Generation of non-stationary stochastic fields using Generative Adversarial Networks

Alhasan Abdellatif *, Ahmed H. Elsheikh †, Daniel Busby 2, and Philippe Berthet 2

1 Heriot-Watt University
2 TotalEnergies

March 9, 2023

Abstract

In the context of generating geological facies conditioned on observed data, samples corresponding to all possible conditions are not generally available in the training set and hence the generation of these realizations depends primary on the generalization capability of the trained generative model. The problem becomes more complex when applied on non-stationary fields. In this work, we investigate the problem of using Generative Adversarial Networks (GANs) models to generate non-stationary geological channelized patterns and examine the models generalization capability at new spatial modes that were never seen in the given training set. The developed training method based on spatial-conditioning allowed for effective learning of the correlation between the spatial conditions (i.e. non-stationary maps) and the realizations implicitly without using additional loss terms or solving optimization problems for every new given data after training. In addition, our models can be trained on 2D and 3D samples. The results on real and artificial datasets show that we were able to generate geologically-plausible realizations beyond the training samples and with a strong correlation with the target maps.

Keywords: Generative Adversarial Networks (GANs), Non-stationary Stochastic Fields, Multipoint Geostatistics.

1 Introduction

Generating stochastic fields has many applications in geosciences and reservoir management. Modelling these fields at the reservoir scale is an essential step for addressing uncertainty quantification or inverse problems in the subsurface. One of the classical approaches is the Multiple Point Statistics (MPS) algorithms Strebelle (2002) which were designed for geo-statistical simulation based on spatial patterns in a training image. Many variants of MPS have been developed over time such as direct sampling techniques Mariethoz et al. (2010) and cross-correlation based methods Tahmasebi & Sahimi (2013). The trained non-parametric model can be used to generate realizations constrained to well and seismic data Hashemi et al. (2014); Rezaee & Marcotte (2017); Arpat & Caers (2007); Tahmasebi et al. (2014). However, MPS algorithms suffer from many limitations such as the high computational cost Li et al. (2016), limited variability Emery & Lantuéjoul (2014) and inability to model complex non-stationary patterns T. Zhang et al. (2019).

Following the success of deep learning in computer vision, recent published work have considered deep generative models such as Generative Adversarial Networks (GANs) Goodfellow et al. (2014)
for generation of stochastic fields. A main advantage of GANs over Multiple Point Statistics (MPS) is that the trained model is used to parametrize a high-dimensional sample using a low-dimensional input. This allowed GANs to be used for a variety of tasks, for example, GANs were used to model 3D structure of porous media Mosser et al. (2017), the low-dimensionality of GANs latent vector was utilized to perform parametrization of the spatial permeability fields in the subsurface Chan & Elsheikh (2017) and geostatistical inversion was performed after training GANs models on 2D and 3D categorical samples Laloy et al. (2018).

GANs have also been used to generate geological realizations conditioned on hard data (e.g., point measurements at wells) and soft data (e.g., probability maps). Approaches for generation of conditioned stochastic realizations could be classified into two categories: post-GANs and concurrent-GANs. In post-GANs approaches, a new optimization problem is solved after training GANs where the latent vector is searched to find realizations that match the target data. For example, gradient descent method was used in Dupont et al. (2018); T. Zhang et al. (2019), a Markov Chain Monte Carlo sampling algorithms were used in Nesvold & Mukerji (2019); Laloy et al. (2018) and Chan & Elsheikh (2019) trained an inference network to map the normally distributed outputs to a distribution of latent vectors that satisfies the required conditions. The main drawback of using post-GANs approaches is the additional cost needed to solve the second optimization problem which can often be expensive. In addition, we would need to solve different problems for every new observed data (e.g. new condition).

