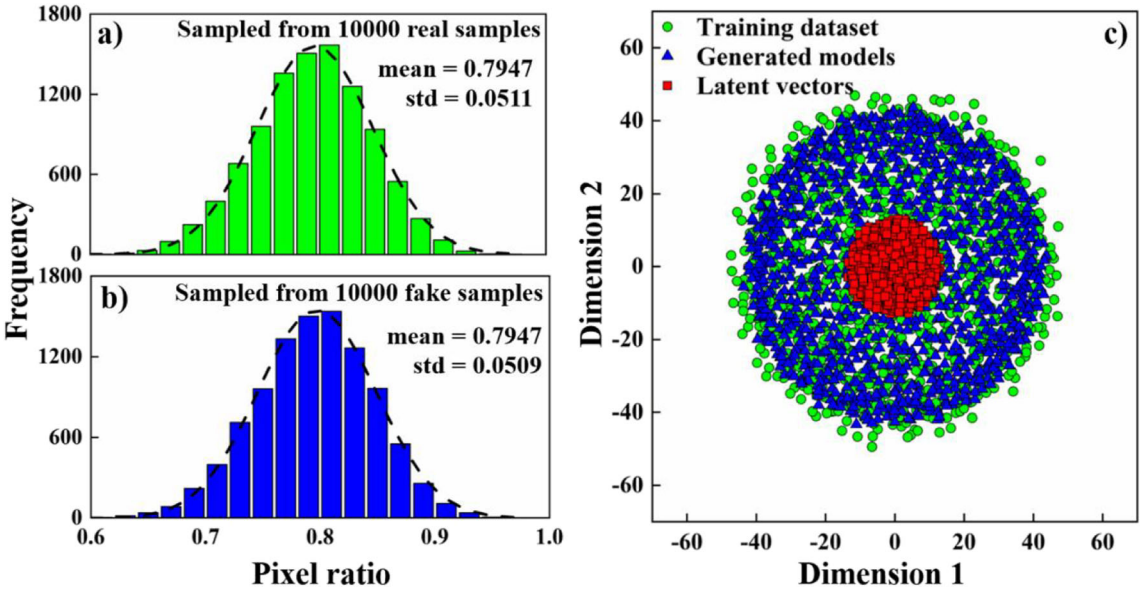

Fig. 8 Histogram of pixel ratios extracted from: **a** 10,000 models from the training dataset; and **b** 10,000 models from the unconditional GAN. **c** MDS plot of models sampled from the training dataset, the latent vector and the ensemble of generated models

Table 3 MSE value, SSIM value and content loss value of different cases

Scenario	MSE	SSIM	L_c	Conditioning accuracy
Original data	0.3263	0.2634	–	–
ucGAN	0.3238	0.2652	–	–
Basic example	0.2238	0.4367	0.0012	99.93%
1 well	0.3183	0.3246	0.0021	99.74%
3 wells	0.2570	0.3964	0	100%
4 wells	0.2637	0.3826	0.0033	99.90%
Simplified U-Net GAN	0.2211	0.4497	0.0010	99.94%

uniformly distributed in the MDS space. So, the unconditional DCGAN learns to produce samples showing diverse patterns.

3.2 Generating facies models conditioned to two wells

Figure 9 shows a set of illustrative facies models generated with the U-Net GAN and locally conditioned to two wells. To facilitate display and analysis, the facies data at the well locations of the generated samples are extracted and drawn in one figure with the real well data, where the real facies log is in the leftmost and the fake ones are arranged at an interval of 5 pixels from left to right.

The models generated with the U-Net GAN are visually indistinguishable from the real ones. For the statistical features, the pixel ratio of conditional facies models would be different from unconditional facies models. The presence of the conditioning data changes the statistics of the model since they must honor locally the information provided by the well data. We discuss the influence of the conditioning data on the statistics reproduction in Section 3.3.

A visual inspection shows that the generated samples are all well constrained to the conditioning data. Figure 10 shows the probability model of facies one and the variance model computed from an ensemble of 1000 U-Net realizations. The probability model of facies one shows the ability of the network to reproduce the well data, while keeping variability as illustrated by the equal probability of occurrence for unconstrained locations within the model. As expected, the probability of occurrence for these locations is approximately 0.5 (Fig. 10). The variance model shows a similar behavior of those resulting from geostatistical simulation, the variance is zero at the location of the experimental data and increases as the distance increases from these locations. Quantitively, Table 3 shows that the average L_c of 10,000 samples is 0.0012 and the conditioning accuracy is 99.93%. This a negligible error, which means the facies data at two well locations of generated samples match very well with the pre-exist conditioning data.

Besides reproducing the observed data, the most important feature is the variability within the ensemble of generated facies models. As in a conventional geostatistical simulation, the U-Net GAN should produce samples different from each other, even if they are conditioned to the same well data. It can be inferred from the mean MSE value and mean SSIM value in Table 3 that these conditioned samples have good variability, even though they are not as good as those from unconditional GAN model. This is because areas around the well locations tend to have the similar spatial pattern. This effect

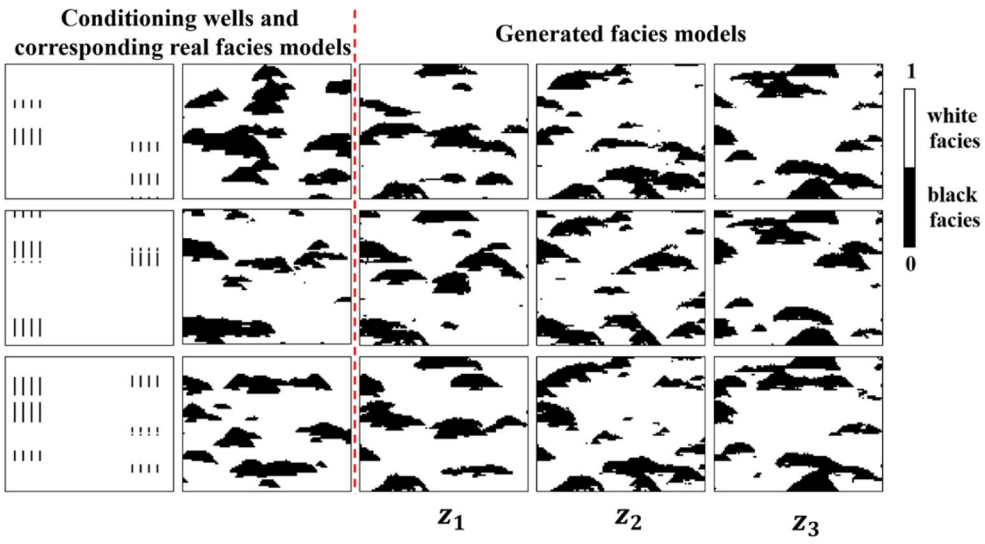

Fig. 9 Random models conditioned to two wells. Each row represents a different well configuration

