Math Geosci
https://doi.org/10.1007/s11004-021-09934-0

GANSim: Conditional Facies Simulation Using an Improved Progressive Growing of Generative Adversarial Networks (GANs)

Suihong Song[1,2,3] · Tapan Mukerji[3] · Jiagen Hou[1,2]

Received: 29 June 2020 / Accepted: 8 February 2021

Abstract Conditional facies modeling combines geological spatial patterns with different types of observed data, to build earth models for predictions of subsurface resources. Recently, researchers have used generative adversarial networks (GANs) for conditional facies modeling, where an unconditional GAN is first trained to learn the geological patterns using the original GAN's loss function, then appropriate latent vectors are searched to generate facies models that are consistent with the observed conditioning data. A problem with this approach is that the time-consuming search process needs to be conducted for every new conditioning data. As an alternative, we improve GANs for conditional facies simulation (called GANSim) by introducing an extra condition-based loss function and adjusting the architecture of the generator to take the conditioning data as inputs, based on progressive growing of GANs. The condition-based loss function is defined as the inconsistency between the input conditioning value and the corresponding characteristics exhibited by the output facies model, and forces the generator to learn the ability of being consistent with the input conditioning data, together with the learning of geological patterns. Our input conditioning factors include global features (e.g., the mud facies proportion) alone, local features such as sparse well facies data alone, and joint combination of global features and well facies data. After training, we evaluate both the quality of generated facies models and the conditioning ability of the generators,

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s11004-021-09934-0.

✉ Jiagen Hou
jghou63@hotmail.com

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

[2] College of Geoscience, China University of Petroleum (Beijing), Beijing 102249, China

[3] Stanford University, 367 Panama St, Stanford, CA 94305, USA

Published online: 29 March 2021

by manual inspection and quantitative assessment. The trained generators are quite robust in generating high-quality facies models conditioned to various types of input conditioning information.

Keywords Conditional facies modeling · Generative adversarial networks (GANs) · Progressive growing of GANs · Deep learning · Geological pattern · Reservoir forecast

1 Introduction

Geological facies modeling is fundamental to the accurate prediction of subsurface resources, such as groundwater, petroleum, and carbon storage potential. Many geostatistical facies modeling approaches have been developed in the past decades, such as variogram-based methods, multiple point-statistics (MPS)-based methods, object-based methods, and process-mimicking methods (Pyrcz and Deutsch 2014). These approaches have various advantages and disadvantages, and they have been widely used in different scenarios. Some of them are still under research, such as the recent development of the tree-based direct sampling MPS method (Zuo et al. 2020).

Basically, geological facies modeling is a process of generating two-dimensional or three-dimensional spatial facies models with realistic geological spatial patterns, given various types of observed data. From the perspective of deep learning, geological facies modeling belongs to the class of generative problems, in which a generative model is trained to reproduce a probability distribution given many samples from that distribution (Goodfellow and Yoshua Bengio 2016). Some widely used deep generative models include deep sigmoid belief networks (Gan et al. 2015), pixel recurrent neural networks (RNN) and pixel convolutional neural networks (CNN) (Van Den Oord et al. 2016b), variational autoencoders (VAE) (Larochelle and Murray 2011; Rezende et al. 2014), and especially, generative adversarial networks (GANs) (Goodfellow et al. 2014).

Among these generative models, GANs generate very realistic results and have been the most widely studied and applied. In the GANs framework, there is a generator network and a discriminator network. The goal of the generator is to "cheat" the discriminator by generating realistic results, while the goal of the discriminator is to avoid being cheated by the generator by discriminating the real data from the outputs of the generator. Finally, after iterations of training, the generator is kept for further generative applications. Appendices A.1 and A.2 in the ESM give more details about GANs. Many variants of GANs have been developed, such as conditional GANs (Mirza and Osindero 2014), cycle GANs (Zhu et al. 2017), and bidirectional GANs (Dumoulin et al. 2016). Karras et al. (2017) proposed progressive growing of GANs, where the networks in the GANs are trained layer by layer. This progressive GAN training method allows the features to be learned from large scales to fine scales, and proves to perform much better than the conventional GAN training method in terms of the training speed, stability, and the quality of the generated results. Appendix B in the ESM shows how progressive growing of GANs is applied

for an unconditional facies modeling case. Based on the progressive growing of GANs, Karras et al. (2018) further proposed style GANs. GANs have been successfully used in many areas, including image generation (Karras et al. 2017), image inpainting (Van Den Oord et al. 2016a), super-resolution image creation (Ledig et al. 2016), text-to-image translation (Reed et al. 2016), and object segmentation (Isola et al. 2016).

Many researchers have studied the application of GANs for geological facies modeling. Mosser et al. (2017, 2018a) used deep convolutional GANs for reconstruction of three-dimensional solid-void structures of porous media and micro-CT-scale oolitic Ketton limestone. Chan and Elsheikh (2017) used convolutional GANs combined with the Wasserstein loss to generate geological facies models. These works are focused on unconditional realizations.

In most cases, geological facies models need to be conditioned to observed data (e.g., facies observed in wells). To achieve conditioning to observed data, some researchers have used "post-GAN" approaches, where unconditional GANs are first trained, and then appropriate latent vectors that generate models consistent with the observed data are searched. Nesvold and Mukerji (2019), Mosser et al. (2020), and Laloy et al. (2018) used Markov chain Monte Carlo (MCMC) algorithms to search for the appropriate latent vectors. Dupont et al. (2018) and Zhang et al. (2019) applied gradient descent method to obtain the appropriate latent vectors. With the above MCMC or the gradient decent optimization algorithm, only one appropriate latent vector is searched every time. In situations where many conditional facies realizations are required, e.g., uncertainty quantification, the latent vector searching process needs to be conducted many times, which is, however, slow and inconvenient. Therefore, Chan and Elsheikh (2019) proposed to train an extra inference neural network to map a known distribution, e.g., Gaussian, into the distribution of the appropriate latent vectors, so that multiple samples from the known distribution can be directly mapped into multiple appropriate latent vectors by the inference network. One problem for the "post-GAN" approaches is that once the values of the observed data change, the time-consuming "post-GAN" process of finding the appropriate latent vector (i.e., MCMC, gradient descent, or inference network training) needs to be performed again.

Sun (2018) applied cycle GANs for bidirectional domain transformation between high-dimensional parameter space and the corresponding model state space. The output of the GAN is directly conditioned to the GAN's input. Similarly, Mosser et al. (2018b) also used cycle GANs for domain transformation between seismic velocity and the geological model. Theoretically, cycle GANs stand out in unsupervised domain transformation tasks, where paired training datasets between two domains are difficult to obtain. One problem for cycle GANs is that concurrently training two GANs is quite difficult and unstable.

In addition, Zhong et al. (2019) used conditional GANs and "U-Net" design to transfer the permeability distribution map into CO_2 saturation maps at different time steps. The GAN takes the time step data and the permeability map as two channels in the input and generates CO_2 saturation maps as outputs. The output maps are conditioned to the input permeability and the time steps. Such a GAN architecture may also be extended for the conditional facies modeling task. Compared to the

"post-GAN" processes, this architecture is more straightforward for achieving multiple conditionings; however, the "U-Net" design in this architecture increases the number of trainable parameters, leading to increased training difficulties.

Therefore, in this paper, we improve GANs for conditional facies simulation, by introducing an extra condition-based loss function and adjusting the architecture of the generator to take conditioning data as inputs, in the context of the progressive growing of GANs. This improved GAN is called GANSim in this paper. The conditioning information for the facies modeling includes prior global features (e.g., the facies proportions, and the sinuosity of channels) alone, sparse well facies data ("hard data") alone, and the joint combination of global features and well facies data. After training, the generator can be directly used for practical conditional facies modeling without further training or "post-GAN" processes.

This paper is organized as follows. Section 2 shows GANSim and how it is used for facies modeling conditioned to global features, well facies data, and their combination. Section 3 illustrates how the trained generators are evaluated in terms of the quality of the generated facies models and the conditioning ability of the trained generators to various types of input conditioning data. Section 4 shows how necessary datasets are built for the training and testing of our GANs. Section 5 presents the results, evaluation, and analyses of the trained generators. Finally, conclusions are provided in Sect. 6.

2 GANSim

For conditional facies modeling, the generator needs two types of abilities: one is to be consistent with the geological patterns, and the other is to be consistent with the conditioning data. In GAN-based unconditional facies modeling, the generator learns the knowledge about geological patterns, and this allows the generator to simulate realistic facies models in an unconditional manner. To enforce the generator to be consistent with the given conditioning data (or conditioning ability) at the same time, we propose an improved GAN workflow called GANSim (Fig. 1) as follows. First, we design the architecture of the generator (G_θ) to also take the given conditioning information (con_0) as an input together with the latent vector (z). Second, we construct a function ($f_{\mathrm{con}}(F)$), which maps the facies model (F) into the given conditioning domain (con). Third, we use $f_{\mathrm{con}}(F)$ to map the output facies model of the generator back into the conditioning domain, i.e., $\mathrm{con}_1 = f_{\mathrm{con}}\left[G_\theta\left(z, \mathrm{con}_0\right)\right]$, where in general con_1 may not be equal to con_0; we define a condition-based loss function as some form of the distance between con_1 and con_0 (Eq. (1)).

$$L\left(G_\theta\right)_{\mathrm{con}} = \mathrm{Dist}\left(\mathrm{con}_1, \mathrm{con}_0\right) = \mathbb{E}_{z\sim p_z, \mathrm{con}_0\sim p_{\mathrm{con}_0}} \mathrm{Dist}\left(f_{\mathrm{con}}\left[G_\theta\left(z, \mathrm{con}_0\right)\right], \mathrm{con}_0\right). \quad (1)$$

Here, $L\left(G_\theta\right)_{\mathrm{con}}$ is the condition-based loss function, and $\mathrm{Dist}\left(\mathrm{con}_1, \mathrm{con}_0\right)$ is some type of distance (made more specific later) between con_1 and con_0, while p_z is the distribution of z, and p_{con_0} is the distribution of con_0. The condition-based loss function is combined with the original GAN loss function as shown in Eq. (2)

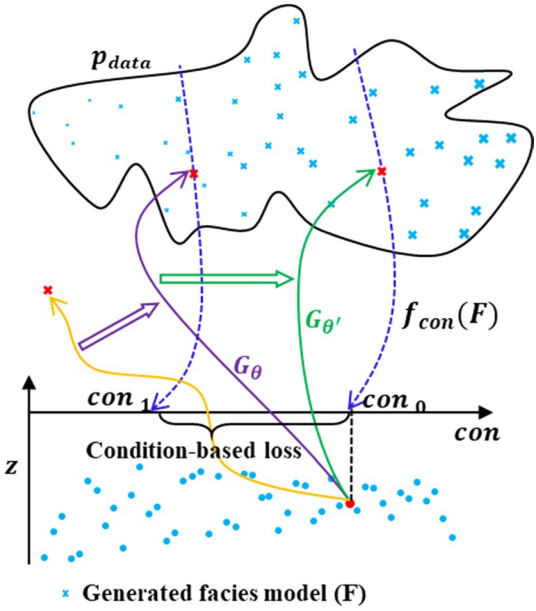

Fig. 1 A schematic for the general GANSim workflow to achieve conditional facies modeling. Axis con and z represents the condition and the latent vector, respectively. The size of the light blue cross represents certain characteristics (e.g., the width of channel) exhibited by the generated facies model, and these characteristics correspond to the input conditioning value

$$L\left(G_\theta, D_\varphi\right)_{\text{combined}} = L\left(G_\theta, D_\varphi\right) + \beta L\left(G_\theta\right)_{\text{con}}, \tag{2}$$

where $L\left(G_\theta, D_\varphi\right)_{\text{combined}}$ is the combined loss, $L\left(G_\theta, D_\varphi\right)$ is the original GAN loss function, D_φ is the discriminator, and β is the weight for $L\left(G_\theta\right)_{\text{con}}$.