In concurrent-GANs approaches, the training of GANs is modified to pass the conditional data to the GANs generator network. After training, the trained generator can then simulate realizations based on the input data without the need to solve another optimization problem. In Abdellatif et al. (2022), conditional GANs were used to generate unrepresented global proportions of geological facies. Cycle-consistent GANs Zhu et al. (2017) has been used for domain mapping, for example mapping between physical parameters and model state variables A. Y. Sun (2018) and mapping between seismic data and geological model Mosser et al. (2018). In Zhong et al. (2019), a GANs model with a U-net architecture Ronneberger et al. (2015) was used to map high-dimensional input to CO₂ saturation maps. However, the one-to-one mapping using Cycle-GANs or U-net architecture is not suitable for generating multiple stochastic facies conditioned on a single observed data.

In Song et al. (2021b), condition-based loss functions were used to condition facies on hard data and global features and later they extended the method for spatial probability maps in Song et al. (2021a). Pix2pix method Isola et al. (2017) has been used in Pan et al. (2021) for geophysical conditioning by adding additional losses for seismic and well log conditions. However, condition-based losses require designing manual functions that compute the consistency between the generated samples and target conditions (e.g., computing facies frequency for the generated realizations to mimic real probability maps Song et al. (2021a)). The design of such function is arbitrary and this is conceptually different than GANs, where the learning is done implicitly from the training data by joint training of both the generator and the discriminator which tells what is good versus bad samples. Moreover, including additional losses in GANs relies on careful weighting between the conditions losses and the original loss in GANs which requires extensive hyper-parameters search (e.g., see Fig 6 in Song et al. (2021b)).

In this work, we developed a concurrent-GANs approach that conditions the geological realizations on spatial maps, describing the distribution of facies proportions across the spatial domain, without using condition-based losses. Further, we were able to generate new realizations that match spatial maps never seen in the training samples; such generalization capability is very useful especially when the characteristics of actual reservoirs differ significantly from those of the available training samples. To model the spatial modes, we used conditional GANs (cGANs) Mirza & Osindero (2014) where each spatial map is provided as a condition to the neural networks. By letting the maps modulate the generator layers in a spatial setting through the SPADE algorithm Park et al. (2019), the GANs models learned the correlation between the spatial conditions and the samples implicitly. The developed training approach solves a single optimization problem (i.e., a concurrent-GANs method) without using condition-based losses and hence there is no need to design condition-consistency functions or to perform careful hyper-parameter search for balancing weights. Results on 2D and 3D samples show a strong correlation between the generated samples and the target maps as well as the trained models.
were able to generalize to unseen spatial modes.

The rest of the paper is organized as follows: in section 2, we discuss the algorithm of conditional GANs used in our experiments and we present the training datasets and the implementation details. In section 3, the results of the experiments are shown. Finally, conclusions are provided in section 4.

2 Method and Materials

2.1 Method

Generative adversarial networks (GANs) Goodfellow et al. (2014) are trained to learn the underlying distribution of training samples. It consists of two convolutional neural networks: a generator \( G \) and a discriminator \( D \). The generator maps a random noise \( z \) to a realization \( G(z) \) while the discriminator takes samples from the real and the generated sets and is optimized to output the probability of the samples being real (i.e., not generated by the generator). The generator is then optimized such that the generated samples have high probability \( D(G(z)) \). The two networks are trained in an adversarial setting defined by the objective function \( V(G,D) \):

\[
\min_G \max_D V(G,D) = \mathbb{E}_{x \sim p_x}[\log D(x)] + \mathbb{E}_{z \sim p_z}[\log(1 - D(G(z)))].
\]

Given a spatial map \( M \) that describes the spatial distribution of a geological facies (e.g., channels), we can direct the generated samples to match a particular spatial map \( M \) by using a conditional GANs Mirza & Osindero (2014) method, where the condition \( M \) is passed to both the generator network \( G \) and discriminator network \( D \) during training. Similar to the concurrent-GANs methods, after training we can generate multiple realizations conditioned on \( M \) by simply passing \( M \) and different latent \( z \) vectors to the generator without solving a new optimization problem. The discriminator would then output the conditional probability of the sample being real given its input map. The objective function of conditional GANs is then:

\[
\min_G \max_D V(G,D) = \mathbb{E}_{x \sim p_x}[\log D(x|M)] + \mathbb{E}_{z \sim p_z}[\log(1 - D(G(z,M)|M))].
\]

To accommodate for the spatial nature of the map, we followed the spatially adaptive de-normalization (SPADE) conditioning method developed in Park et al. (2019), where a segmentation mask is used to modulate the generator layers such that it could generate natural images based on the mask.