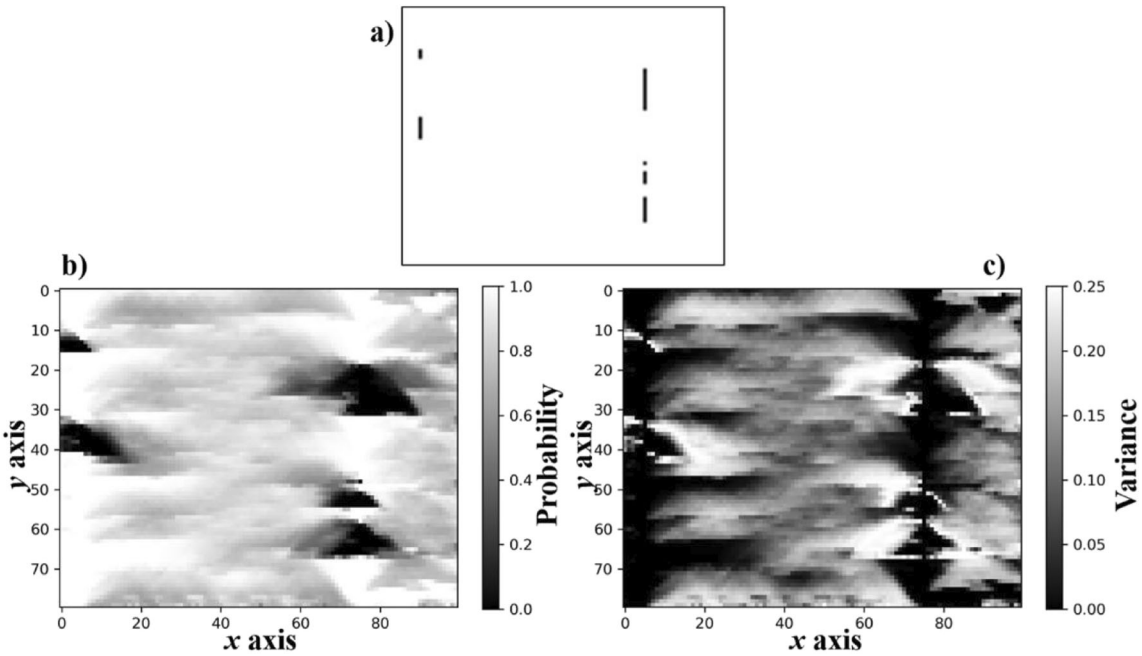

Fig. 10 **a** The conditioning data, **b** the white-facies probability model and **c** the variance model computed from an ensemble of 1000 U-Net realizations conditioned to two wells

is also observed in geostatistical simulation where the variance within an ensemble of realizations increases when moving again from the location of the experimental data.

Figure 9 shows three scenarios conditioned to different well sets and the corresponding facies models produced from three different latent vectors (i.e., z_1, z_2, z_3). Different latent vectors produce variable facies modes. However, when feeding the U-Net GAN model with same $\boldsymbol{z}$ but different conditioning data, the generated samples show some similar features. This phenomenon is resulted from the model structure and the diversity loss function. The U-Net structure takes the conditioning well data as input and it tends to generate samples following the information contained in these data. The information delivered to the U-Net structure from convolution layers processing latent vector are treated as auxiliary information designed to increase the uncertainty of the output. In other word, the U-Net framework is designed to form the basic structure of the facies model, while the noise-layers as well as the diversity loss function is expected to vary the facies data on the basis of the facies sample from U-Net.

To verify this possibility, a U-Net GAN model only consisting of U-Net structure without noise-layers and $\boldsymbol{L_d}$ is trained and the results are shown in Fig. 11. In this case, the generative model can still give well-constrained facies

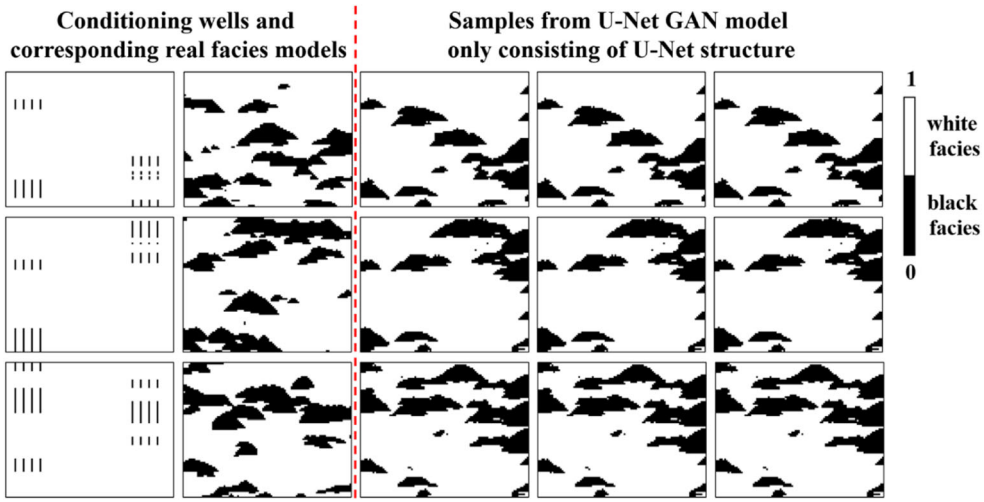

Fig. 11 Samples from U-Net GAN model only consisting of U-Net structure

models, but all the realizations are same (i.e., MSE = 0, SSIM = 1) once the conditional well image is fixed. The generated sample from a well-trained model is dependent more on the input conditioning well data than the latent vector z. Therefore, it varies when any one of these two factors change but will change more when the conditional image changes.

3.3 Generating facies models conditioned to multiple wells

Besides constraining the model to two wells, we explored the training of the proposed U-Net GAN framework conditioned to one, three and four wells. To test the performance of the U-Net GAN structure, the architecture of the G and the D remain the same in all cases while the conditioning well location is changed to (50) for one well, (5, 40, 75) for three wells and (5, 30, 50, 75) for four wells. The learning rate in each case is adjusted to obtain better results. The optimal learning rate of each case used in this paper is 0.0001 for one-well, 0.0005 for three-wells and 0.0002 for four-wells. The results are displayed in Fig. 12. The generated facies models are visually similar to the real ones while honoring the conditional well data. Also, the probability models of facies one and the variance models from a set of 1000 U-Net GAN realizations in each scenario are shown in Figs. 13, 14 and 15. These models show the ability of the network to match considerably well the conditioning data, while keeping the variability of the simulated ensemble of models. Table 3 summarizes the performance in sample variability and conditioning accuracy of the U-Net GAN models.

Figure 16 shows that the mean value of each scenario is similar to the true one while the standard deviation (std) tends to decrease. Moreover, the range of pixel ratio values gets smaller with the increase of the number of conditional wells. This is expected as there are more cells within the model assigned to experimental data and therefore less possibilities to create variable models. This effect can also be observed in the MSE and SSIM values, which are both smaller than the real ones. Therefore, the distribution of all pixel ratios of four conditional cases are not as wide as the real one and the unconditional one. And the range would get smaller with more conditional wells.

Figure 17 depicts the MDS results of the four scenarios. It can be inferred that the cluster range of generated samples is getting smaller when there are more conditioning wells, which is in consistence with changing trends of pixel ratios. From one hand, this means the conditional U-Net GAN is not able to learn the full distribution of the training dataset because of the conditioning data. From the other, this illustrates that the diversity of these samples is decreasing. The same effect is also observed when we consider the first N dimensions responsible to explain 75% of the original variance of the ensembles. Figure 18 shows the Euclidean distance of each model within