Finally, we apply this combined loss to train the GAN in a progressive growing manner. The condition-based loss function only affects the training of the generator. This workflow is universal for all forms of conditioning, so it is called the general GANSim workflow hereafter in this paper.

In this general GANSim workflow, there are two "objectives" working: (1) the GAN framework and the original GAN loss function push the generator to map its input into the distribution of the training dataset (p_{data}), so the output facies model of the generator would be realistic, i.e., $G_\theta\left(z, \text{con}_0\right) \rightarrow p_{\text{data}}$ (shown by purple arrows in Fig. 1); (2) the condition-based loss function pushes the generator to the proper subspace of the distribution that is consistent with the input conditioning value through $f_{\text{con}}(F)$, i.e., $f_{\text{con}}\left[G_\theta\left(z, \text{con}_0\right)\right] \rightarrow \text{con}_0$ (shown by green arrows in Fig. 1). With the above two objectives, the output facies model of the generator is both realistic in terms of spatial patterns and consistent with the input conditioning data.

Three most important elements in the general GANSim workflow are (1) the architecture of generator to take the conditioning data as input, (2) the construction of $f_{\text{con}}(F)$, and (3) the definition of the distance between con_1 and con_0 in the condition-based loss function. These elements are decided depending on the input conditioning data type. In the following parts, we discuss these three elements in detail, with conditioning data as non-spatial global features, spatially sparse well facies data, and both jointly.

The GAN architecture for conditional facies modeling in this paper is based on an unconditional GAN for facies modeling (Song et al. 2020) which is also described in detail in Appendix B in the ESM. The generator and the discriminator in the unconditional GAN are called the base generator and the base discriminator in this paper. For conditioning, the generators are modified from the base generator; the discriminator is modified from the base discriminator for the case of conditioning to global features alone and the case of conditioning to both global features and well facies data, and remains the same as the base discriminator for the case of conditioning to well facies data alone. The resolution of the base generator's output is 64×64 (two-dimensional). The Wasserstein loss function with gradient penalty (W-gp) [Eq. (A4) in Appendix A in the ESM] and other settings of training in the unconditional work (Appendix B.2 in the ESM) are also used in this work. After training, the generator is kept for further evaluation and practical applications.

2.1 Facies Modeling Conditioned to Global Features

In practice, sometimes we need to simulate facies models that have certain types of features, such as the proportion of facies and the sinuosity of channels. These features describe the global characteristics of the facies models and are not related to the spatial distribution of facies, so these features are called global features (g) in this paper.

According to the general GANSim workflow, we specify the three elements for the facies modeling conditioned to global features as follows. First, we modify the input layer of the base generator to also include the global features, and accordingly adjust the first fully connected layer of the base generator (Fig. 2). Second, the facies model-to-condition function (specifically called the facies model-to-global features function in this case, $f_g(F)$) can be easily obtained for a small number of global features, such as facies proportion, but can be difficult to calculate for other global features, such as the sinuosity, orientation, width, wavelength, and amplitude of

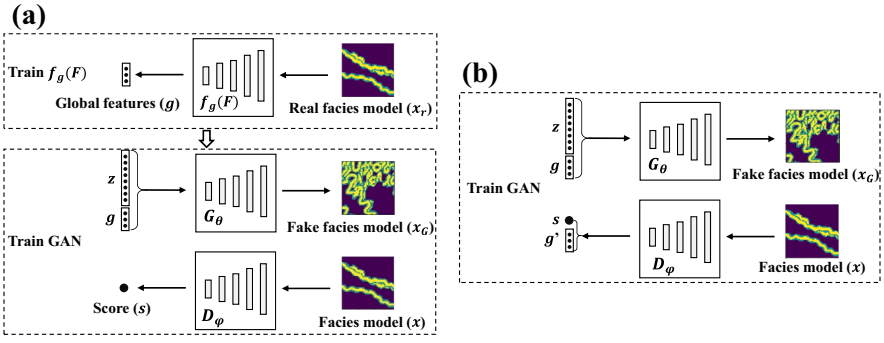

Fig. 2 **a** Two-step procedure of the facies modeling conditioned to global features. First, we train $f_g(F)$ with training facies models and corresponding global features. Second, based on the trained $f_g(F)$, we train the modified GANs. **b** The discriminator is modified from the base discriminator to integrate the $f_g(F)$ network, and it produces a score value (s) and the global feature values. In this way, we only need to train the modified GAN

channel complexes, as it could involve some image processing on every generated facies model. For example, Clerici and Perego (2016) proposed to first obtain the centerline of a channel by gradually moving two channel boundary curves towards each other, then calculate the width of the channel by averaging many transect lines that are orthogonal to the centerline, and finally calculate the sinuosity index by dividing the length of the centerline by the distance between the start and end points of the centerline. However, such calculations are difficult to express using parameterized functions, and would be specific to each global feature. An efficient and more general way to obtain $f_g(F)$ (valid for any global feature) is to train a separate deep neural network with a labeled training dataset, where the input is the facies model and the outputs are the global features. Considering that the architecture and function of such a deep neural network are very similar to that of the discriminator, we propose to integrate $f_g(F)$ into the discriminator so that the discriminator produces a score value (s) and an array of the global feature values (Fig. 2). Third, the distance between the input and output global features in the condition-based loss function (Eq. (1)) is defined as the L2 norm distance.

As the Wasserstein loss function with gradient penalty (W-gp) is used, we combine Eq. (A4) in the ESM and Eqs. (1) and (2) to derive the final loss function of the modified generator in this case as follows

$$L\left(G_\theta\right)_{\text{combined}} = \mathbb{E}_{z\sim p_z, g\sim p_g}\left\{-D_{s_\varphi}\left(G_\theta(z,g)\right) + \beta\left\|D_{g_\varphi}\left[G_\theta(z,g)\right] - g\right\|_2\right\}, \quad (3)$$

where p_z and p_g are the distributions of the latent vector (z) and the global features (g), and D_{s_φ} and D_{g_φ} represent the output score (s) and the output global features of the modified discriminator. In terms of the loss function for the modified discriminator, loss in Eq. (A4) in the ESM can only be used to train the modified discriminator to produce a meaningful score to assess the realism of the input facies model, but cannot train the modified discriminator to produce meaningful global features of the input facies model. Thus, we add an additional term to the loss of Eq. (A4) in the ESM [$-\gamma \parallel D_{g_\varphi}(x) - g \parallel_2$ in Eq. (4)] to train the modified discriminator to extract meaningful global features of the input facies model in a supervised way, using the training facies models and the corresponding ground truth global features; the final loss function for the modified discriminator is given in the following equation

$$\begin{aligned} L\left(D_\varphi\right)_{\text{combined}} = \mathbb{E}_{(x,g)\sim p_{(\text{data},g)}, z\sim p_z}\left[-\gamma \parallel D_{g_\varphi}(x) - g \parallel_2 - D_{s_\varphi}\left[G_\theta(z,g)\right] + D_{s_\varphi}(x)\right] \\ - \lambda \mathbb{E}_{\hat{x}\sim p_{\hat{x}}}[(\parallel \nabla_{\hat{x}} D_{s_\varphi}(\hat{x}) \parallel_2 - 1)^2], \end{aligned} \quad (4)$$

where (x, g) is a pair of training facies models and the corresponding global features, $p_{(\text{data},g)}$ is their joint distribution, γ is a weight, and $\hat{x}$ is sampled between $x \sim p_{\text{data}}$ and $x_G = G_\theta(z, g)$, i.e., $\hat{x} = tx + (1-t)x_G, t \sim \text{uniform}(0, 1)$.

The loss function $L\left(G_\theta\right)_{\text{combined}}$ is minimized when training the modified generator, while the loss function $L\left(D_\varphi\right)_{\text{combined}}$ is maximized when training the modified

discriminator. In our study, we train GANs in a progressive growing process for better performance, but it can also be trained in a conventional process.

2.2 Facies Modeling Conditioned to Well Facies Data

Well facies data have very high certainty and resolution, but they are sparsely distributed around the whole study area. One approach for feeding in the well facies data into the generator is the "U-Net" design (e.g., Ledig et al. 2016; Zhong et al. 2019), where the spatial well facies data are first coded into a low-dimensional space and then encoded back into the high-dimensional facies models. Inspired by the progressive growing of GANs, we propose a simpler encoding approach for feeding in the well facies data (Fig. 3). Let N be the number of different facies categories. The input sparse well facies data (w, 64×64) are decomposed into multiple channels: one well location indicator channel ($I_{w\text{loc}}$, $64 \times 64 \times 1$), and $N-1$ well facies indicator channels one for each of the $N-1$ facies types (I_{w1}, I_{w2}, …, I_{wN-1}, $64 \times 64 \times 1$), i.e., $w \rightarrow (I_{w\text{loc}}, I_{w1}, I_{w2}, \ldots, I_{wN-1})$. The indicator of the last facies type I_{wN} is not included, because the information of I_{wN} is included by the other indicators. In progressive growing of GANs, the real samples are fed in at multiple scales from coarse to fine (Karras et al. 2017). Thus, these well indicator channels ($64 \times 64 \times N$) are downsampled into different resolution levels ($4 \times 4 \times N$, $8 \times 8 \times N$, …, $32 \times 32 \times N$). The well location indicator channel is downsampled using maximizing, and the well facies indicator channels are downsampled using averaging. These downsampled and the original $64 \times 64 \times N$ indicator channels are converted into feature cubes of the same resolution, using convolutional layers with kernel size of 1×1 (Fig. 3). The number of feature maps in these feature cubes should be proportional to the number of facies types (N). The 1×1 convolutional layers are commonly used when there is a need to change only the number of the feature maps, (e.g. as in

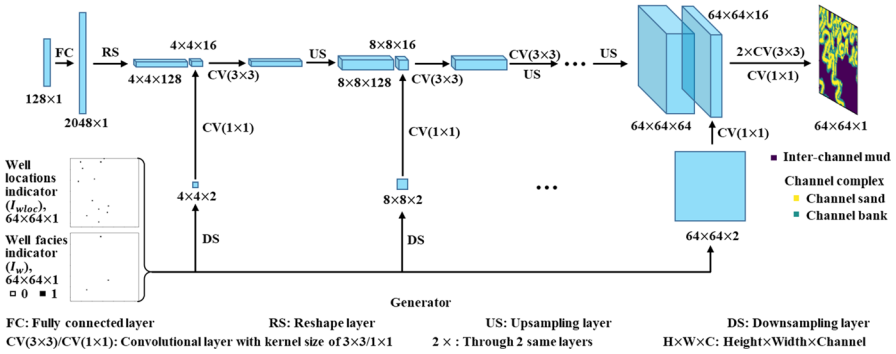

Fig. 3 The architecture of the generator for the facies modeling conditioned to the well facies data. In this figure, there are three facies: inter-channel mud, channel sand, and channel bank. We combine channel sand and channel bank facies together as one channel complex composite facies in the input well facies data, and only take the well location indicator and channel complex facies indicator as inputs. The input channel complex facies can be generated as either channel sand or channel bank in the generated facies models

the inception network, Szegedy et al. (2015)). They do not conduct spatial feature abstraction, which is especially important for well facies conditioning because the constraint of well facies data is strictly localized.

Finally, we concatenate the feature cubes obtained in the previous step with the corresponding feature cubes of the base generator. Because progressive growing is used for training, the generator first grasps the geological knowledge and the well facies conditioning ability at larger scales (or at lower resolutions) and then progressively learns them at finer scales (or at higher resolutions).

The facies model-to-condition function (specifically called the facies model-to-well facies function in this case, $f_w(F)$) is simply the process of extracting the facies indicators at the well locations from the generated facies models. Given that the progressive growing process generates facies models at various resolution levels, $f_w(F)$ first upsamples the generated facies models into 64×64 resolution scale and then extracts the facies indicators at the well locations from the upsampled facies models (Eq. (5)).

$$f_w\left(x_G\right) : I_{w\mathrm{loc}} \odot US(x_G), \tag{5}$$

where $US(x_G)$ denotes the upsampling operator that upsamples the generated facies model (x_G) into the resolution of 64×64 using nearest-neighbor upsampling method, and $\odot$ is the element-wise product.