In our work, we replaced the categorical mask by a continuous map representing the spatial proportions of the channels, the maps were calculated for each sample before training. The SPADE method goes as following, for each layer \( i \) and channel \( c \) in the generator, each activation \( h_{i,c,x,y} \) (\( h_{i,c,x,y,z} \) in case of 3D samples) is normalized using the mean \( \mu_{i,c} \) and standard variation \( \sigma_{i,c} \) computed over both batch instances and channel spatial locations. The result is then de-normalized spatially, i.e., per spatial position, using parameters \( \gamma \) and \( \beta \) which are learnable functions of \( M \). The calculation of SPADE algorithm for the 2D case is shown in following equation:

\[
\hat{h}_{i,c,x,y}(M) = \gamma_{i,c,x,y}(M) \frac{h_{i,c,x,y} - \mu_{i,c}}{\sigma_{i,c}} + \beta_{i,c,x,y}(M), \quad M \in \mathbb{R}^{H \times W},
\]

where the learnable parameters \( \gamma \) and \( \beta \) were obtained using two successive convolutional layers, separated by a ReLU activation function, directly applied to the map \( M \) and \( H \times W \) is the grid dimension at which the channels proportions were calculated.

We can then modify the spatial proportions of the generated facies by modifying \( M \) which will in turn modulate the generator activations through \( \gamma \) and \( \beta \). Since each layer of the generator operates at different resolution, the map \( M \) is down-sampled (or up-sampled in case \( M \) has a lower resolution) to match the resolution of the features at each layer. When extending the problem of 3D samples,
the proportions maps will be calculated for all 3 dimensions. Equation 3 for conditioning on 3D \( M \) becomes:

\[
\hat{h}_{i,c,x,y,z}(M) = \gamma_{i,c,x,y,z}(M) \frac{h_{i,c,x,y,z}(M) - \mu_{i,c}}{\sigma_{i,c}} + \beta_{i,c,x,y,z}(M), \quad M \in \mathbb{R}^{H \times W \times D}. \tag{4}
\]

In the discriminator, features computed from the map \( M \), using convolutional layers, are concatenated with spatial features computed from the input image at an intermediate layer of the discriminator. The intermediate layer is chosen such that its resolution matches the resolution of \( M \). For example, if the input images are of resolution 64 × 64 and \( M \) is 4 × 4 then the concatenation is done at the fourth layer.

### 2.2 Implementation Details

All models are based on the ResNet architecture He et al. (2016) following Gulrajani et al. (2017). The discriminator halves the features resolution and doubles the number of features channels after each layer before it outputs a single value indicating the probability of the image being real. We apply spectral normalization to the discriminator’s weights Miyato et al. (2018) and the self-attention mechanism H. Zhang et al. (2019) in both the generator and discriminator at an intermediate layer of resolution 32×32. For all experiments, the models are trained using the Adam optimizer with fixed learning rate of 0.0002 for both networks and a batch-size of 32. The latent vector \( z \) is sampled from a multivariate standard normal distribution of dimension 128. The final checkpoint used is based on an exponential moving average of the generator weights with a decaying factor of 0.999 following Brock et al. (2018). When updating the generator we used \(-E_{z \sim p_z} \log(D(G(z)))\) as proposed in Goodfellow et al. (2014).

For the 3D model, 3D convolutions, 3D batch-normalization and 3D up-sampling operations are adapted from the PyTorch framework. In this case, the starting vector \( z \) is reshaped to 4 × 4 × 2 and passed to the generator to form 64 × 64 × 32 3D images. To handle the size of the 3D samples, we trained our models on 4 parallel GPUs of GeForce RTX 3090. The architectures of the generator and discriminator networks are depicted in Figures 1 and 2, respectively, for simplicity, we removed the self-attention blocks and showed only the 2D models.