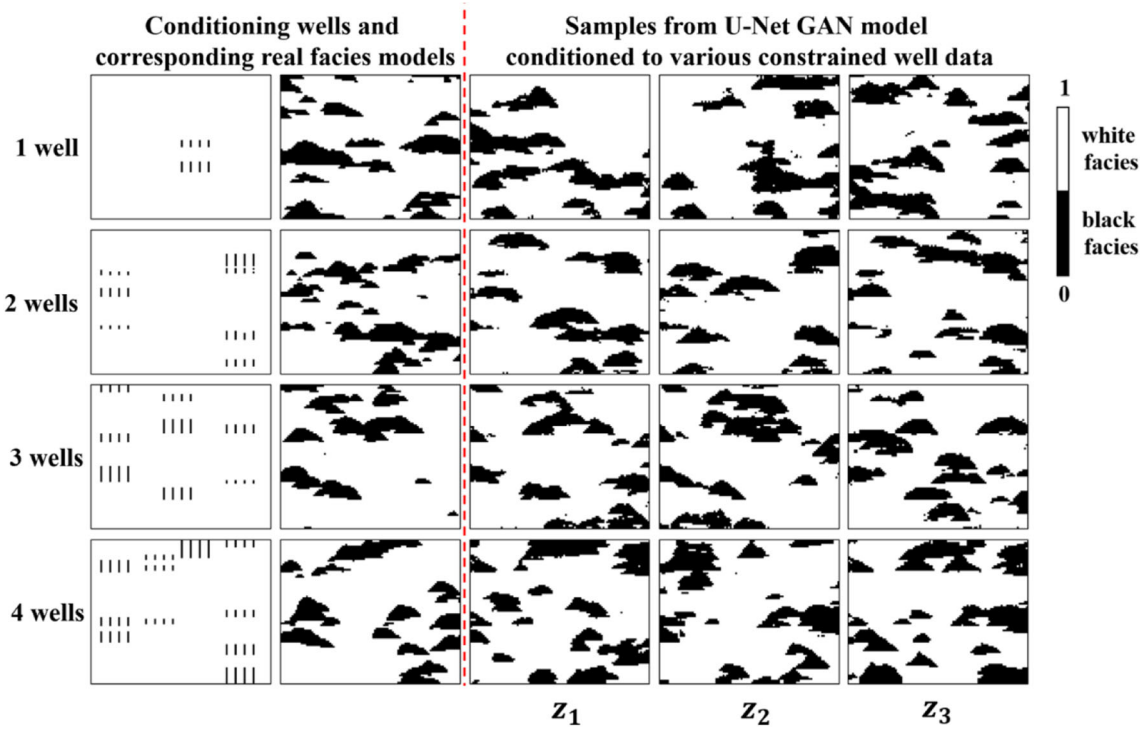

Fig. 12 Facies models resulting from the U-Net GAN conditioned to 1 well, 2 wells, 3 wells and 4 wells. Each row represents a conditioning scenario

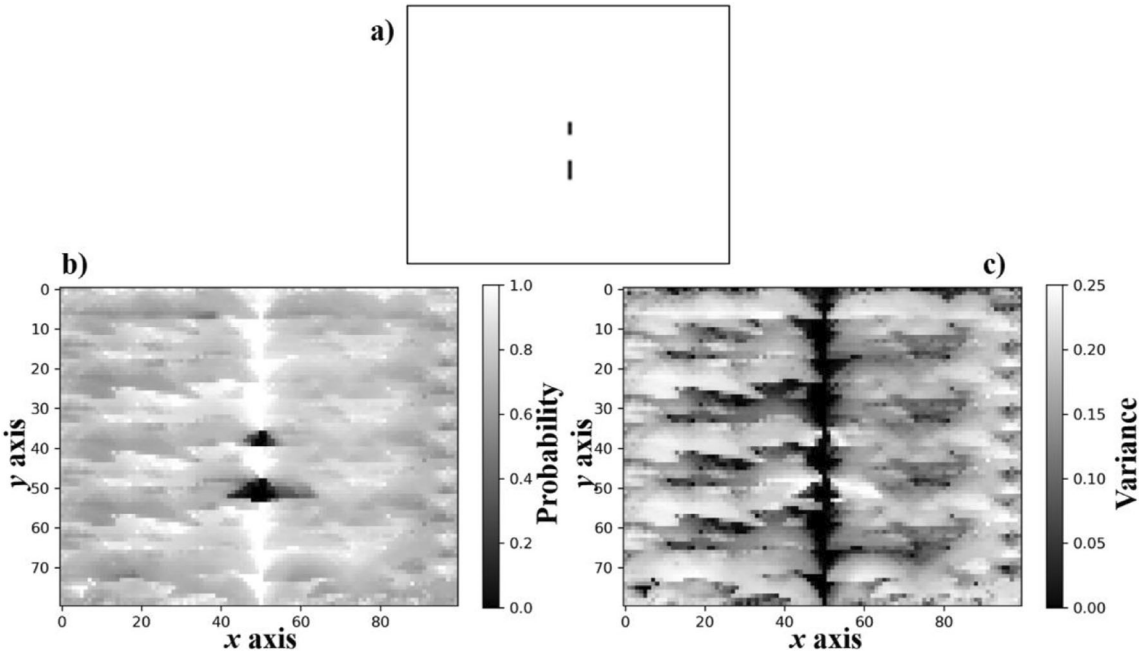

Fig. 13 **a** The conditioning data, **b** probability model of facies 1 and **c** the variance model computed from an ensemble of 1000 U-Net realizations conditioned to one wells

the ensemble of facies models generated with the proposed GAN framework and the origin of the MDS space. It is clear that as the number of conditioning wells increase the variability of the models decrease (i.e., are models all located within the same region in the MDS space). The lack of variability, a characteristic of models produced by GANs, might be in this example enhanced by the relatively small number of cells in the model,which do not allow much variability when we start imposing locations with experimental data that need to be reproduced.

3.4 Influence of U-net structure and loss functions

The U-Net architecture of the GAN model and the two loss functions are key elements significantly affecting the results.

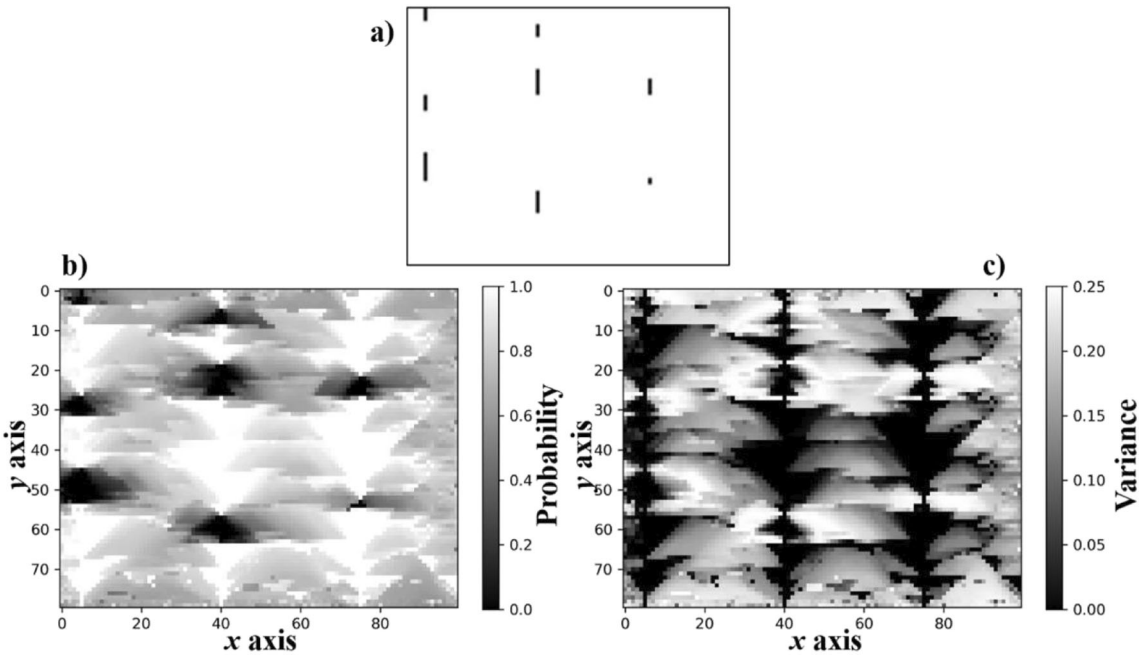

Fig. 14 **a** The conditioning data, **b** probability model of facies 1 and **c** the variance model computed from an ensemble of 1000 U-Net realizations conditioned to three wells