The distance in the condition-based loss function (Eq. (1)) is defined as the L2 distance between the input-sparse well facies data (w) and the generated facies data at well locations; the well facies condition-based loss function is given as in Eq. (6)

$$L\left(G_\theta\right)_w = \mathbb{E}_{z\sim p_z, w\sim p_w}\left\| I_{w\mathrm{loc}} \odot US\left(G_\theta(z, W)\right) - w\right\|_2, \tag{6}$$

where p_w represents the distribution of possible sparse well facies data (w), and W represents the well indicators $\left(I_{w\mathrm{loc}}, I_{w1}, I_{w2}, \ldots, I_{wN-1}\right)$ which are decomposed from w.

One pitfall of the current procedure is that sometimes the generated facies type do not change smoothly from well location pixels to the surrounding pixels (e.g., (b) and (c) in Fig. 4). Such local abrupt transition of facies types around the well location will be called "local pixel noise" for brevity in this paper. The reasons for this local pixel noise might be as follows: (1) each conditioning well facies datum generally occupies only one of the 64×64 pixels in the whole simulation area; (2) the original GANs loss function enforces the global spatial patterns of the generated facies models, while the condition-based loss function enforces facies conditioning only at well point pixels (Eq. (2)); (3) the local pixel noise occurring only at the single-pixel well locations may not hurt the global spatial pattern reproduction greatly, i.e., the discriminator easily neglects this local pixel noise when obtaining the global score.

To address the local pixel noise problem, we propose to enlarge the well datum occupation area from 1×1 pixel to 4×4 pixels (e.g., from (a) to (d) in Fig. 4) in the sparse well facies data before training the GANs. In this way, the local pixel noise phenomenon would have a larger impact on the global pattern reproduction, so it would be

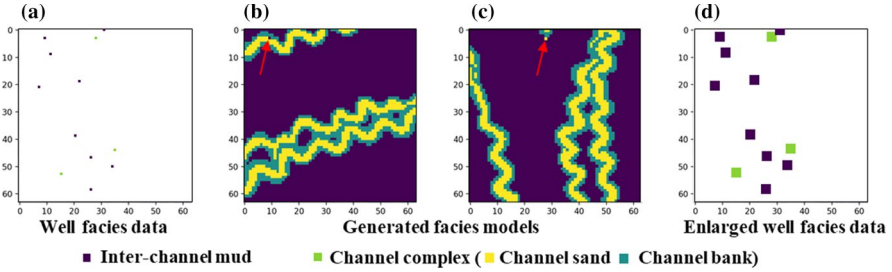

Fig. 4 **a** The original input sparse well facies data. **b–c** The generated facies models with **a** as the input condition, where the red arrows point to the local pixel noise phenomenon at the single-pixel well locations. **d** The enlarged sparse well facies data corresponding to **a**. In this figure, there are three facies: inter-channel mud, channel sand, and channel bank facies. We combine channel sand and channel bank facies together as one channel complex composite facies in the input well facies data (**a**, **d**); the input channel complex facies can be generated as either channel sand or channel bank in the generated facies models (**b**, **c**)

penalized during the training. We train GANs with both the original well facies data (before well datum enlargement) and the enlarged well facies data for facies modeling, and then compare the two trained generators.

2.3 Facies Modeling Conditioned to Both Global Features and Well Facies Data

The specifications of the three elements of the general GANSim workflow (i.e., the settings of generator architecture, the facies model-to-condition function $f_{\text{con}}(F)$, and the condition-based loss function) for conditioning to global features is distinct from that for conditioning to well facies data. Therefore, we can combine the settings in Sects. 2.1 and 2.2, and use both global features and well facies data as joint conditioning data for facies modeling. The generator takes global features and well facies data together as inputs, in the manner shown in Figs. 2(b) and 3; the architecture of discriminator is the same as the discriminator in the case of only conditioning to global features (Fig. 2(b)). The final loss function is a weighted combination of the original GAN loss function $L\left(G_\theta, D_\varphi\right)$, global features condition-based loss function $L\left(G_\theta\right)_g$, and well facies condition-based loss function $L\left(G_\theta\right)_w$, as shown in the following Eq. (7)

$$L\left(G_\theta, D_\varphi\right)_{\text{combined}} = L\left(G_\theta, D_\varphi\right) + \beta_1 L\left(G_\theta\right)_g + \beta_2 L\left(G_\theta\right)_w, \tag{7}$$

where $L\left(G_\theta, D_\varphi\right)_{\text{combined}}$ is the combined loss, and β_1 and β_2 are weights. The magnitudes of β_1 and β_2 control the ability of the generated facies models being similar to training facies models, being conditioned to input global features, and being conditioned to input well facies data during training. To better tune the magnitudes of β_1 and β_2, we normalize the three types of losses into standard Gaussian distribution, i.e., $L\left(G_\theta, D_\varphi\right)$, $L\left(G_\theta\right)_g$, and $L\left(G_\theta\right)_w$, before multiplying the weights. By combining

Eqs. (4), (6), and (7), the loss function of the modified generator in this case can be represented as in Eq. (8)

$$L\left(G_\theta\right)_{\text{combined}} = \mathbb{E}_{z\sim p_z, g\sim p_g, w\sim p_w}\Big\{-D_{s_\varphi}\left(G_\theta(z,g,W)\right) + \beta_1 \parallel D_{g_\varphi}\left[G_\theta(z,g,W)\right] - g \parallel_2 + \beta_2 \parallel I_{w\text{loc}} \odot US\left(G_\theta(z,g,W)\right) - w \parallel_2\Big\}, \tag{8}$$

where W represents well indicators $\left(I_{w\text{loc}}, I_{w1}, I_{w2}, \ldots, I_{wN-1}\right)$, which are decomposed from w. The loss function of the modified discriminator is very similar to Eq. (4), except the inputs of the generator also include W in this case

$$L\left(D_\varphi\right)_{\text{combined}} = \mathbb{E}_{(x,g)\sim p_{(\text{data},g)}, z\sim p_z, w\sim p_w}\left[-\gamma \parallel D_{g_\varphi}(x) - g \parallel_2 - D_{s_\varphi}\left[G_\theta(z,g,W)\right] + D_{s_\varphi}(x)\right] - \lambda \mathbb{E}_{\hat{x}\sim p_{\hat{x}}}[(\parallel \nabla_{\hat{x}} D_{s_\varphi}(\hat{x}) \parallel_2 - 1)^2], \tag{9}$$

where D_{s_φ} and D_{g_φ} represent the output score and output global features of the modified discriminator, respectively, and $\hat{x}$ is sampled between $x \sim p_{\text{data}}$ and $x_G = G_\theta(z,g,W)$, i.e., $\hat{x} = tx + (1-t)x_G, t \sim \text{uniform}(0,1)$. In this case, only the enlarged well facies data is used to train GANs.

3 Evaluation Metrics

The metrics assess both the quality (i.e., the realism and the diversity) of the generated facies models and the conditioning ability of the generator. We use manual inspection to evaluate the quality of the generated facies models. Manual inspection is one of the most common and intuitive ways to evaluate GANs (Borji 2018). We generate a large number of facies models and assess the generator by comparing the generated facies models with the training facies models in terms of the realism and the diversity.

Assessing the conditioning ability of the generator means checking whether the output of the generator exhibits characteristics that are consistent with the input conditioning data. We propose different metrics to assess the conditioning ability of the generator for different types of conditioning data.

(1) Global features metrics

We use both manual inspection and quantitative metrics to assess the generator's conditioning ability to global features. Manual inspection includes the following two aspects. First, manually observe the gradual change of certain characteristics exhibited by the generated facies models, when the input global feature values of the generator change gradually; this is a relative assessment of the conditioning ability, thus a weak metric. Second, manually compare certain characteristics exhibited by the generated facies models with the corresponding input global feature values. Because human eyes are not sensitive to the magnitude of values, we further replace the input global feature values with the

real facies models that correspond to the same global feature values and directly compare the generated facies models with the real facies models with respect to certain characteristics. This metric compares the generated facies models with the input global feature values, so it is a relatively strong metric.

To quantitatively assess the generator's conditioning ability, we randomly generate many facies models and directly calculate or measure the global features (e.g., the facies ratio or width of channels) from each generated facies model. We compare the calculated global feature values with the corresponding input global feature values for each generated facies model and measure their closeness. We also compare the distributions of calculated global features from the generated facies models with that from the training facies models.

(2) Well facies metrics

The assessment of the generator's conditioning ability to well facies data includes two aspects: the well facies reproduction accuracy at well points and the local pixel noise around well points. We expect the generated facies models to reproduce the input well facies types at well points, so we define the well facies reproduction accuracy as the percentage of the well facies data that are accurately reproduced in the generated facies models, for each facies type. In addition, we randomly generate many facies models and manually inspect the local pixel noise problem.

4 Dataset

We build a large systematic synthesized dataset, which includes 35,640 two-dimensional (64×64) facies models, their corresponding global features, and 285,120 sparse well facies data (64×64).

The facies models were synthesized in the commercial Petrel platform using object-based modeling. It includes three facies types: inter-channel mud, channel sand, and channel bank facies. Each facies model includes multiple channels, and these channels have similar features (e.g., orientation, sinuosity, etc.). During the synthesizing process, we tune the input number, orientation, wavelength, amplitude, and width of channel sand to create a variety of synthesized facies models. Figure 5 shows some facies model examples. These input parameters are set as the global features for the synthesized facies models. We also include two extra parameters as global features, i.e., the proportion of the inter-channel mud facies and the sinuosity index of the channel sand, which is defined as the amplitude divided by the wavelength.

Well facies data are produced from the synthesized facies models. For each facies model, eight sets of well facies data are randomly sampled, and each well facies set includes 1 to 20 well points. Each well point occupies one pixel. The channel sand and channel bank are lumped together as one channel complex composite facies in well facies data, so the final well facies types include channel complex composite facies and inter-channel mud facies (Fig. 5).

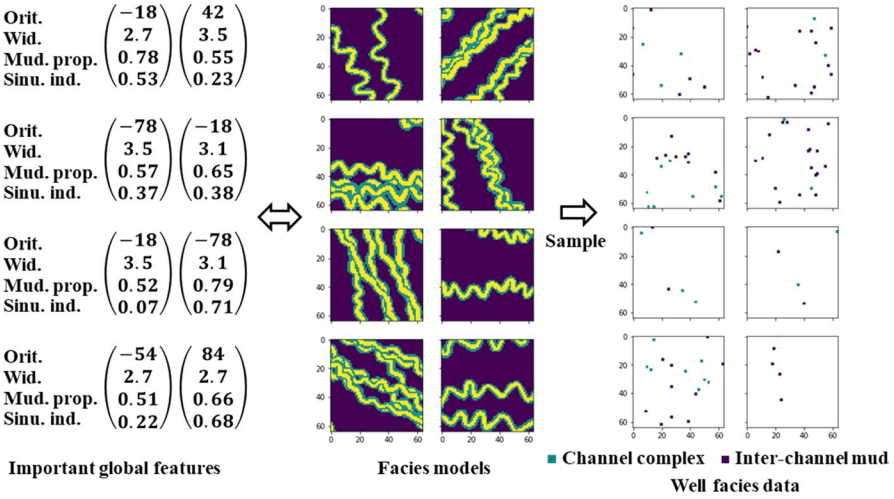

Fig. 5 Random examples of the facies models, corresponding global features, and the sparse well facies data in the synthesized dataset

We split the synthesized dataset into the training dataset and the test dataset. The training dataset include 32,640 facies models and their corresponding global features, and well facies data, while the test dataset includes the remaining 3,000 facies models and their corresponding global features, and well facies data. The training dataset was used for training the GANs, while the test dataset was used for evaluation of the trained generators.