Figure 1: Generator architecture: the stochastic input \( z \) is projected into a generated image and the conditioning map is passed to each layer in the generator for spatial modulation. SPADE layer is similar to the one explained in 232019Park et al.Park, Liu, Wang, & Zhu ().
Figure 2: Discriminator architecture: the conditioning map is passed to convolutional layers and the resulting features are concatenated with the input image features (i.e., blue and green features). The discriminator output is the conditional probability of $x$ being real given its corresponding map $M$.

## 2.3 Datasets

In our experiments, we used three datasets to test our models: a) a 2D artificial dataset of 3 facies: channels, levees and background b) a 2D dataset of masks of the Brahmaputra river with binary facies: channels and background and c) a 3D artificial dataset of binary facies: channels and background. Samples from the 2D datasets are shown in Figure 3. The non-stationarity in the datasets are due to the variations in the channels proportions across the spatial domain, we describe the 2D datasets and the preprocessing steps below. Details about the 3D dataset are available in A.

Samples of the first dataset (a) are generated using a geo-modelling tool that mimics depositional environment formation based on random walks Massonnat (2019). Horizontal and vertical flipping were performed to increase the sample size from 2,000 to 8,000, this added an additional mode in the training set (i.e., large channel proportion on the right side). The Brahmaputra river mask, dataset (b), was obtained from the supplementary data in Schwenk et al. (2020). The large mask of size $13091 \times 11680$ was cropped to images of size $256 \times 256$ with a stride of 64 then the cropped images were rotated such that they have vertical alignment with the central line of the large mask. The central line was computed using RivGraph library Schwenk et al. (2020). Horizontal and vertical flipping were performed to increase the variations in the training set (increasing sample size from 1788 to 7152). All images of the 2D two datasets were resized to $64 \times 64$ resolution to match the networks input.

For each sample in the training set, its the channel proportions map was calculated at a resolution of $M \in 4 \times 4$ for the 2D samples and $M \in 4 \times 4 \times 2$ for the 3D samples. Although they could be calculated at higher resolutions, we chose to mimic the low resolutions usually obtained from seismic surveys, see Figures 4 and 5. After training, $M$ can be arbitrary selected to mimic the non-stationarity (e.g., $p > 0.4$ is a high proportion and $p \leq 0.16$ is a low proportion).

## 3 Results and Discussion

Results on the 2D artificial dataset and the Brahmaputra river masks are shown in Figures 4 and 5, respectively. The leftmost column shows the conditioning maps $M$, the middle columns are the corresponding generated images $G(z, M)$ and rightmost column shows the mean per pixel maps calculated over 2,000 generated samples. For each row, we used the same conditioning map and different $z$ vectors. As shown, the generated realizations exhibit variability due to the randomness of $z$ but overall respect the conditioning map, we show the stochastic variations due to $z$ in B. The results of the 3D models are shown in A.
From the results, it is clear that the models were able to generalize successfully to modes never seen in the training datasets. Visually, the generated samples at all modes are geologically-plausible; for example, the channels connectivity is maintained for both datasets as well as the models were able to generate levees surrounding the channels, in the case of the first dataset, irrespective of the location of the channels which indicates that the models have not memorized the training samples.

To quantify the correlation between the target proportions maps and the corresponding generated samples, Figures 6 and 7 show $4 \times 4$ cross plots where each plot represents proportions of each section in the $4 \times 4$ grid of $M$. The results show a strong correlation between the target and the generated proportions with an $R^2$ value of almost 1. In addition, the models were able to extrapolate to unseen range of proportions (the red dots in Figure 6) in the artificial training set while the proportions for the entire range $[0, 1]$ were represented in the real dataset. The generalization capacity of GANs can then be understood from two perspectives:

1. The model can generalize to unseen proportions within each section in the grid as shown in Figure 6.

2. The model can generalize to unseen non-stationary patterns over the whole image (combining structures from patches not in the training set) as shown in Figures 4 and 5.