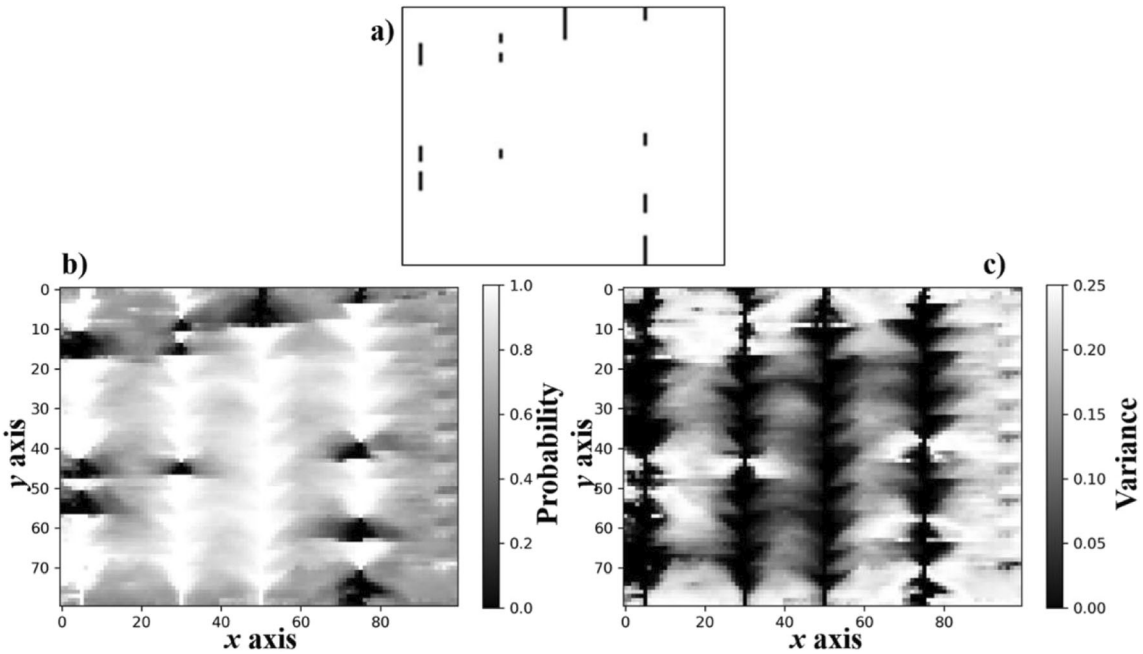

Fig. 15 **a** The conditioning data, **b** probability model of facies 1 and **c** the variance model computed from an ensemble of 1000 U-Net realizations conditioned to four wells

In this section, we assess the influence of both factors on the generation of facies models.

For the GAN structure proposed herein, there are more than 10 million parameters in G and only 470 thousand parameters in, indicating that the Generator's parameters may be redundant. Therefore, the U-Net Generator structure is simplified and then the model is trained to test its performance. The channels of all the convolution layers are reduced and the convolution layers processing the latent vector are removed so the concatenation is only implemented in the bottom-neck layer. The new G structure consisting of 1.7 million parameters is shown in Fig. 19. The D structure remains unchanged and the batch size is set as 128. Figure 20 depicts a comparison of results obtained from the original G structure and the simplified structure. The results show that the simplified model can still generate satisfying samples which are visually indistinguishable. Moreover, we achieve sample variability when either the conditional well data and the latent vector is fixed and the other is changed, indicating a great visual diversity which can also be proved by the MSE value (0.2211) and SSIM value (0.4497) in Table 3. And the $\boldsymbol{L_c}$ value (0.0010) and conditioning accuracy are very close to that of the original U-Net GAN model and small enough for a good conditionality. Thus, the simplified U-Net GAN model works as well as the original model with less training time and lower requirement for hardware.

Although most of the convolution layers processing the latent vector designed to add noise to feature maps are removed, the samples still show good diversity and conditionality. It proves that with limited parameters the U-Net framework still has a capacity to fit the data distribution. However, it is worth to note that the simplified model is not robust enough as most of the times this configuration results in loss of diversity (i.e., model collapse). To avoid this result, the simplified structure needs a more careful parameterization and in general takes additional time to converge to generate satisfying results (i.e., conditioned facies models with diversity within the ensemble). Concatenating noise-layers with the U-Net structure aims at increasing sample diversity. For this reason, we recommend the complex structure with more noise-layers, since it results in more reliable and robust models. Theoretically, it should work even if there is only one noise-layer (i.e., we only add noise in the bottom-neck layer). However, this configuration is not robust enough in model generation and in this configuration the diversity loss function is the parameter that drives the model generation. The content loss function and the diversity function are expected to play a role in generating well-constrained and diverse samples. Therefore, the influence of the two loss functions are explored.

Based on the basic two-well conditioned example, three cases with and without the loss functions are studied. All hyperparameters were kept unchanged except the loss functions. Figure 21 illustrates the results conditioned to the same well data, where samples in each line are from the same latent vector and in each column are from the

same GAN model. Table 4 shows the calculation results of the diversity metric and the conditionality metric.

In general, these samples are visually plausible from the real ones whether the GAN model they come from use the two loss functions. For the content loss function, it can be seen that samples in the 1st and the 3rd column from the GAN model using content loss show good conditionality. In addition, Table 4 indicates the L_c value is very small. As for samples in the 2nd and the 4th column, the pixel error is around 2 and the conditioning accuracy is 98.27%, meaning that in the total 160 pixels only 2 pixels are different from the real ones (i.e., pixels in the red boxes in Fig. 21). The result is not bad. As far as the author's concerned, this is due to the U-Net structure, which fully extracts the features of the input conditional well image. The content loss function just plays a subsidiary role. However, we believe it is necessary when a more accurate result with nearly zero error between experimental samples and model is required. For the diversity loss, samples in the 1st and the 2nd column from the GAN model