5 Facies Modeling Results and Analyses

We use TensorFlow (tensorflow.org), an open-sourced deep learning framework, to construct and train our GANs. Two GPUs (NVIDIA Tesla V100-PCIE-32GB), ten CPUs, and 80 G of RAM are used in parallel for training the GANs conditioned to different types of inputs, as described in the following cases.

5.1 Conditioning to Global Features

In our study, we used three global features for facies modeling, namely, the inter-channel mud facies proportion, the sinuosity index of the channel sand, and the width of the channel sand. Based on the approach described in Sect. 2.1, the input of the generator is a vector of 124×1 dimensions, which include 121×1 dimensions for the latent vector and 3×1 dimensions for the three global features. The output of the modified discriminator (D_φ) is 4×1 dimensions corresponding to one score value and the three global feature values.

There are in total three predefined weights in this case (see Eqs. (3), (4)): β, γ, and λ. Weight λ is set to the default value of 10 as in the Wasserstein loss paper

(Gulrajani et al. 2017). Weights β and γ are decided based on the realism of the generated facies models and their conditioning ability to input global features, in quick trial-and-error experiments, as we scan through a range of the weight values. Since the evaluation of conditioning to input mud facies proportion is more straightforward than the other two global features, we only assessed the conditioning to mud facies proportion in the quick experiments. The experiments are based on a GANs training schedule, in which 10,000 iterations of alternative training of the discriminator and generator were conducted in the first phase, while 20,000 iterations were conducted in other phases (see Appendix B.3 in the ESM). The training is stopped after a total of 80,000 training iterations, when 64 × 64 dimensional facies models are produced. Figure 6 shows randomly generated facies models and cross-plots between input and calculated real mud facies proportion values of generated facies models, for different combinations of β and γ. The choice of weight β and γ relates to a trade-off between realistic reproduction of the spatial patterns and their conditioning ability to input mud facies proportion value. The results are acceptable when β is roughly between 1 and 10 and γ between 0.01 and 10 (the area inside the red polygon in Fig. 6). In this case, we chose the weight combination of β and γ both

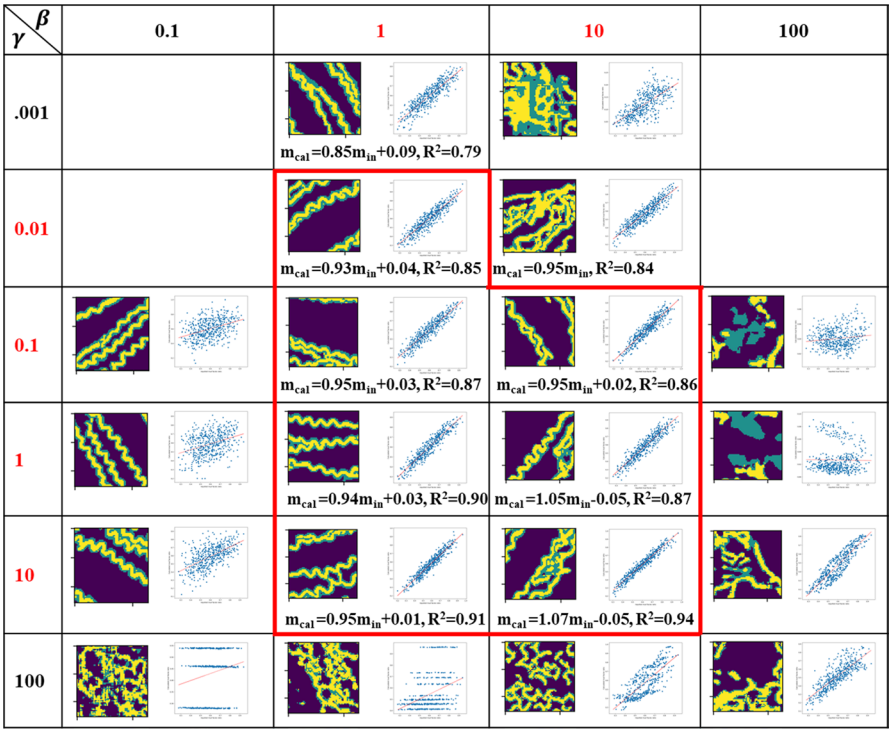

Fig. 6 Randomly generated facies models (64 × 64), cross-plots between input and real mud facies proportion values of generated facies models, and the trendline equations of the cross-plots for different weight combinations of β and γ, based on a quick training schedule. In the equations, m_{in} and m_{cal} represent input and calculated real mud facies proportion value for generated facies models, respectively

being 10, because the best conditioning ability to input mud facies proportion is achieved at this weight combination.

The formal training schedule we used here and also in the following cases includes 20,000 training iterations for phase 1 (4×4), 40,000 training iterations for each phase during phase 2 (8×8) to phase 4 (32×32), and an unlimited number of iterations for phase 5 (64×64) until stopping criterion is achieved (see Appendix B.3 in the ESM). The stopping criterion is mainly manual inspection of the realism, diversity, and conditioning ability of generated facies models. In this case, the GAN is trained for 13 h, and we kept the final generator for further assessments and practical applications. Figure 7 shows the loss of the modified discriminator (Eq. (4)) versus alternative training iterations. We used the 3,000 groups of global feature values in the test dataset and randomly sampled 3,000 latent vectors (from a Gaussian distribution) to generate 3,000 facies models for evaluation of the generator. Then, we arranged the generated facies models and the 3,000 real facies models in the test dataset, according to the magnitude of the corresponding global feature values, in Figs. 8 and 9. Compared to the facies models in the test dataset, the generated facies models are very realistic and diversified, in spite of minor flaws.

In Figs. 8 and 9, the test facies models are used as the ground truth for the generated facies models. We see that when a certain input global feature gradually changes, the corresponding characteristics exhibited in the generated facies models also gradually change; for example, in the first column of Fig. 8, the mud facies proportion in the generated facies models gradually increases, as the input inter-channel mud facies proportion value gradually increases. In addition, the generated facies models are also very similar to the corresponding ground truth test facies models, with respect to the mud facies proportion, the width, and the sinuosity of channel sand; for example, in Fig. 9, the upper left generated facies model is very similar to the upper left test facies model, with respect to these characteristics.

Figure 10 shows the cross-plot between the input mud facies proportion values of the generator and the mud facies proportion values calculated from the corresponding generated facies models. These two proportions are almost equal to each other

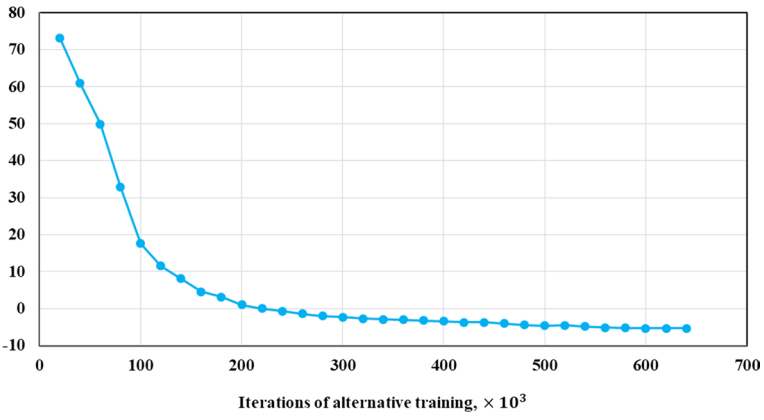

Fig. 7 The loss of the modified discriminator versus training iterations

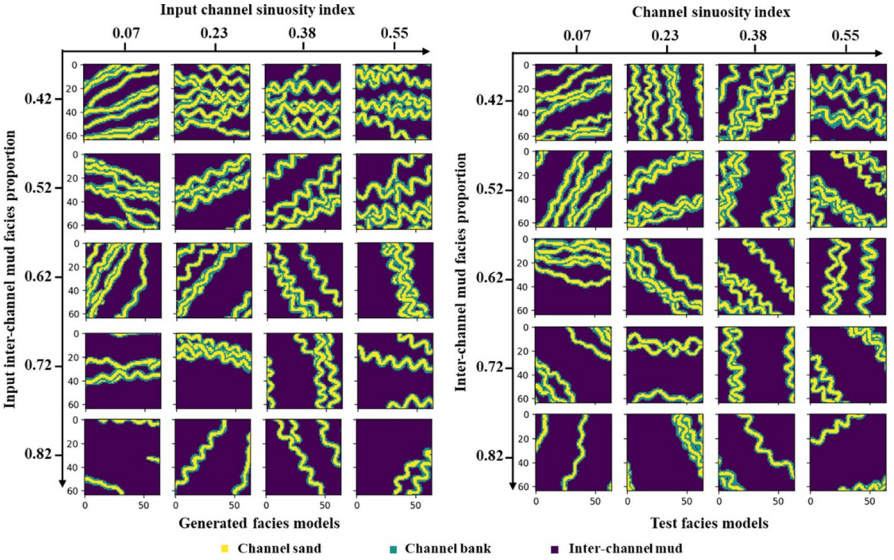

Fig. 8 Generated facies models with various input inter-channel mud proportion and channel sinuosity index values, and ground truth test facies models with the same inter-channel mud proportion and channel sinuosity index values. The width of channel sand is fixed at 3.1 pixels

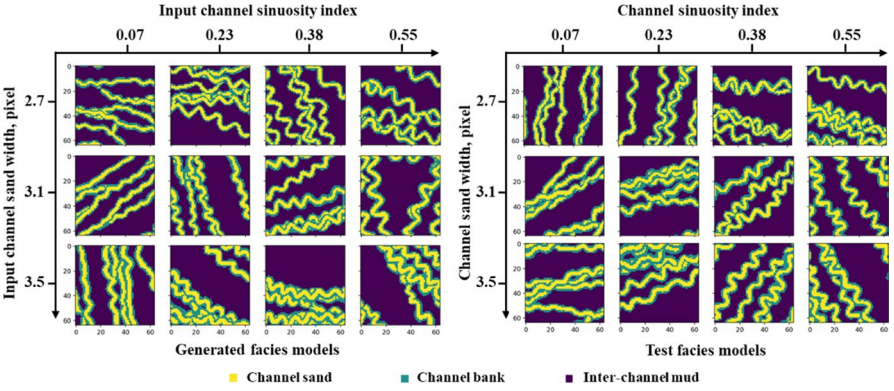

Fig. 9 Generated facies models with various input channel sand width and channel sinuosity index values, and ground truth test facies models with the same channel sand width and channel sinuosity index values. The inter-channel mud facies proportion varies from 0.51 to 0.6

with an R-squared value of 0.91. This proves the generator's strong conditioning ability to the inter-channel mud facies proportion.