We also calculated the two-point probability function which measures the probability that two points, separated by a given distance, have the same channel facies. In Figure 8, the function is computed for only two sections (the top left section and the section at the second row and the second column) in the $4 \times 4$ grid for the 2D datasets at 4 different conditions. Figures (a) and (b) show the probability functions for the artificial dataset and Figures (c) and (d) are for the real binary masks dataset. As shown, the function for the generated samples (i.e., the solid lines) is consistent with the one for the training samples (i.e., dashed lines). The generalization capability of the model is demonstrated at some unrepresented conditions (e.g., 80% in Figure (a) and 60% and 80% in Figure (d)) where the functions follow the general trend. At large distance, the functions mismatch increases, this could be due to the fact that the model tries to adjust the geological consistency at the boundaries which might be different from the training samples.
Figure 4: Generated non-stationary realizations on the artificial dataset: the input conditioning maps are in the leftmost columns, the middle columns are the generated samples and the per-pixel mean maps are in the rightmost columns. The last four rows show generated samples with never seen maps.
Figure 5: Generated non-stationary realizations on the real masks of the Brahmaputra river: the input conditioning maps are in the leftmost columns, the middle columns are the generated samples and the per-pixel mean maps are in the rightmost columns. The last three rows shows generated samples with never seen maps.
Figure 6: Cross plots between the generated channel proportions and target proportions in the 4 × 4 conditioning maps for the artificial dataset. In each plot, we change the corresponding value in the map and fix all the other values. The blue dots lie within the represented range in the training set while the red ones lie beyond the seen range.
Figure 7: Cross plots between the generated channel proportions and target proportions in the $4 \times 4$ conditioning maps for the real masks dataset. In each plot, we change the corresponding value in the map and fix all the other values. Notice that all points are blue as the training proportions cover the whole range $[0, 1]$. 
Figure 8: Two-point probability functions calculations for the 2D datasets. The dashed lines indicate calculations from the training samples at the same proportions while the solid lines are for the generated samples. The computation was done for two sections in the $4 \times 4$ grid for the 2D datasets. Figures (a) and (b) show the probability functions calculated for the artificial dataset while Figures (c) and (d) show the probability functions for the real binary masks.

Due to the convolutional nature of the conditioning method, after training we can use a conditioning map of a higher resolution than the one used during training. In Figure 9, we show realizations generated using conditioning maps with resolutions of $4 \times 4$, $8 \times 8$, $16 \times 16$ and $32 \times 32$. As shown, despite the fact that the model has been trained only on $4 \times 4$ maps, it is able to generate plausible realizations using maps of higher resolutions.

To further assess the trained models, we have performed a uniform flow simulation on the artificial samples and the corresponding generated samples and presented the flow statistics in C.

## 4 Conclusion

In this work, we investigated the generalization ability of GANs models to generate non-stationary 2D and 3D realizations. The conditioning algorithm used allowed the model to learn the spatial correlations between the target maps and the generated realizations without solving optimization problem for every new observed data and without using arbitrary loss functions. Our models were able to generate consistent 2D and 3D realizations, even with spatial maps never seen during training,
on cases representing both artificial and real geological samples. Future work might include generating non-stationary data from stationary training sets and extending the generated field of view to infinite dimensions.

## 5 Data Availability Statement

The artificial datasets used to train the models were generated using a geostastical simulation program based on random walks Massonnat (2019) found on https://doi.org/10.1016/j.cageo.2022.105085. The mask of the Brahmaputra river was obtained from the supplementary data of Schwenk et al. (2020) found on https://doi.org/10.5194/esurf-8-87-2020 and were pre-processed by the RivGraph library https://github.com/jonschwenk/RivGraph. The raw 3D dataset can be found on https://github.com/GeoDataScienceUQ/GANRiverI. The code used to train the GANs models is available on the public repository https://github.com/Alhasan-Abdellatif/NonstationaryGANs.