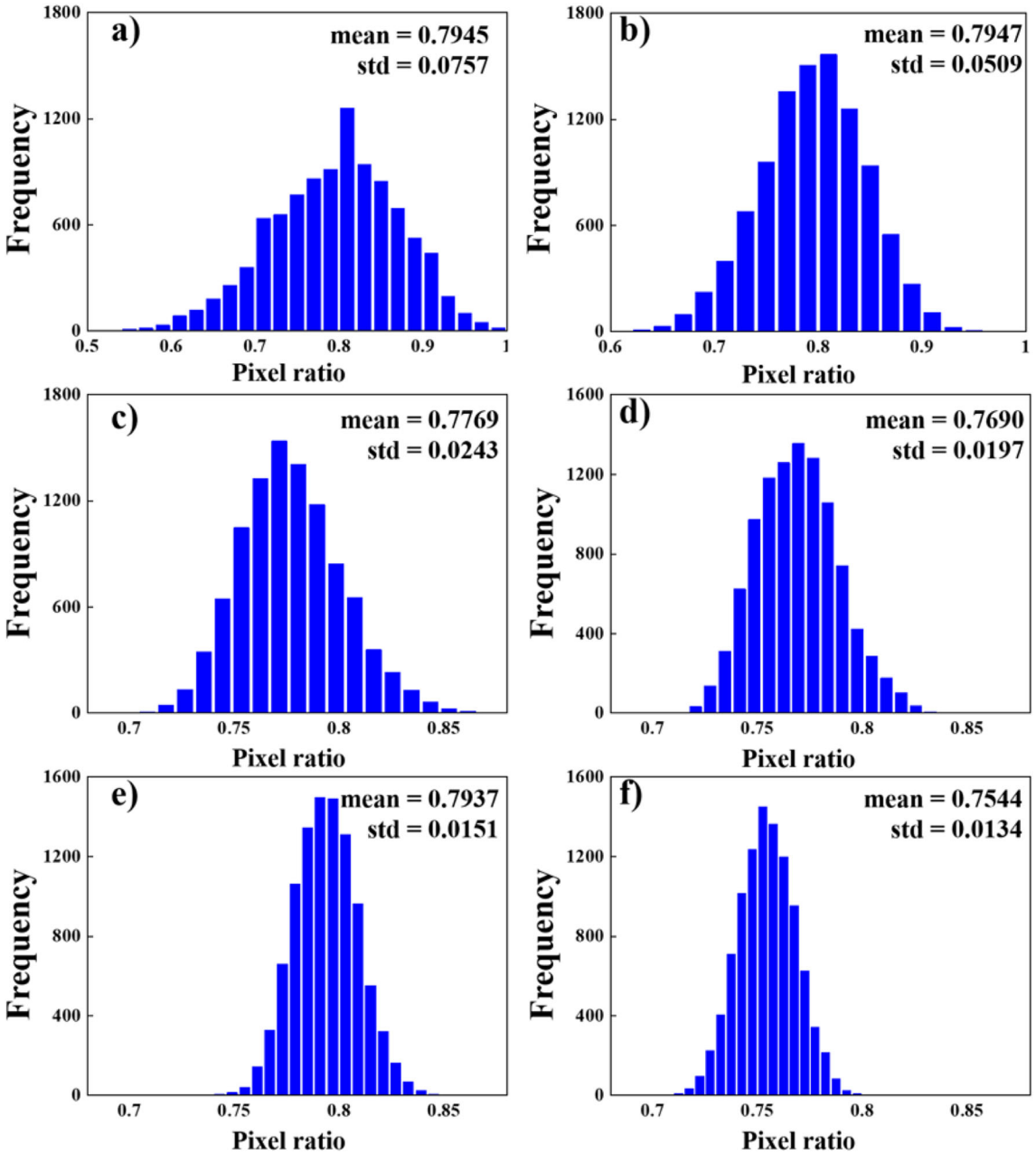

Fig. 16 Histograms of pixel ratios for 10,000 samples from **a** well-log data, **b** training dataset, **c** U-Net GAN conditioned to 1 well, **d** U-Net GAN conditioned to 2 wells, **e** U-Net GAN conditioned to 3 wells and **f** U-Net GAN conditioned to 4wells

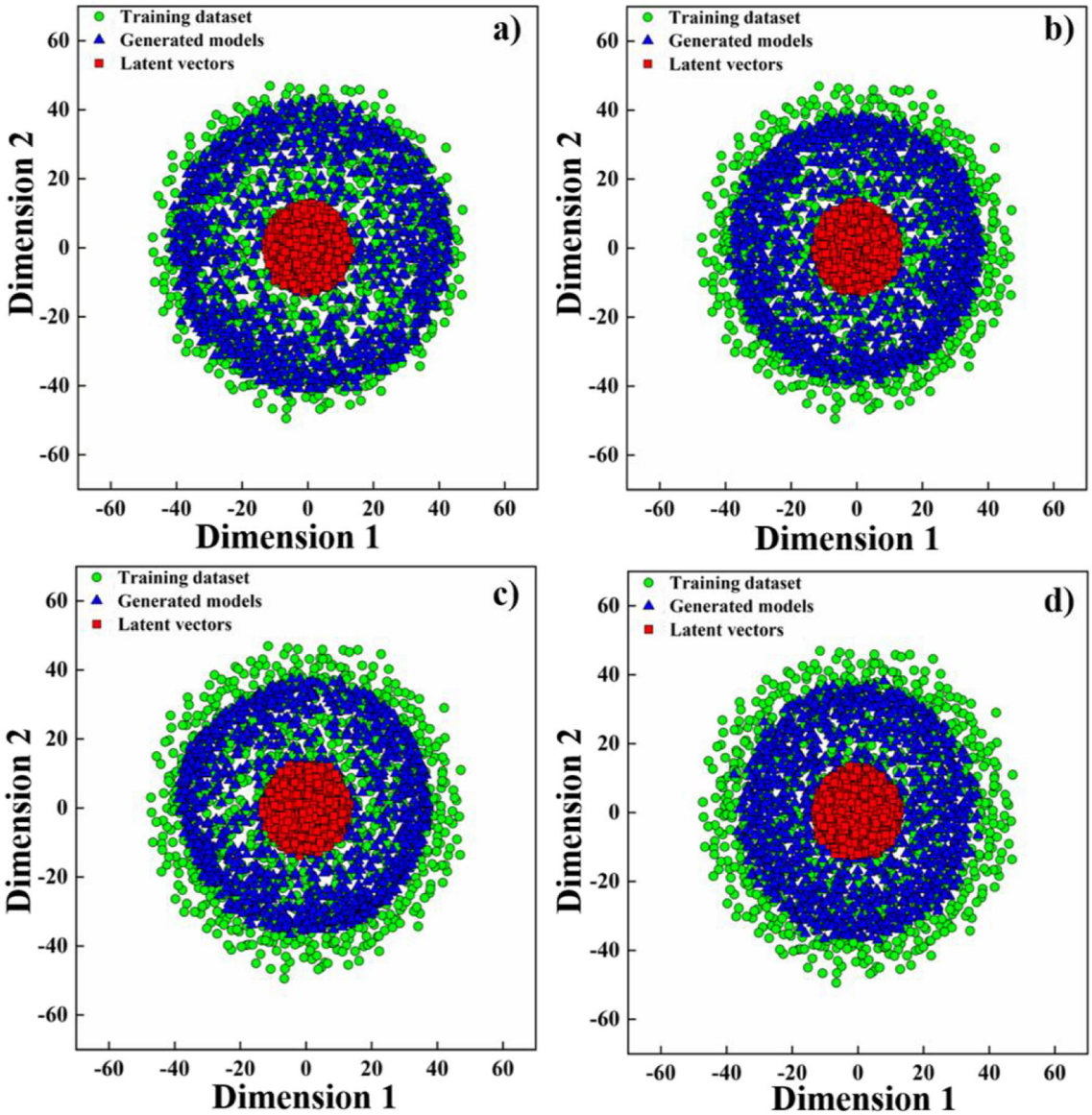

Fig. 17 MDS results from samples generated with U-Net GAN conditioned to **a** 1 well, **b** 2 wells, **c** 3 wells and **d** 4wells

using diversity loss display good diversity, which can also be proved by the MSE value and the SSIM value in Table 4. However, for samples in the 3rd and the 4th column, they look identical and calculation shows a very small MSE value and high SSIM value, suggesting these sample have bad diversity. Therefore, the diversity loss function truly plays a vital role in generating distinct samples as it is designed to.

4 Conclusions and future work

How to generate facies models conditioned to some observed data while showing high diversity is still a challenge in deep learning. In this paper, a deep learning model named Generative Adversarial Network (GAN) is used to tackle this problem. To generate unconditional geological facies, a conventional DCGAN model is trained using a facies model dataset. Then, a U-Net GAN framework is implemented for the generation of conditional facies models. Based on the conventional U-Net GAN model, several improvements are made. Noise-layers are integrated in the GAN architecture with the U-Net network, the content loss function to make the generated samples constrained to pre-exist data and the diversity loss function to generate distinct facies models are utilized. Moreover, an elaborative dataset consisting of paired facies models and conditional facies data is