We used geodesic distance and Euclidean distance transform algorithms (i.e., bwdistgeodesic and bwdist functions in MATLAB) to calculate the width, arc length, and straight-line length of channels in facies models. Figure 11 shows the cross-plot between the input channel sand width of the generator and the channel sand width calculated from the corresponding generated facies models. The input

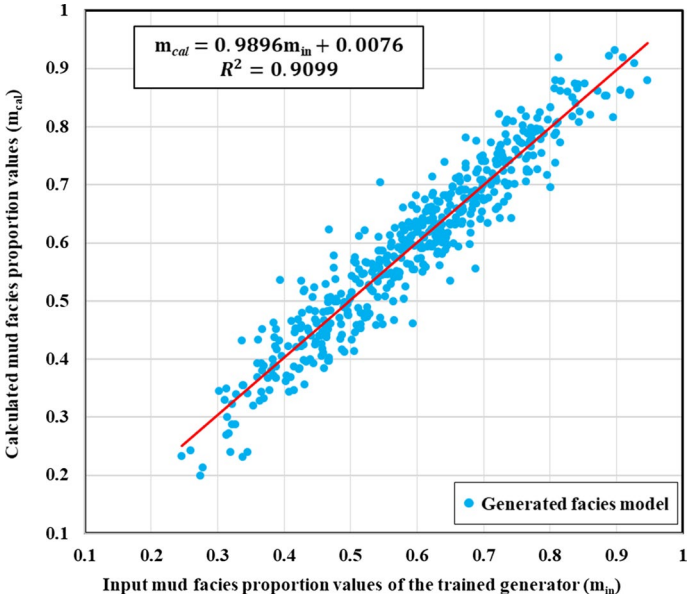

Fig. 10 Cross-plot between the input inter-channel mud facies proportion of the generator and the calculated mud facies proportion values from the corresponding generated facies models

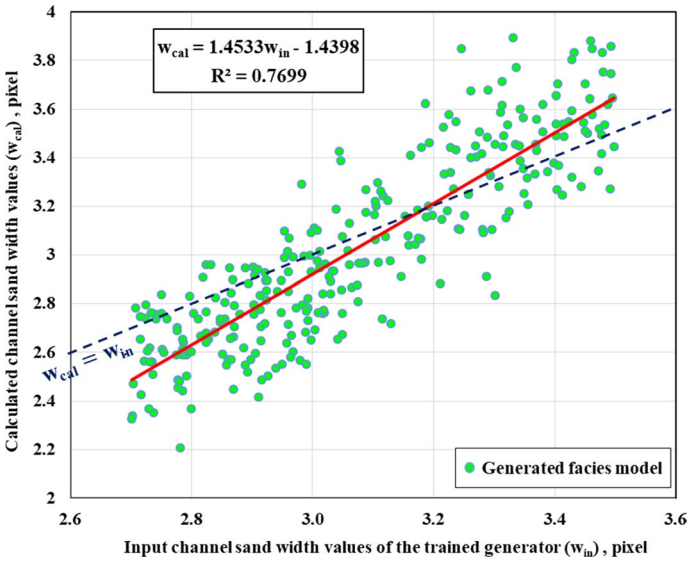

Fig. 11 The cross-plot between the input channel sand width of the generator and the channel sand width calculated from the corresponding generated facies models

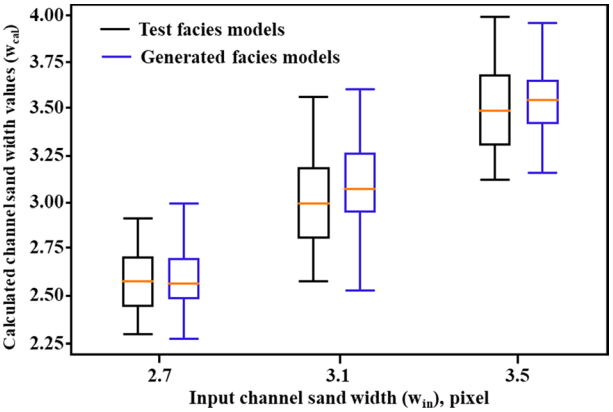

Fig. 12 The box plot of the channel sand width measured from the generated facies models and from the test set facies models

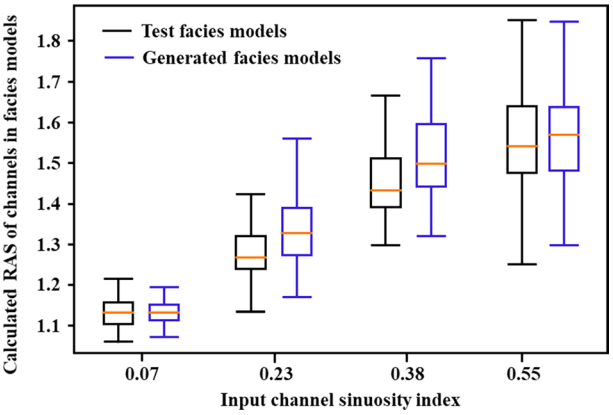

Fig. 13 The box plot of RAS of channels measured from the generated facies models and from the test set facies models

width values are close to the calculated width values, to some extent, but not as close as in the mud facies proportion case (Fig. 10); this may result from the errors in the measurement of channel sand width from facies models. Therefore, we generated 100 facies model realizations for each of the three input channel sand width values (i.e., 2.7, 3.1, and 3.5) and measured the channel sand width for each generated realization. Then, we found 100 facies models from the test dataset for each of the three input channel sand width values, and measured the channel sand width for each test facies model. Figure 12 compares the distributions (in the form of box plot) of the channel sand width measured from the generated facies model realizations and from the test facies models for the three input width values. Their distributions are very similar, indicating the generator's strong conditioning ability to the channel sand width.

In this study, we use the ratio of channel arc length to straight-line length (RAS) to represent the sinuosity of channel sand facies. Figure 13 compares the distribution of RAS calculated from the generated facies model realizations with that from the test set facies models, for each of the four input sinuosity index values (ie., 0.07, 0.23, 0.38, and 0.55). There are minor deviations in the distribution of RAS between the generated and test set facies models when the input sinuosity index equals 0.23 and 0.38, but generally speaking, the distributions of the RAS for the generated and the test set facies models are very close in terms of the four input values. This indicates the generator's strong conditioning ability to the input channel sinuosity. To sum up, the generator is quite robust in generating high-quality facies models and in conditioning to the three input global features, i.e., inter-channel mud facies proportion, width, and sinuosity index of channel sand facies.

5.2 Conditioning to Well Facies Data

The well facies data include two facies types (i.e., the inter-channel mud facies and the channel complex composite facies), so the input of the generator includes one well location indicator and one well facies indicator of the channel complex facies. The channel complex composite facies can be generated as either the channel sand or channel bank facies in the generated facies models. Based on the approach described in Sect. 2.2, the number of feature maps converted from the input well facies data is set to be 16 (Fig. 3).

In this case, we trained GANs using both the original well facies data (before well datum enlargement) and the enlarged well facies data (after well datum enlargement), and compared the two trained generators, in terms of the quality of the generated facies models, the well facies reproduction accuracy, and the local pixel noise around well points.

Similar to the previous case, weight λ in the Wasserstein loss Eq. (A4) in the ESM is set at the default value of 10 (Gulrajani et al. 2017), and weight β in Eq. (2) is decided based on quick trial-and-error experiments. Figure 14 shows randomly generated facies models and reproduction accuracies of input well facies data for different β values in the two scenarios of with and without input well data enlargement. The experiments suggest that the setting of β value relates to a trade-off between the

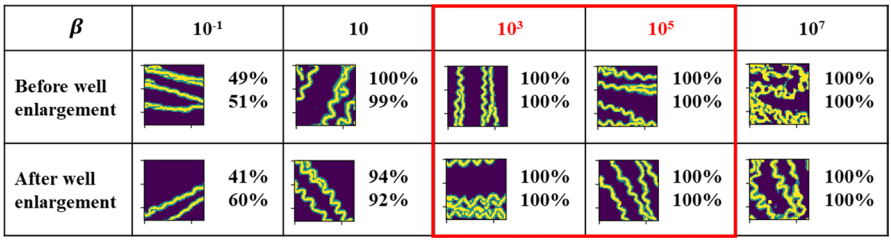

Fig. 14 Randomly generated facies models (64 × 64) and reproduction accuracies of input well facies data (upper is for channel complex facies, and lower is for mud facies) for different weight β based on the quick training schedule explained in Sect. 5.1, in the two scenarios of with and without input well data enlargement

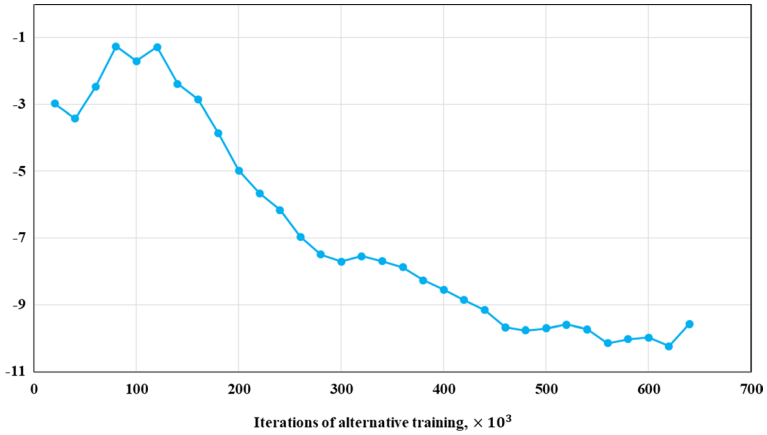

Fig. 15 The W-gp loss versus training iterations, during the training of the GAN before well datum enlargement

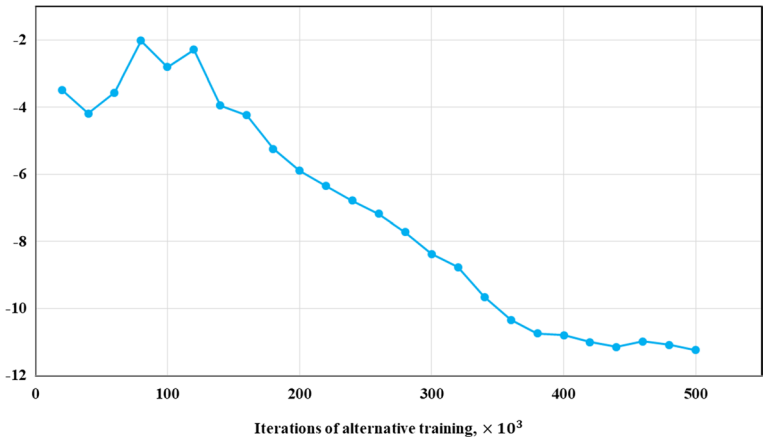

Fig. 16 The W-gp loss versus training iterations, during the training of the GAN after well datum enlargement

realism of the facies models and the reproduction accuracy of input well facies data. Weight β is suggested to be located roughly between 10^3 and 10^5.

In our study, we set weight β to be 10^3 in both scenarios. Both GANs were trained for 15 h with two GPUs and ten CPUs in parallel. Figures 15 and 16 show the Wasserstein loss with gradient penalty (W-gp loss) (Eq. (2)) versus training iterations, during training the two GANs; this loss is also called the critic loss in the GAN research community. After training, to evaluate the trained generators, we randomly sampled well facies data from the test facies models, and took the sampled well facies data and random latent vectors as inputs into the trained generators to produce facies models. Figures 17 and 19 show some facies model examples that are

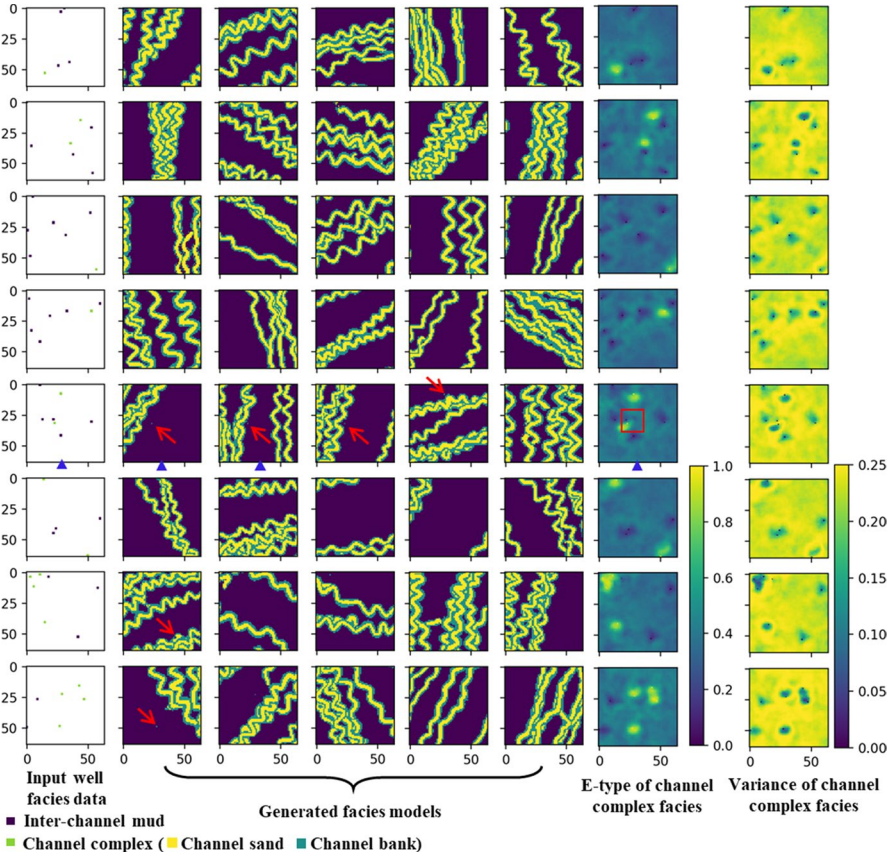

Fig. 17 Some random facies model examples generated from the generator that was trained with the original well facies data (i.e., before well datum enlargement). The red arrows point to the local pixel noise phenomena around well facies data. The E-type and variance of channel complex are calculated over 500 generated facies models. The red rectangle marks the abrupt transition around well points in the E-type map, because of the local pixel noise problem. The sub-figures marked by the blue triangles are also shown in Fig. 18 in a larger version

produced from the two trained generators with the same input well facies data and corresponding E-type and variance for channel complex facies. By manual inspection, over 90% of the generated facies models from both generators are very realistic and diversified. The number and the configuration of the input well facies data affect the quality of the generated facies models. At input well points, the E-type values of the channel complex are very close to either 1 or 0, indicating perfect conditioning of the generated facies models to input well facies data. The variance values at areas away from the well data are pretty close to the maximum variance value of 0.25; this proves good diversity of the generated facies models, to a large extent.