## 6 Acknowledgment

The first author thanks TotalEnergies for the financial support. The authors acknowledge TotalEnergies for authorizing the publication of this paper and would like to acknowledge the use of the flow simulation codes developed by Dr. Shing Chan (Oxford Big Data Institute) during his PhD work at Heriot-Watt university.
References

Abdellatif, A., Elsheikh, A. H., Graham, G., Busby, D., & Berthet, P. (2022). Generating unrepresented proportions of geological facies using generative adversarial networks. *Computers and Geosciences*, 105085. doi: https://doi.org/10.1016/j.cageo.2022.105085

Arpat, G. B., & Caers, J. (2007). Conditional simulation with patterns. *Mathematical Geology*, 39(2), 177–203.

Brock, A., Donahue, J., & Simonyan, K. (2018). Large scale GAN training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*.

Chan, S., & Elsheikh, A. H. (2017). Parametrization and generation of geological models with generative adversarial networks. *arXiv preprint arXiv:1708.01810*.

Chan, S., & Elsheikh, A. H. (2019). Parametric generation of conditional geological realizations using generative neural networks. *Computational Geosciences*, 23(5), 925–952.

Dupont, E., Zhang, T., Tilke, P., Liang, L., & Bailey, W. (2018). Generating realistic geology conditioned on physical measurements with generative adversarial networks. *arXiv preprint arXiv:1802.03065*.

Emery, X., & Lantuëjoul, C. (2014). Can a training image be a substitute for a random field model? *Mathematical Geosciences*, 46(2), 133–147.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., . . . Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems* (pp. 2672–2680).

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., & Courville, A. C. (2017). Improved training of Wasserstein GANs. In *Advances in neural information processing systems* (pp. 5767–5777).

Hashemi, S., Javaherian, A., Ataee-pour, M., Tahmasebi, P., & Khoshdel, H. (2014). Channel characterization using multiple-point geostatistics, neural network, and modern analogy: A case study from a carbonate reservoir, southwest Iranian. *Journal of Applied Geophysics*, 111, 47–58.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 770–778).

Isola, P., Zhu, J.-Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 1125–1134).

Laloy, E., Héralt, R., Jacques, D., & Linde, N. (2018). Training-image based geostatistical inversion using a spatial generative adversarial neural network. *Water Resources Research*, 54(1), 381–406.

Li, X., Mariethoz, G., Lu, D., & Linde, N. (2016). Patch-based iterative conditional geostatistical simulation using graph cuts. *Water Resources Research*, 52(8), 6297–6320.

Mariethoz, G., Renard, P., & Straubhaar, J. (2010). The direct sampling method to perform multiple-point geostatistical simulations. *Water Resources Research*, 46(11).

Massonnat, G. (2019). Random walk for simulation of geobodies: A new process-like methodology for reservoir modelling. In *Petroleum geostatistics 2019* (Vol. 2019, pp. 1–5). doi: https://doi.org/10.1016/j.cageo.2022.105085

Mirza, M., & Osindero, S. (2014). Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.
Miyato, T., Kataoka, T., Koyama, M., & Yoshida, Y. (2018). Spectral normalization for generative adversarial networks. arXiv preprint arXiv:1802.05957.

Mosser, L., Dubrule, O., & Blunt, M. J. (2017). Reconstruction of three-dimensional porous media using generative adversarial neural networks. Physical Review E, 96(4), 043309.

Mosser, L., Kimman, W., Dransch, J., Purves, S., De la Fuente Briceño, A., & Ganssle, G. (2018). Rapid seismic domain transfer: Seismic velocity inversion and modeling using deep generative neural networks. In 80th eage conference and exhibition 2018 (Vol. 2018, pp. 1–5).

Nesvold, E., & Mukerji, T. (2019). Geomodeling using generative adversarial networks and a database of satellite imagery of modern river deltas. In Petroleum geostatistics 2019 (Vol. 2019, pp. 1–5).

Pan, W., Torres-Verdín, C., & Pyrcz, M. J. (2021). Stochastic pix2pix: a new machine learning method for geophysical and well conditioning of rule-based channel reservoir models. Natural Resources Research, 30(2), 1319–1345.

Park, T., Liu, M.-Y., Wang, T.-C., & Zhu, J.-Y. (2019). Semantic image synthesis with spatially-adaptive normalization. In Proceedings of the ieee conference on computer vision and pattern recognition (pp. 2337–2346).