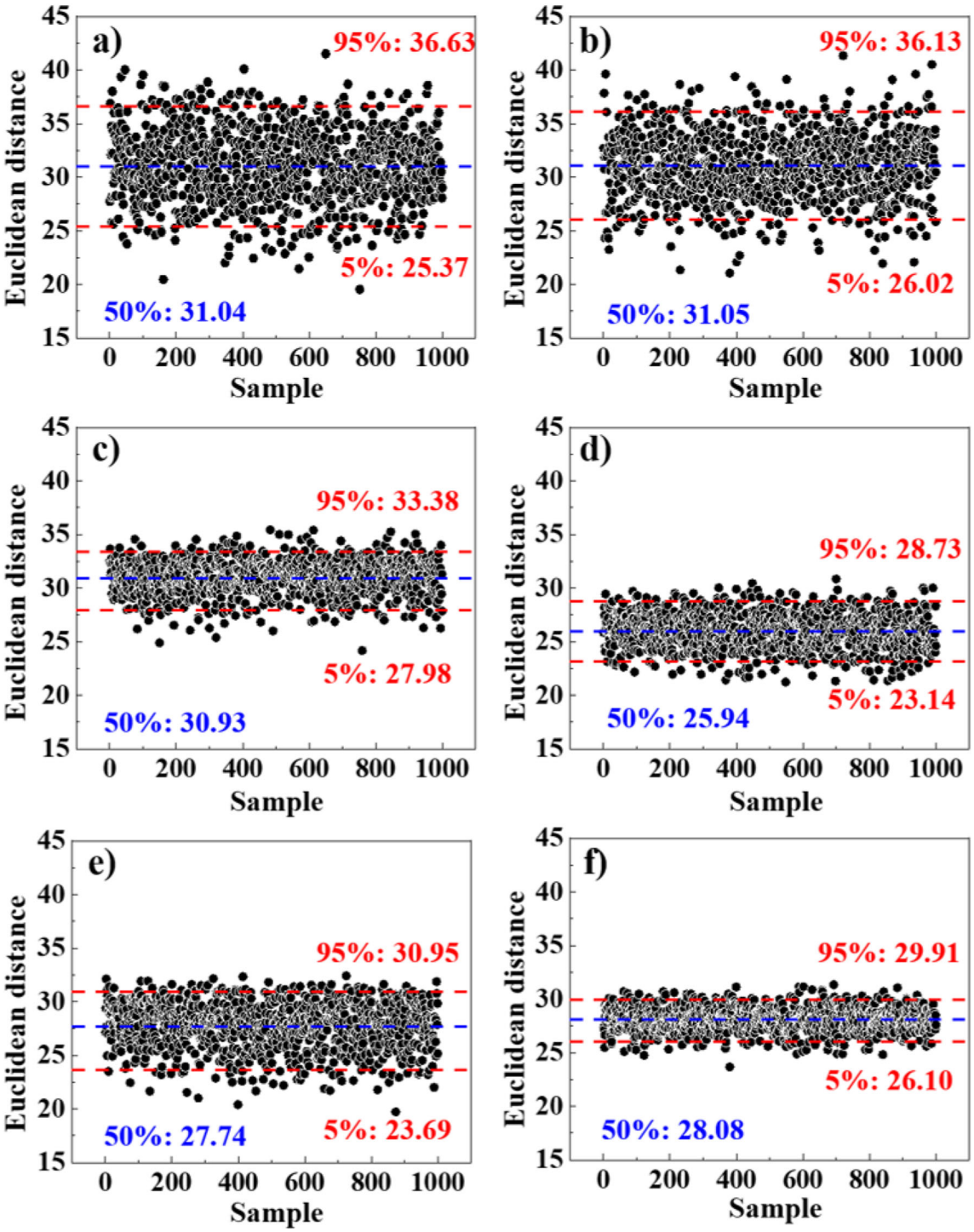

Fig. 18 Euclidean distance between the location of each model in the MDS space and the origin of this space considering the first N dimensions, which explain 75% of the original variance for: **a** the training dataset; **b** the unconditional GAN; **c** the U-Net GAN conditioned to one well, **d** the U-Net GAN conditioned to two wells, **e** the U-Net GAN conditioned to three wells, **f** the U-Net GAN conditioned to four wells. The 5%, 50% and 95% confidence intervals are shown in the figure

prepared. Results of the two generation models are visually and quantitively checked in diversity, reality and conditionality. Capability of the improved novel framework is evaluated by varying the constrained well data. Influence of GAN structure on the results is discussed by decreasing the model parameters and simplifying the convolution