By quantitative evaluation over 3,000 randomly generated facies models, the well facies reproduction accuracies of the two generators are both 100% for both the channel complex facies and the inter-channel mud facies. Among the facies models

generated from the generator that was trained using the original well facies data, the local pixel noise problem was found in a small group of the facies models. These areas are pointed out by the red arrows in Fig. 17, and some of them are shown in Fig. 18 with a larger version. We also calculated the E-type map of the channel complex for each input well facies data from 500 generated facies models (the second to last column in Figs. 17, 19). Because of the local pixel noise problem, there are abrupt transitions from some well points to the surrounding values in some E-type maps; one such area is marked with the red rectangle in Fig. 17, and Fig. 18 shows a larger version of this E-type map.

Among the facies models generated from the generator trained with the enlarged well facies data, no local pixel noise problem was found. In the E-type maps of the channel complex, the transitions from the well points to their surrounding values are smooth. Figure 19 shows some random facies models generated by enlarging and inputting the well facies data in Fig. 17.

In sum, the trained generators can generate high-quality facies models with 100% well facies reproduction accuracy. The local pixel noise problem is addressed by using the well datum enlargement approach. However, well datum enlargement means forcing the surrounding 4×4-pixel area to have the same facies type as the concerning well point; this introduces an artifact bias and reduces the uncertainty of the generated facies models to some extent. Compared to the local pixel problem, this artifact bias may be acceptable in practical applications modeling spatially correlated geology. An alternative for the local pixel noise problem might be adding an additional local discriminator to especially penalize the local unrealism of generated facies models around wells, just as in GAN-based image in-painting works (e.g., Demir and Unal 2018).

We further analyzed the generator trained with the enlarged well facies data by comparing the distributions of sinuosity of the generated facies models and the test facies models. Theoretically, the two distributions should be as close as possible. Figure 20 shows the closeness of channel sinuosity distributions of test facies models, and generated facies models with different input well facies conditioning data,

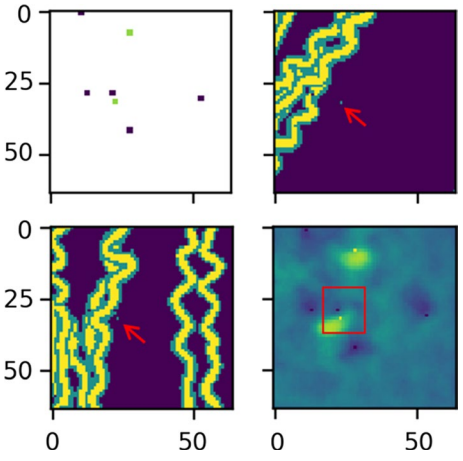

Fig. 18 A large version of the input-sparse well facies data, the generated facies models, and the E-type map of channel complex marked by blue triangles in Fig. 17

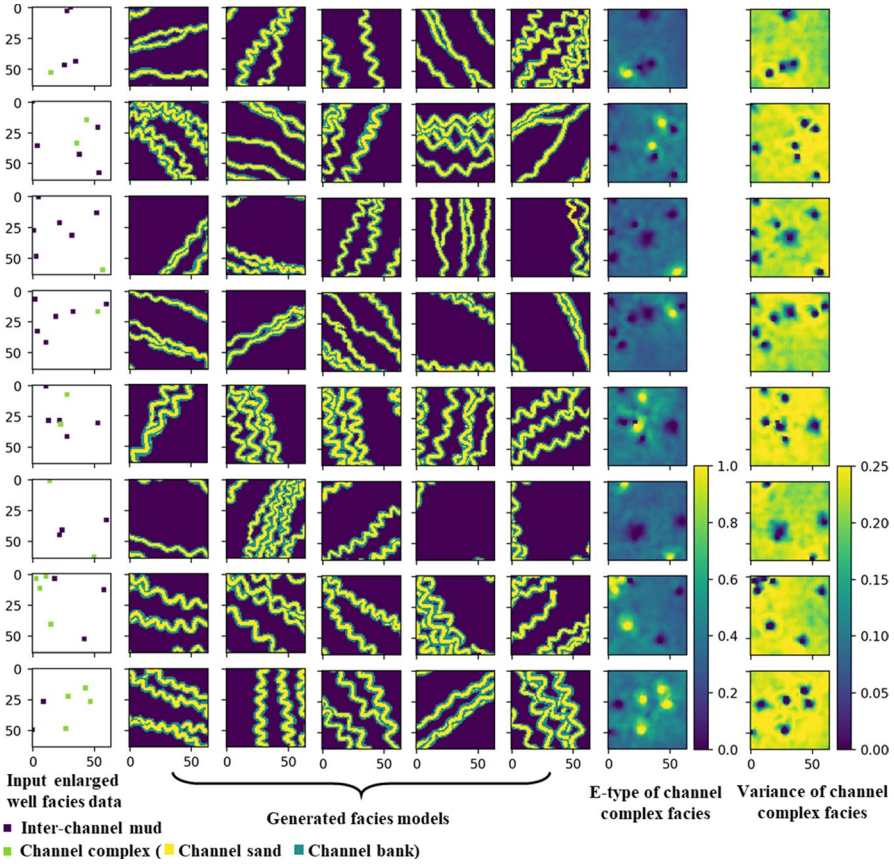

Fig. 19 Some random facies model examples generated from the generator that was trained with the enlarged well facies data. The E-type and variance of channel complex are calculated over 500 generated facies models

and the aggregate of all generated facies models. Therefore, the trained generator generates conditional facies models that capture the distribution of sinuosity present in the training data.

5.3 Conditioning to Both Global Features and Well Facies Data

We consider two subcases: (1) conditioning to both mud facies proportion and well facies data, and (2) conditioning to channel sinuosity and well facies data. The well facies data are enlarged to avoid local pixel noise.

In both subcases, weight λ in the discriminator loss Eq. (9) is set at the default value of 10 (Gulrajani et al. 2017). The discriminator loss (Eq. (9)) in this case is very similar to the discriminator loss (Eq. (4)) in the case of only conditioning to global features. Figure 6 shows good performance when weight γ was between 0.1 and 10, in the case of conditioning to global features only. Thus, γ is set to 10 here

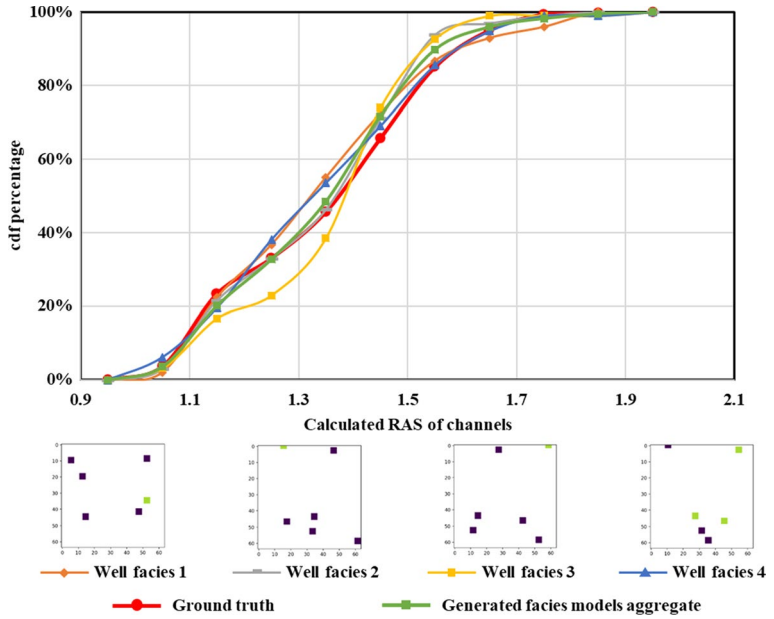

Fig. 20 Channel sinuosity distributions (cdf) of test facies models, generated facies models with different input well facies data, and the aggregate of all generated facies models

in the both subcases. The weight for global feature-based loss and well facies-based loss, β_1 and β_2 in Eqs. (7) and (8), are decided based on quick trial-and-error experiments. We only conducted the experiments for the first subcase (i.e., conditioning to mud facies proportion and well facies data). Weight β_1 and β_2 for the second subcase (i.e., conditioning to channel sinuosity and well facies data) are set to be the same as the first subcase, because both subcases share the same loss functions for the generator and the discriminator (Eqs. (7), (8), (9)). Figure 21 shows generated facies models, cross-plots between input and real mud facies proportion value of generated facies models, and reproduction accuracies of input well facies data, for various weight combinations of β_1 and β_2, in the first subcase. The settings of weight β_1 and β_2 involve a trade-off among conditioning ability to input mud facies proportion, conditioning ability to input well facies data, and realism of generated facies models. From Fig. 21, we can conclude a rough range for weight β_1 and β_2: $0.05 < \beta_1 < 0.5$ and $0.25 < \beta_2 < 25$. Because normalization is applied for the three losses (i.e., the original GAN loss, the global feature-based loss, and well facies-based loss) of the generator loss function (Eq. (7)) in this case, the magnitude of weights β_1 and β_2 is not comparable to the corresponding weights in previous cases. In the both subcases, we set β_1 and β_2 as 0.05 and 0.25, respectively.