Rezaee, H., & Marcotte, D. (2017). Integration of multiple soft data sets in mps thru multinomial logistic regression: a case study of gas hydrates. Stochastic Environmental Research and Risk Assessment, 31(7), 1727–1745.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In International conference on medical image computing and computer-assisted intervention (pp. 234–241).

Schwenk, J., Piliouras, A., & Rowland, J. C. (2020). Determining flow directions in river channel networks using planform morphology and topology. Earth Surface Dynamics, 8(1), 87–102. doi: https://doi.org/10.5194/esurf-8-87-2020

Song, S., Mukerji, T., & Hou, J. (2021a). Bridging the gap between geophysics and geology with generative adversarial networks. IEEE Transactions on Geoscience and Remote Sensing.

Song, S., Mukerji, T., & Hou, J. (2021b). GANSim: Conditional facies simulation using an improved progressive growing of generative adversarial networks (GANs). Mathematical Geosciences, 1–32.

Strebelle, S. (2002). Conditional simulation of complex geological structures using multiple-point statistics. Mathematical geology, 34(1), 1–21.

Sun, A. Y. (2018). Discovering state-parameter mappings in subsurface models using generative adversarial networks. Geophysical Research Letters, 45(20), 11–137.

Sun, C., Demyanov, V., & Arnold, D. (2021). Comparison of popular generative adversarial network flavours for fluvial reservoir modelling. In 82nd eage annual conference & exhibition (Vol. 2021, pp. 1–5).

Sun, C., Demyanov, V., & Arnold, D. (2023). Gan river-i: A process-based low ntg meandering reservoir model dataset for machine learning studies. Data in Brief, 46, 108785.

Tahmasebi, P., & Sahimi, M. (2013). Cross-correlation function for accurate reconstruction of heterogeneous media. Physical review letters, 110(7), 078002.

Tahmasebi, P., Sahimi, M., & Caers, J. (2014). Ms-cs:im: accelerating pattern-based geostatistical simulation of categorical variables using a multi-scale search in fourier space. Computers & Geosciences, 67, 75–88.
Figure 10: Samples from the 3D training dataset, all images used to train the 3D models are of size $64 \times 64 \times 32$.

Zhang, H., Goodfellow, I., Metaxas, D., & Odena, A. (2019). Self-attention generative adversarial networks. In *International conference on machine learning* (pp. 7354–7363).

Zhang, T., Tilke, P., Dupont, E., Zhu, L., Liang, L., & Bailey, W. (2019). Generating geologically realistic 3d reservoir facies models using deep learning of sedimentary architecture with generative adversarial networks. In *International petroleum technology conference*.

Zhong, Z., Sun, A. Y., & Jeong, H. (2019). Predicting CO$_2$ plume migration in heterogeneous formations using conditional deep convolutional generative adversarial network. *Water Resources Research*, 55(7), 5830–5851.

Zhu, J.-Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision* (pp. 2223–2232).

A Generating non-stationary 3D realizations

In this section, we present the 3D datasets and results of the trained models using the methodology in section 2.

A.1 3D Dataset

The 3D dataset (c) was obtained from C. Sun et al. (2023) and has been used in C. Sun et al. (2021) to compare different GANs models. The original dataset is composed of 25 3D images, of size $256 \times 256 \times 640$, produced using FLUMY$^\text{Tm}$ computer simulation program and grouped into 5 groups with different avulsion rates. We selected samples from only the first two groups with low avulsion rates and performed cropping on each image such that the cropped images are of size $64 \times 64 \times 32$. We used the 3-facies dataset and converted all images to be binary by merging the point bar and channels facies. Finally, flipping was performed to increase the diversity within the dataset. Samples from the dataset are depicted in Figures 10 and 11.