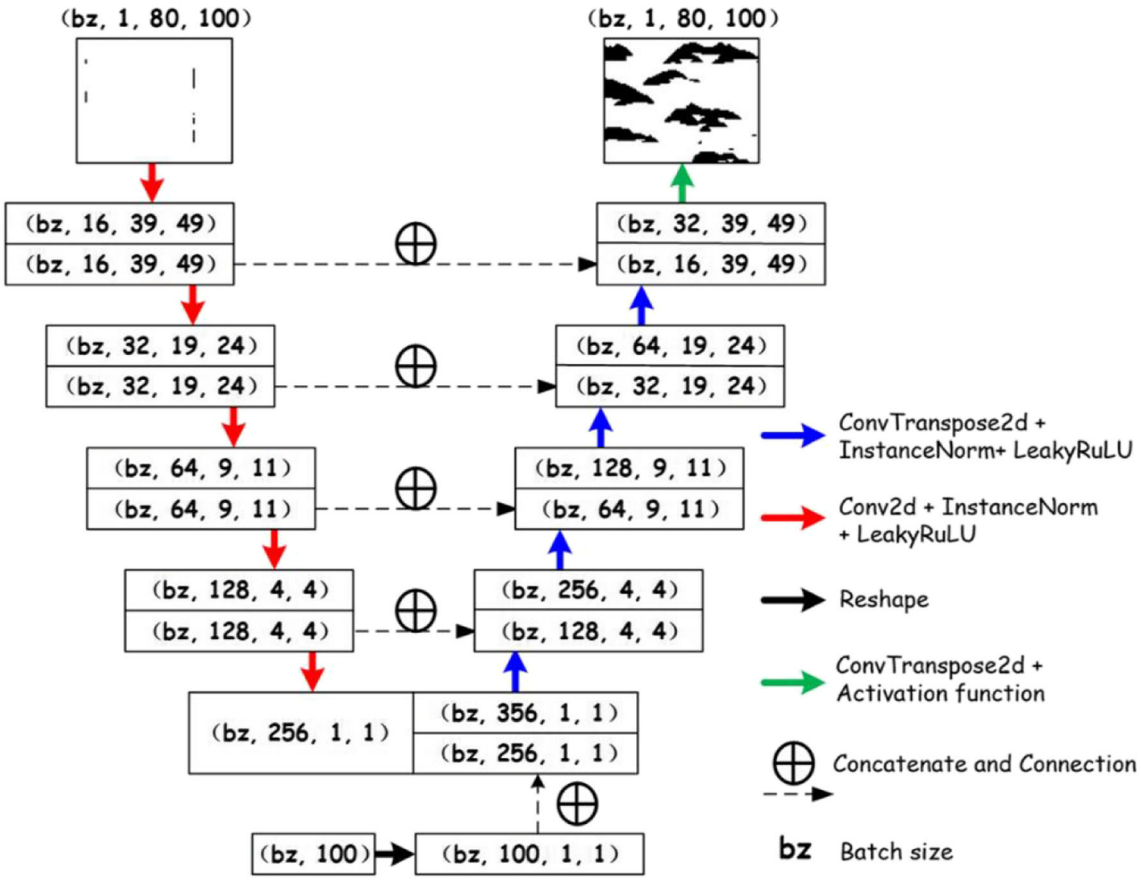

Fig. 19 Architecture of the simplified Generator

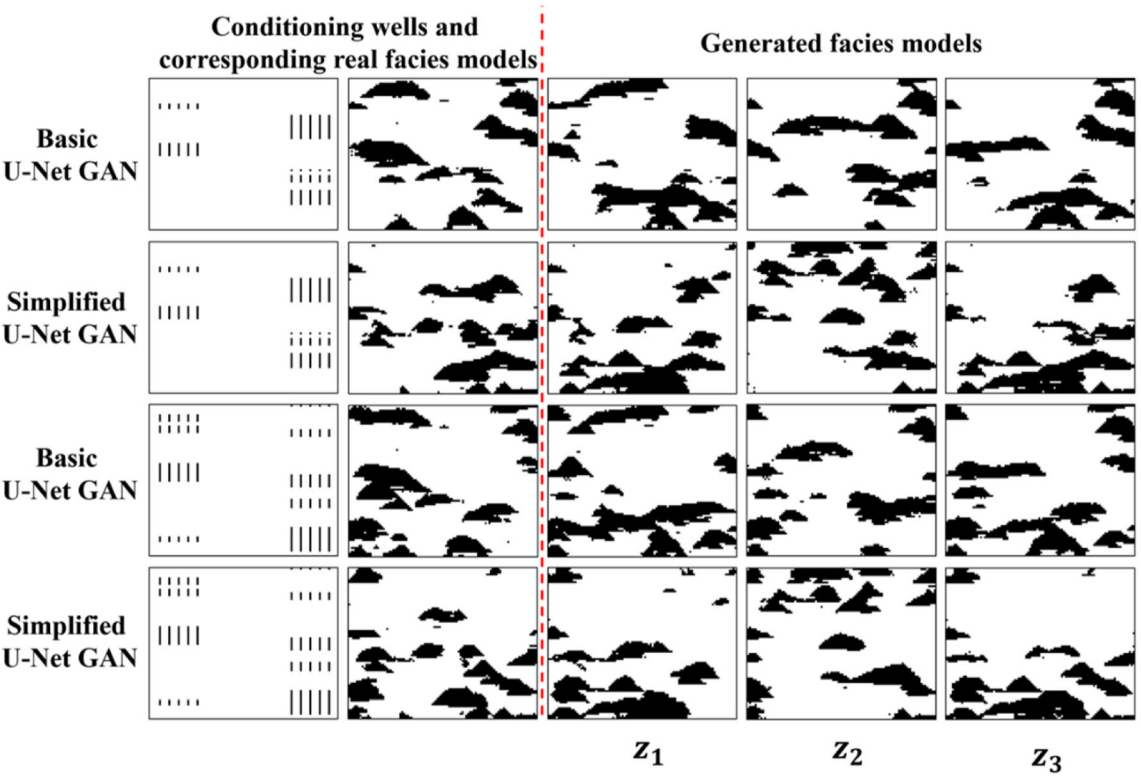

Fig. 20 Samples of the basic U-Net GAN model and the simplified U-Net GAN model

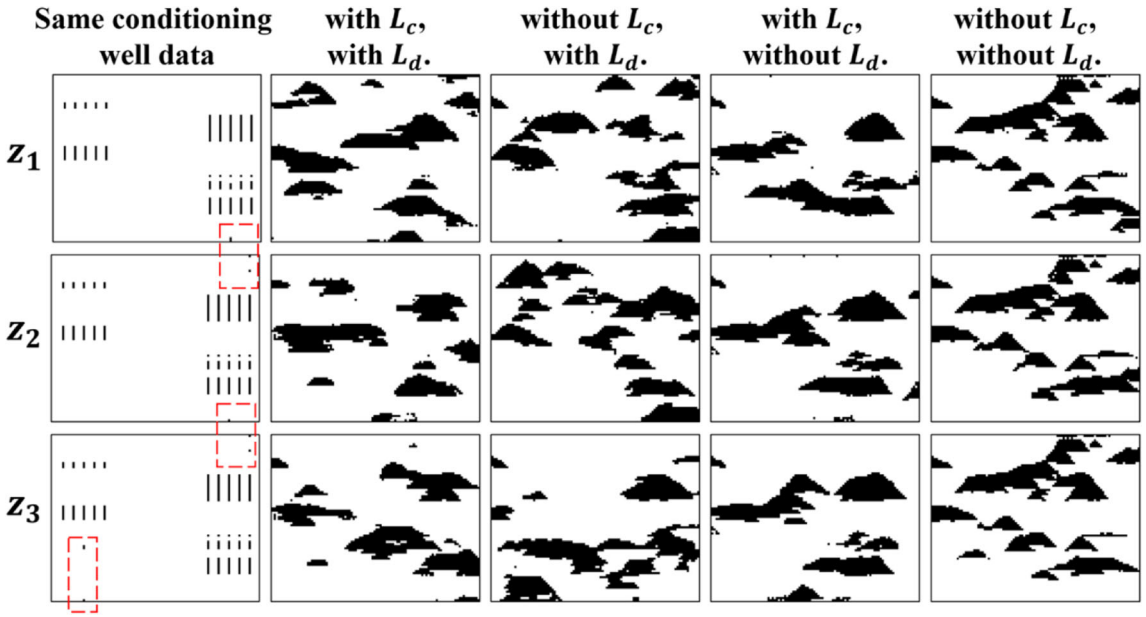

Fig. 21 Results of U-Net GAN models varying the loss functions

layers. Roles of the two loss functions in generating good results are studied. The key findings are as follows:

1) The unconditional DCGAN model learns the distribution of the facies models and is able to generate diverse and indistinguishable samples compared with the real ones;
2) The improved U-Net GAN framework is capable of producing plausible facies models in distinct patterns while honoring conditional data with high accuracy;
3) The improved conditional U-Net GAN framework is powerful and flexible. It is not only suitable for two conditional wells but for multiple constrained well data;
4) The U-Net structure is vital for generating good results. It learns the intrinsic patterns of the input conditioning well images and outputs facies models honoring the conditional data;
5) The content loss plays a subsidiary role in making the fake realizations being constrained, while the diversity loss is critical to generate samples in diverse patterns.
6) The proposed U-Net framework is reliable and robust in all conditional generation cases, which consumes very little time on hyperparameters turning.

Further studies can be focused on the extension of the U-Net GAN framework to generate 1) continuous subsurface geological models, 2) 3-D facies models and 3) to comprise more than two facies data. Additionally, this generative model could be integrated into a seismic inversion framework to generate facies. Moreover, the framework is believed to be useful not only in geostatistics, but for the scientific tasks where finite patterns of an implicit distribution and some constrained physical measurements are available.

Acknowledgements The authors gratefully acknowledge the support of the CERENA (strategic project FCT-UIDB/04028/2020) and the National Key Research and Development Project of China (project number 2019YFA0708300). The authors also thank the student exchange program between China University of Petroleum (Beijing) and Instituto Superior Técnico. We acknowledge the two anonymous reviewers that contributed to increase the quality of the original version of this work.

Compliance with ethical standards

Code availability The U-NET GAN code used in this work can be accessed using the following link: ttps://github.com/kyle4git/U-Net-GAN-for-subsurface-facies-modeling.git

Table 4 MSE value, SSIM value and content loss value of different cases with or without loss functions

Cases	MSE	SSIM	L_c	Conditioning accuracy
With L_c, with L_d	0.2238	0.4367	0.0012	99.93%
Without L_c, with L_d	0.2350	0.4188	2.7670	98.27%
With L_c, without L_d	0.0595	0.7977	0.0003	99.98%
Without L_c, without L_d	0.0951	0.7180	1.9520	98.78%