In the first subcase, the GAN was trained for 15 h with two GPUs and ten CPUs in parallel. Figure 22 shows the loss of the modified discriminator (Eq. (4)) versus training iterations. After training, the generator takes well facies data, mud facies proportion value, and latent vector as inputs and produces a corresponding realistic facies model. Figure 23 shows some generated facies model examples and E-type

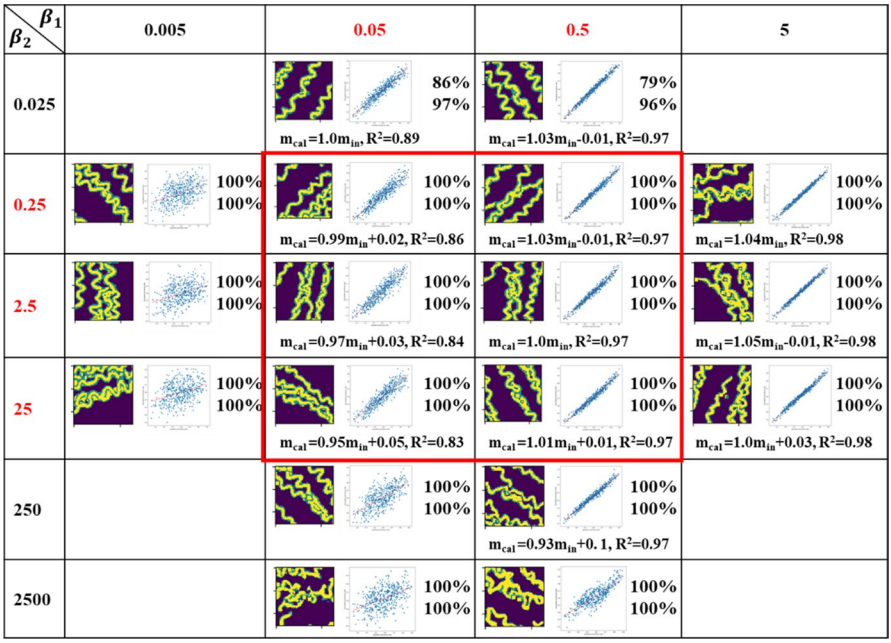

Fig. 21 Randomly generated facies models (64 × 64), cross-plots between input and real mud facies proportion values of generated facies models, trendline equations of the cross-plots, and reproduction accuracies of input well facies data (upper is for channel complex facies, and lower is for mud facies), for various weight combinations of β_1 and β_2, in the first subcase. The trainings of GANs in this figure are based on the quick training schedule explained in Sect. 5.1. In the equations, m_{in} and m_{cal} represent the input and calculated real mud facies proportion values for generated facies models, respectively

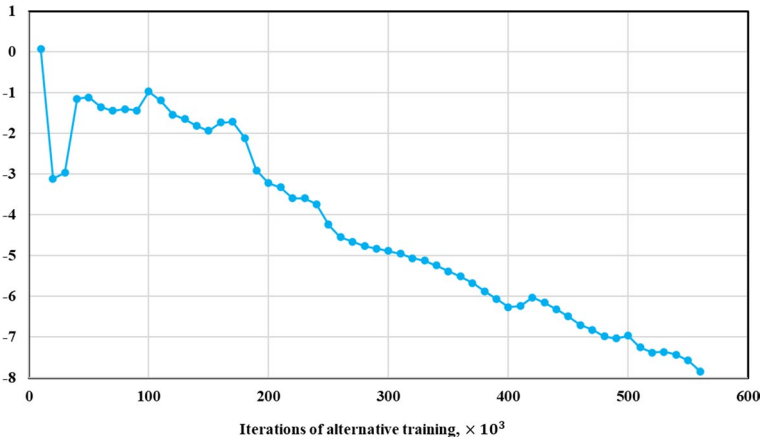

Fig. 22 The W-gp loss versus training iterations, during training of the GAN in the subcase of conditioning to mud facies proportion and well facies data

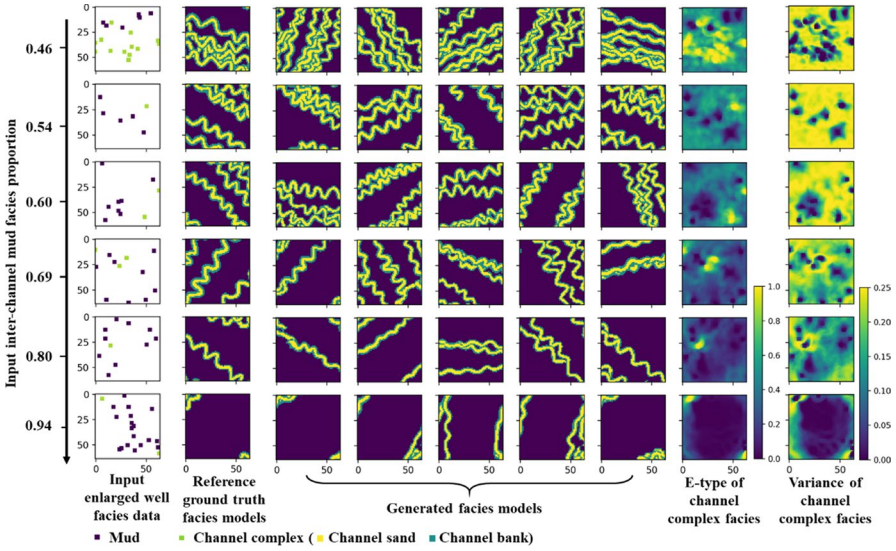

Fig. 23 Some random facies model examples generated from the trained generator in the subcase of conditioning to mud facies proportion and well facies data. The second column shows the ground truth facies models with respect to the input mud facies proportion and well facies data. The E-type and variance of the channel complex are calculated over 500 generated facies models

and variance for channel complex facies, for various input mud proportion values and random well facies datasets. By manual inspection, the generated facies models are very realistic and diversified. The variance of channel complex in areas away from wells are close to the maximum variance value of 0.25, especially when the input mud facies proportion varies from 0.46 to 0.69, also indicating good diversity in the generated facies models.

As shown in Fig. 23, the generated facies models are similar to the referenced ground truth facies models (second column of Fig. 23) with respect to mud proportion characteristic. As the input mud facies proportion value increases, the mud proportion of the generated facies models also increases. In addition, we randomly generated 500 facies models, and Fig. 24 shows the cross-plot between the input mud facies proportion values into the generator and the mud facies proportion values calculated from the corresponding generated facies models. These two proportion values are very close with an R-squared value of 0.83. This proves the generator's strong conditioning ability to input mud facies proportion values, both qualitatively and quantitatively.

In Fig. 23, the E-type values of channel complex at input well points are very close to 1 or 0. By further quantitative evaluation of 3,000 randomly generated facies models, the well facies reproduction accuracies for channel complex and inter-channel mud facies are 99.4% and 98.8%, respectively, quantitatively showing the generator's strong conditioning ability to input well facies data.

In the second subcase of conditioning to channel sinuosity and well facies data, the GAN was trained for 20 h with two GPUs and ten CPUs in parallel.

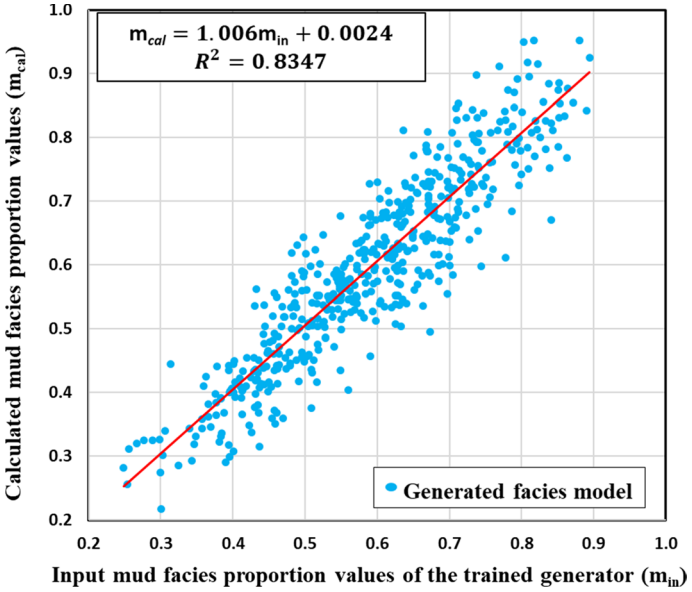

Fig. 24 Cross-plot between the input inter-channel mud facies proportion and the mud facies proportion calculated from the corresponding generated facies models, when the generator is conditioning to both mud proportion and well facies data

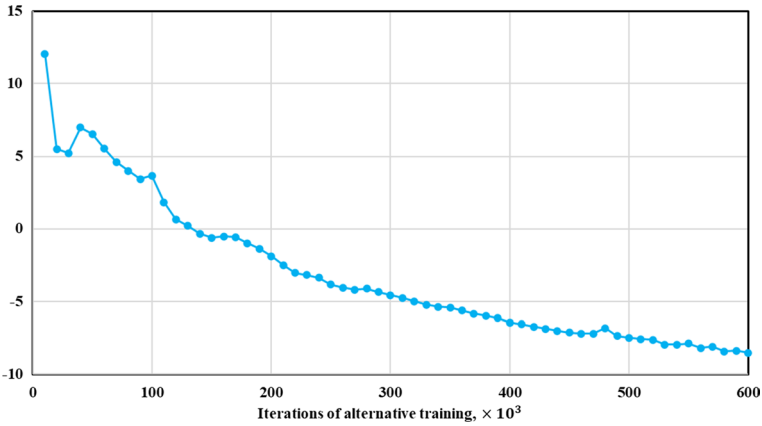

Fig. 25 The W-gp loss versus training iterations during training of the GAN in the subcase of conditioning to channel sinuosity and well facies data

Figure 25 shows the loss of the modified discriminator. The trained generator takes well facies data, channel sinuosity value, and latent vector as inputs and produces corresponding facies models. Figure 26 shows some generated facies model examples and E-type and variance of channel complex facies, for various input channel sinuosity values and random well facies data. Similar to the first subcase, by manually inspecting the generated facies models, comparing them

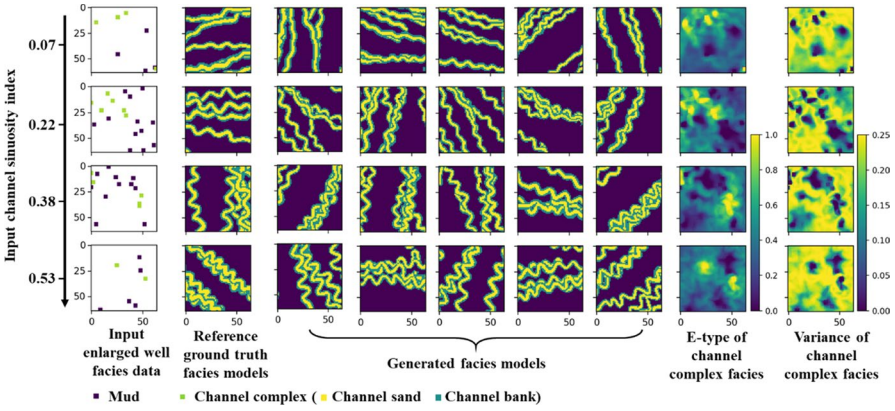

Fig. 26 Some random facies model examples generated from the trained generator in the subcase of conditioning to channel sinuosity index values and well facies data. The second column shows the ground truth facies models with respect to the input channel sinuosity values and well facies data. The E-type and variance of channel complex are calculated over 500 generated facies models

with the corresponding ground truth facies models, and inspecting E-type and variance maps, we can qualitatively conclude that the generated facies models are realistic, diversified, and conditioned to input sinuosity values and input well facies data.

Figure 27 compares the calculated RAS distributions of generated facies models and the ground truth test facies models for different input sinuosity index values. In spite of minor deviations, the overall RAS distributions of the generated facies models are very close to that of the test facies models for different sinuosity index values, further proving the generator's strong conditioning ability to the input channel sinuosity. In addition, quantitative evaluation of 3,000 randomly

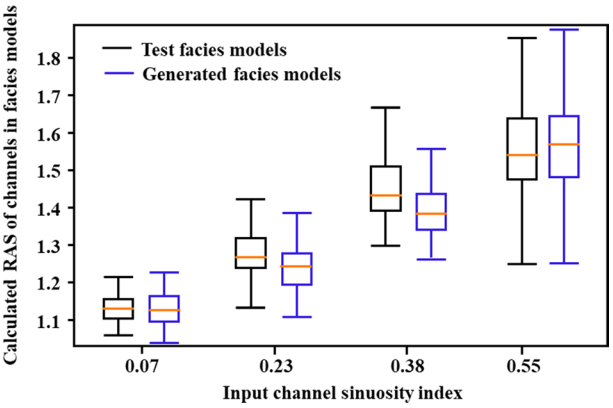

Fig. 27 The RAS box plot of generated facies models and the ground truth test facies models for different input sinuosity index values in the subcase of conditioning to channel sinuosity and well facies data

generated facies models shows that the well facies reproduction accuracies for channel complex facies and inter-channel mud facies are 99.6% and 97.9%, respectively, also indicating the generator's strong conditioning ability to input well facies data.