A.2 Results of the 3D Models

To demonstrate the effectiveness of our models to reconstruct the 3D samples, we trained an unconditional model and show the 3D generated realizations in Figures 12 and 13. As shown, the images across different planes resemble those found in the training set in Figure 11; which confirms that GANs were
Figure 11: Samples from the 3D training dataset as 2D slices: (a) layers of xy-plane (b) layers of xz-plane (c) layers of yz-plane.
Figure 12: Unconditional generated 3D samples.

able to model the 3D structure of the channels. In Figure 14, we present the 3D conditional results: in each row, the first column shows the target conditional map $M$ and the remaining columns show 3 different generated 3D realizations $x_i = G(z_i, M)$, where each realization is generated with a different $z_i$ and the same map $M$. From these results, it is clear that our 3D models have learnt a disentangled representation between the $z$ vector which drives the stochastic variation and $M$ which forces the generated samples to respect the given channel distribution.

In Figure 15, we present more results of the 3D models across different layers. The first two columns show the two layers of the $4 \times 4 \times 2$ 3D map which describes the target channel proportions for the $64 \times 64 \times 32$ images across the 3 dimensions. The remaining columns show generated 2D slices at different layers, namely the 1st, 8th, 16th and 24th layers. These results demonstrate that the 3D generated samples are spatially-correlated with the target maps in the 3 dimensions. We note here that although we show results as 2D slices, the 3D images are generated by a single-pass to the generator.

B Quantifying Differences Between Generated Samples with the Same Conditioning Maps

In this section, we present additional results for the braided river masks generated with the same condition maps but with different latent vectors and show the stochastic differences between the samples using standard deviation maps. As shown in Figure 16, the rightmost column shows standard deviation map which represents the stochastic variation around the mean value for each pixel. Small values mean low stochasticity (i.e., it is more likely for the pixel to be either a channel or background facies).

C Flow Simulation Results on Training and Generated Samples

In this section, we present results of a flow simulation problem on the training samples showed on left side of Figure 3 and the corresponding generated samples showed on the top row in Figure 4. We consider the problem of a uniform flow where water is injected in order to displace contaminate in a subsurface reservoir. Flow is injected at the left side boundary and produced from the right side boundary and no-flow boundary conditions are imposed on the top and bottom sides. The problem formulation and settings are identical to those presented in Chan & Elsheikh (2017).

We performed a total of 4000 flow simulations, 2000 corresponding to the training samples and 2000 simulations on the GANs generated samples. Flow statistics of the saturation map at $t = 0.5$ PVI are shown in Figure 17 for the real and generated samples. As shown, statistics from generated realizations
Figure 13: Unconditional generated 3D samples as 2D slices: (a) layers of xy-plane (b) layers of xz-plane (c) layers of yz-plane.
Figure 14: Conditional generated 3D samples. In each row, the first column shows the 3D target map and the remaining columns show the generated 3D realization using different random $z$ and the same $M$. 
Figure 15: Conditional generated 3D samples with target 3D maps: the first two columns constitute the two layers of the $4 \times 4 \times 2$ target maps and the remaining columns show the corresponding generated 3D sample at different layers, namely the 1st, 8th, 16th and 24th layers.
Figure 16: First column shows the target maps, the second, third and forth columns show generated non-stationary realizations using the target maps, the fifth columns shows the mean per pixel map and the standard deviation maps are shown in the rightmost column.

are very similar to the statistics from the training samples. Saturation histograms calculated at the point where the saturation has the highest variance are shown in Figure 18, where the two histograms from the training and generated samples match very well.

Production curves statistics are shown in Figure 19, where we calculated the mean and the variance of the production curves at different times. We have also plotted the histogram of the water breakthrough time (i.e., the time where the injected clean water level reaches the production well with a 1% threshold). As we can see, the calculated statistics on the generated realizations showed very good agreement with the ones on the training samples which reflect the capabilities of the GANs models.
Figure 17: Saturation statistics of a uniform flow on training and generated samples at $t = 0.5$ PVI.

Figure 18: Saturation histogram at a fixed point in the domain where the saturation has the highest variance.
Figure 19: Production statistics of a uniform flow on training and generated samples at. (a) Mean of production curves (b) Variance of production curves (c) Histogram of water breakthrough times (expressed in pore volume injected).