References

1. Arjovsky, M., Bottou, L.: Towards principled methods for training generative adversarial networks. arXiv preprint arXiv:1701.04862 (2017)
2. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein gan. arXiv preprint arXiv:1701.07875 (2017)
3. Arpat, G.B., Caers, J.: Conditional simulation with patterns. Math. Geol. **39**(2), 177–203 (2007)
4. Audebert, N., Le Saux, B., Lefèvre, S.: Generative adversarial networks for realistic synthesis of hyperspectral samples. In: IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium 2018, pp. 4359-4362. IEEE
5. Azevedo, L., Nunes, R., Correia, P., Soares, A., Guerreiro, L., Neto, G.S.: Multidimensional scaling for the evaluation of a geostatistical seismic elastic inversion methodology. Geophysics. **79**(1), M1–M10 (2014)
6. Azevedo, L., Paneiro, G., Santos, A., Soares, A.: Generative adversarial network as a stochastic subsurface model reconstruction. Comput. Geosci. **24**(4), 1673–1692 (2020)
7. Caterini, A., Chang, D.E.: A Novel Representation of Neural Networks. arXiv e-prints (2016)
8. Caterini, A.L., Chang, D.E.: A geometric framework for convolutional neural networks. arXiv preprint arXiv:1608.04374 (2016)
9. Chan, S., Elsheikh, A.H.: Parametric generation of conditional geological realizations using generative neural networks. Comput. Geosci. **23**(5), 925–952 (2019)
10. Cox, T.F., Cox, M.A.A.: Multidimensional scaling. Chapman & Hall/CRC, Boca Raton (2001)
11. Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., Bharath, A.A.: Generative adversarial networks: an overview. IEEE Signal Process. Mag. **35**(1), 53–65 (2018)
12. Daly, C., Caers, J.: Multi-point geostatistics–an introductory overview. First Break **28**(9), 39–47 (2010)
13. Deutsch, C.V., Journel, A.G.: Geostatistical software library and user's guide. New York 119(147) (1992)
14. Donahue, C., McAuley, J., Puckette, M.: Synthesizing audio with generative adversarial networks. arXiv preprint arXiv:1802.04208 1 (2018)
15. Drozdzal, M., Vorontsov, E., Chartrand, G., Kadoury, S., Pal, C.: The importance of skip connections in biomedical image segmentation. In: Carneiro G, et al. (eds.) Deep Learning and Data Labeling for Medical Applications. DLMIA 2016, LABELS 2016. Lecture Notes in Computer Science, vol 10008. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46976-8_19
16. Dupont, E., Zhang, T., Tilke, P., Liang, L., Bailey, W.: Generating realistic geology conditioned on physical measurements with generative adversarial networks. arXiv preprint arXiv:1802.03065 (2018)
17. Goodfellow, I.: NIPS 2016 tutorial: Generative adversarial networks. arXiv preprint arXiv:1701.00160 (2016)
18. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in neural information processing systems 2014, pp. 2672-2680
19. Hong, Y., Hwang, U., Yoo, J., Yoon, S.: How generative adversarial networks and their variants work: an overview. ACM Comput. Surv. (CSUR). **52**(1), 1–43 (2019)
20. Hore, A., Ziou, D.: Image quality metrics: PSNR vs. SSIM. In: 2010 20th international conference on pattern recognition 2010, pp. 2366-2369. IEEE
21. Isola, P., Zhu, J.-Y., Zhou, T., Efros, A.A.: Image-to-Image Translation with Conditional Adversarial Networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition 2017, pp. 1125-1134
22. Jetchev, N., Bergmann, U., Vollgraf, R.: Texture synthesis with spatial generative adversarial networks. arXiv preprint arXiv:1611.08207 (2016)
23. Karras, T., Laine, S., Aila, T.: A Style-Based Generator Architecture for Generative Adversarial Networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition 2019, pp. 4401-4410
24. Laloy, E., Hérault, R., Jacques, D., Linde, N.: Training-image based geostatistical inversion using a spatial generative adversarial neural network. Water Resour. Res. **54**(1), 381–406 (2018)
25. Laloy, E., Linde, N., Ruffino, C., Hérault, R., Gasso, G., Jacques, D.: Gradient-based deterministic inversion of geophysical data with generative adversarial networks: is it feasible? Comput. Geosci. **133**, 104333 (2019)
26. Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z.: Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. In: Proceedings of the IEEE conference on computer vision and pattern recognition 2017, pp. 4681-4690
27. Mao, Q., Lee, H.-Y., Tseng, H.-Y., Ma, S., Yang, M.-H.: Mode Seeking Generative Adversarial Networks for Diverse Image Synthesis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2019, pp. 1429-1437
28. Mariethoz, G., Caers, J.: Multiple-Point Geostatistics: Stochastic Modeling with Training Images. John Wiley & Sons, Hoboken (2014)
29. Mariethoz, G., Renard, P., Straubhaar, J.: The direct sampling method to perform multiple-point geostatistical simulations. Water Resources Research 46(11) (2010)
30. Matheron, G., Beucher, H., De Fouquet, C., Galli, A., Guerillot, D., Ravenne, C.: Conditional simulation of the geometry of fluvio-deltaic reservoirs. In: Spe annual technical conference and exhibition 1987. Society of Petroleum Engineers
31. Metz, L., Poole, B., Pfau, D., Sohl-Dickstein, J.: Unrolled generative adversarial networks. arXiv preprint arXiv:1611.02163 (2016)
32. Mirza, M., Osindero, S.: Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784 (2014)
33. Mosser, L., Dubrule, O., Blunt, M.J.: Reconstruction of three-dimensional porous media using generative adversarial neural networks. Phys. Rev. E. **96**(4), 043309 (2017)
34. Mosser, L., Dubrule, O., Blunt, M.J.: Stochastic reconstruction of an oolitic limestone by generative adversarial networks. Transp. Porous Media. **125**(1), 81–103 (2018)
35. Mosser, L., Dubrule, O., Blunt, M.J.: Conditioning of three-dimensional generative adversarial networks for pore and reservoir-scale models. arXiv preprint arXiv:1802.05622 (2018)
36. Pyrcz, M.J., Deutsch, C.V.: Geostatistical reservoir modeling. Oxford university press, Oxford (2014)
37. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434 (2015)
38. Ronneberger, O., Fischer, P., Brox, T.: U-net: convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention 2015, pp. 234-241. Springer
39. Sarkar, B.C.: Geostatistics in groundwater Modelling. In: Sikdar, P.K. (ed.) Groundwater Development and Management: Issues and Challenges in South Asia, pp. 147–169. Springer International Publishing, Cham (2019)
40. Song, S., Mukerji, T., Hou, J.: GANSim: Conditional Facies Simulation Using an Improved Progressive Growing of Generative Adversarial Networks (GANs). EarthArXiv. https://doi.org/10.31223/osf.io/fm24b (2020)

41. Strebelle, S.: Conditional simulation of complex geological structures using multiple-point statistics. Math. Geol. **34**(1), 1–21 (2002)
42. Sun, A.Y.: Discovering state-parameter mappings in subsurface models using generative adversarial networks. Geophys. Res. Lett. **45**(20), 11,137–111,146 (2018)
43. Wackernagel, H.: Multivariate Geostatistics: an Introduction with Applications. Springer Science & Business Media (2013)
44. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. **13**(4), 600–612 (2004)
45. Yang, D., Hong, S., Jang, Y., Zhao, T., Lee, H.: Diversity-sensitive conditional generative adversarial networks. arXiv preprint arXiv: 1901.09024 (2019)
46. Yeh, R.A., Chen, C., Yian Lim, T., Schwing, A.G., Hasegawa-Johnson, M., Do, M.N.: Semantic Image Inpainting with Deep Generative Models. In: Proceedings of the IEEE conference on computer vision and pattern recognition 2017, pp. 5485-5493
47. Zhang, T.-F., Tilke, P., Dupont, E., Zhu, L.-C., Liang, L., Bailey, W.: Generating geologically realistic 3D reservoir facies models using deep learning of sedimentary architecture with generative adversarial networks. Pet. Sci. **16**(3), 541–549 (2019)
48. Zhong, Z., Sun, A.Y., Jeong, H.: Predicting co2 plume migration in heterogeneous formations using conditional deep convolutional generative adversarial network. Water Resour. Res. **55**(7), 5830–5851 (2019)
49. Zhu, J.-Y., Park, T., Isola, P., Efros, A.A.: Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. Proceedings of the IEEE international conference on computer vision, In (2017) pp. 2223-2232
50. Zhu, L., Chen, Y., Ghamisi, P., Benediktsson, J.A.: Generative adversarial networks for hyperspectral image classification. IEEE Trans. Geosci. Remote Sens. **56**(9), 5046–5063 (2018)