We further analyzed the trained generators of both subcases using the distributions of the global features that were left free and were not used for conditioning the generated facies models. Figure 28 compares the channel sand width distributions (cdf's) of the test facies models and the facies models generated by the generator of the second subcase with various input sinuosity values. The cdf's of the generated facies models are close to the cdf of the test ground truth facies models. It is a similar case for channel sinuosity and mud facies proportion in both subcases. Therefore, the two trained generators of both subcases capture the distribution of global features that are not conditioned by input data.

In terms of conditional generation using GANs, one widely used GAN architecture is conditional GAN (CGAN; Mirza and Osindero 2014), such as in Leinonen et al. (2019). Both CGAN and GANSim can achieve conditioning for the generated samples, but in slightly different ways. In CGAN, the discriminator first learns the differences between realism and non-realism and between conditioning and non-conditioning, and then guides the generator to learn the knowledge about realism and conditioning. However, in GANSim, the discriminator only learns the difference between realism and non-realism and only gives guidance about realism to the generator. The generator learns knowledge about conditioning with the help of an additional condition-based loss function (Eq. (1)). Compared to GANSim, two potential disadvantages of CGAN are as follows. First, if the discriminator does not learn the clear difference between conditioning and non-conditioning, then it cannot

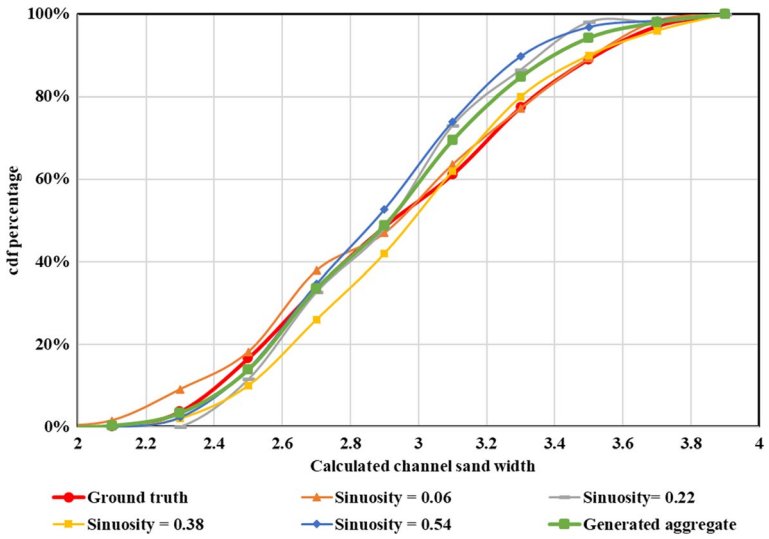

Fig. 28 Channel width distributions (cdf's) of test ground truth facies models, generated facies models by the trained generator of the second subcase with different input sinuosity values, and the aggregate of all generated facies models

give clear guidance to the generator about the learning of conditioning. Second, in the cases of multiple conditions, especially when the conditions slightly contradict each other, which is quite common in geosciences, which condition should be given priority? There is no mechanism to control the priority assignment in CGAN, but in GANSim, it can be controlled by weight assignment in loss functions (e.g., Eq. (8)).

6 Conclusions

In the GAN-based unconditional facies modeling, researchers use the original generative adversarial networks (GANs) loss function to force the generator to learn the geological patterns from the training facies models. To train the generator to also grasp the conditioning ability to input conditioning data, in the proposed GANSim, we introduce an extra loss function into GANs, which is defined as the inconsistency between the input conditioning value and the corresponding characteristics exhibited by the output facies model. In addition, we design efficient architectures for including non-spatial global features (e.g., facies ratio), sparse well facies data, and both jointly as input conditions into the generator of the GANs. The global features are taken as inputs by concatenating with the latent vector. To input the well facies data, (1) we decompose it into multiple indicator channels, (2) we downsample the indicator channels into various resolution levels, and (3) we input these downsampled and the original indicator channels into different hidden layers of the generator during the progressive growing of GANs. Such a design allows the generator to learn the geological patterns and the conditioning ability progressively from coarse scales to fine scales. We train GANs in a progressive growing manner, and after training, we evaluate both the quality of generated facies models and the conditioning ability of the generators. It turns out that the trained generators are quite robust both in generating high-quality facies models and in conditioning to the global and local data. The performance is not very sensitive to choice of weights for the different components of the loss function. The reasonable ranges of predefined weights in loss functions are quite wide, with a spread of one to three orders of magnitude. Within the range, the generated facies models are realistic, and their conditioning to input data is excellent.

The generated facies models from current generators are in two dimensions. We are extending the proposed GANSim workflow to three dimensions, and expect to also achieve conditioning ability of GANs to low-resolution “soft” probability data in future work.

Acknowledgements We acknowledge the sponsors of the Stanford Center for Earth Resources Forecasting (SCERF) and support from Prof. Steve Graham, the Dean of the Stanford School of Earth, Energy and Environmental Sciences. Some of the computing for this project was performed on the Sherlock cluster. We would like to thank Stanford University and the Stanford Research Computing Center for providing computational resources and support that contributed to these research results. Codes, data, and some results of this work are available at the GitHub site (https://github.com/SuihongSong/GeoModeling_Conditional_ProGAN).

Funding Funding was provided by the National Natural Science Foundation of China (42072146).

References

Borji A (2018) Pros and cons of GAN evaluation measures. arXiv e-prints arXiv:1802.03446
Chan S, Elsheikh AH (2017) Parametrization and generation of geological models with generative adversarial networks. arXiv e-prints arXiv:1708.01810
Chan S, Elsheikh AH (2019) Parametric generation of conditional geological realizations using generative neural networks. Comput Geosci. https://doi.org/10.1007/s10596-019-09850-7
Clerici A, Perego S (2016) A set of GRASS GIS-based shell scripts for the calculation and graphical display of the main morphometric parameters of a river channel. Int J Geosci. https://doi.org/10.4236/ijg.2016.72011
Demir U, Unal G (2018) Patch-based image inpainting with generative adversarial networks. arXiv preprint arXiv:1803.07422
Dumoulin V, Belghazi I, Poole B, Mastropietro O, Lamb A, Arjovsky M, Courville A (2016) Adversarially learned inference. arXiv e-prints arXiv:1606.00704
Dupont E, Zhang T, Tilke P, Liang L, Bailey W (2018) Generating realistic geology conditioned on physical measurements with generative adversarial networks. arXiv e-prints arXiv:1802.03065
Gan Z, Henao R, Carlson D, Carin L (2015) Learning deep sigmoid belief networks with data augmentation. J Mach Learn Res 268–276
Goodfellow I, Yoshua Bengio AC (2016) Deep learning. MIT press, Cambridge
Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial networks. arXiv e-prints arXiv:1406.2661
Gulrajani I, Ahmed F, Arjovsky M, Dumoulin V, Courville A (2017) Improved training of Wasserstein GANs. arXiv e-prints arXiv:1704.00028
Isola P, Zhu J-Y, Zhou T, Efros AA (2016) Image-to-image translation with conditional adversarial networks. arXiv e-prints arXiv:1611.07004
Karras T, Aila T, Laine S, Lehtinen J (2017) Progressive growing of GANs for improved quality, stability, and variation. arXiv e-prints arXiv:1710.10196
Karras T, Laine S, Aila T (2018) A style-based generator architecture for generative adversarial networks. arXiv e-prints arXiv:1812.04948
Laloy E, Hérault R, Jacques D, Linde N (2018) Training-image based geostatistical inversion using a spatial generative adversarial neural network. Water Resour Res. https://doi.org/10.1002/2017WR022148
Larochelle H, Murray I (2011) The neural autoregressive distribution estimator. In: Proceedings of the 14th international conference on artificial intelligence and statistics (AISTATS 2011)
Ledig C, Theis L, Huszar F, Caballero J, Cunningham A, Acosta A, Aitken A, Tejani A, Totz J, Wang Z, Shi W (2016) Photo-realistic single image super-resolution using a generative adversarial network. arXiv e-prints arXiv:1609.04802
Leinonen J, Guillaume A, Yuan T (2019) Reconstruction of cloud vertical structure with a generative adversarial network. Geophys Res Lett. https://doi.org/10.1029/2019GL082532
Mirza M, Osindero S (2014) Conditional generative adversarial nets. arXiv e-prints arXiv:1411.1784
Mosser L, Dubrule O, Blunt MJ (2017) Reconstruction of three-dimensional porous media using generative adversarial neural networks. Phys Rev E. https://doi.org/10.1103/PhysRevE.96.043309
Mosser L, Dubrule O, Blunt MJ (2018a) Stochastic reconstruction of an oolitic limestone by generative adversarial networks. Transp Porous Media. https://doi.org/10.1007/s11242-018-1039-9
Mosser L, Kimman W, Dramsch J, Purves S, De la Fuente Briceño A, Ganssle G (2018b) Rapid seismic domain transfer: seismic velocity inversion and modeling using deep generative neural networks. In: 80th EAGE conference and exhibition 2018: opportunities presented by the energy transition
Mosser L, Dubrule O, Blunt MJ (2020) Stochastic seismic waveform inversion using generative adversarial networks as a geological prior. Math Geosci. https://doi.org/10.1007/s11004-019-09832-6
Nesvold E, Mukerji T (2019) Geomodeling using generative adversarial networks and a database of satellite imagery of modern river deltas. In: Petroleum geostatistics. European Association of Geoscientists & Engineers, Vol 2019, No. 1, pp 1–5
Pyrcz MJ, Deutsch CV (2014) Geoestatistical reservoir modeling. Oxford university press

Reed S, Akata Z, Yan X, Logeswaran L, Schiele B, Lee H (2016) Generative adversarial text to image synthesis. arXiv e-prints arXiv:1605.05396

Rezende DJ, Mohamed S, Wierstra D (2014) Stochastic backpropagation and approximate inference in deep generative models. In: 31st International conference on machine learning, ICML 2014

Song S, Mukerji T, Hou J (2020) Geological facies modeling based on progressive growing of generative adversarial networks (GANs). EarthArXiv. https://doi.org/10.31223/osf.io/ycufs

Sun AY (2018) Discovering state-parameter mappings in subsurface models using generative adversarial networks. Geophys Res Lett. https://doi.org/10.1029/2018GL080404

Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1–9

Van Den Oord A, Dieleman S, Zen H, Simonyan K, Vinyals O, Graves A, Kalchbrenner N, Senior A, Kavukcuoglu K (2016a) WaveNet: a generative model for raw audio. arXiv e-prints arXiv:1609.03499

Van Den Oord A, Kalchbrenner N, Vinyals O, Espeholt L, Graves A, Kavukcuoglu K (2016b) Conditional image generation with PixelCNN decoders. arXiv e-prints arXiv:1606.05328

Zhang TF, Tilke P, Dupont E, Zhu LC, Liang L, Bailey W (2019) Generating geologically realistic 3D reservoir facies models using deep learning of sedimentary architecture with generative adversarial networks. Pet Sci. https://doi.org/10.1007/s12182-019-0328-4

Zhong Z, Sun AY, Jeong H (2019) Predicting CO2 plume migration in heterogeneous formations using conditional deep convolutional generative adversarial network. Water Resour Res. https://doi.org/10.1029/2018wr024592

Zhu JY, Park T, Isola P, Efros AA (2017) Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE international conference on computer vision

Zuo C, Yin Z, Pan Z, MacKie EJ, Caers J (2020) A tree-based direct sampling method for stochastic surface and subsurface hydrological modeling. Water Resour Res. https://doi.org/10.1029/2019WR026130