Research Article

ENERGY EXPLORATION & EXPLOITATION

Energy Exploration & Exploitation
2020, Vol. 38(6) 2558–2578
© The Author(s) 2020
DOI: 10.1177/0144598720937524
journals.sagepub.com/home/eea
SAGE

Conditioning well data to rule-based lobe model by machine learning with a generative adversarial network

Honggeun Jo[1], Javier E Santos[1] and Michael J Pyrcz[1,2,3]

Abstract

Rule-based reservoir modeling methods integrate geological depositional process concepts to generate reservoir models that capture realistic geologic features for improved subsurface predictions and uncertainty models to support development decision making. However, the robust and direct conditioning of these models to subsurface data, such as well logs, core descriptions, and seismic inversions and interpretations, remains as an obstacle for the broad application as a standard subsurface modeling technology. We implement a machine learning-based method for fast and flexible data conditioning of rule-based models. This study builds on a rule-based modeling method for deep-water lobe reservoirs. The model has three geological inputs: (1) the depositional element geometry, (2) the compositional exponent for element stacking pattern, and (3) the distribution of petrophysical properties with hierarchical trends conformable to the surfaces. A deep learning-based workflow is proposed for robust and non-iterative data conditioning. First, a generative adversarial network learns salient geometric features from the ensemble of the training rule-based models. Then, a new rule-based model is generated and a mask is applied to remove the model near local data along the well trajectories. Last, semantic image inpainting restores the mask with the optimum generative adversarial network realization that is consistent with both local data and the surrounding model. For the deep-water lobe example, the generative adversarial network learns the primary geological spatial features to generate reservoir realizations that reproduce hierarchical trend as well as the surface geometries and stacking

[1]Hildebrand Department of Petroleum and Geosystems Engineering, The University of Texas at Austin, Austin, USA
[2]Jackson School of Geosciences, The University of Texas at Austin, Austin, USA
[3]Bureau of Economic Geology, Jackson School of Geosciences, The University of Texas at Austin, Austin, USA

Corresponding author:
Honggeun Jo, University of Texas at Austin Cockrell School of Engineering, 200 E. Dean Keeton, Stop C0300, Austin, TX 78712-0284, USA.
Email: honggeun.jo@utexas.edu

pattern. Moreover, the trained generative adversarial network explores the latent reservoir manifold and identifies the ensemble of models to represent an uncertainty model. Semantic image inpainting determines the optimum replacement for the near-data mask that is consistent with the local data and the rest of the model. This work results in subsurface models that accurately reproduce reservoir heterogeneity, continuity, and spatial distribution of petrophysical parameters while honoring the local well data constraints.

Keywords

Rule-based model, data conditioning, geostatistics, deep learning, generative adversarial network

Introduction

Geological heterogeneity characterization is one of the key processes to understand subsurface rock and fluid systems, essential in the fields of groundwater management such as CO_2 sequestration, water, oil and gas production, mineral mining, and foundation and excavation design. Reliable subsurface models are crucial for subsurface management decision making (Pyrcz and Deutsch, 2014). In some subsurface systems, rock heterogeneity is complex, and response variables (e.g. produced fluid flow rates and available volumes for sequestering CO_2) can be highly sensitive to heterogeneity. Complex subsurface heterogeneity is defined as a non-linear, non-Gaussian, multivariate, and multiscale spatial property distribution of rock and fluid features caused by convoluted geological deposition, preservation, and in-situ alteration processes (Mariethoz and Caers, 2014; Pyrcz and Deutsch, 2014). Standard geostatistical approaches are a limited set of stationary spatial continuity statistics such as variogram model parameters, training images conditional probabilities, or geometric object parameter distributions. As such, they are limited in their ability to characterize complicated subsurface heterogeneity (Pyrcz and Deutsch, 2014). In other words, conventional geostatistical workflows often fail to integrate information such as geological concepts, depositional settings, or structures of stacking elements (Michael et al., 2010; Pyrcz et al., 2005; Pyrcz and Deutsch, 2014).

Rule-based modeling has been suggested as an alternative to better integrate geological information in subsurface models; therefore, reproduce salient complex subsurface heterogeneity. Rule-based models use depositional process-informed rules in a temporal sequence to update topographic surfaces and preserve sediment units. In recent work, rule-based modeling has been referred by various names including surface-based (Bertoncello et al., 2013; Pyrcz et al., 2005), event-based (Pyrcz and Strebelle, 2006), hybrid (Michael et al., 2010), and rule-based modeling (Jo and Pyrcz, 2019; Pyrcz et al., 2015). These methods enable us to build hierarchical reservoir models that integrate geological rules, within grid-free frameworks for flexible reservoir representation (Hassanpour et al., 2013; Pyrcz and Deutsch, 2014). Consequently, rule-based models can preserve complex subsurface heterogeneity that is not readily achievable with other conventional geostatistical methods (e.g. variogram-based, object-based, and multipoint-based).

While conventional geostatistical methods are limited to the reproduction of features characterized by input statistics, constrained by local conditioning, and assumptions such as multiGaussianity, rule-based models add the ability to capture heterogeneity efficiently

Table 1. Comparison of strengths and weaknesses of the reservoir models.

Type of model	*Strengths*	*Weaknesses*
Variogram-based	Direct data input and straightforward inference from well and analog data Highly flexible to condition to well data	Limited to interpret available geological information such as connectivity, structures, and stacking pattern Maximum entropy beyond correlation length, resulting in anticipating uniform waterfront
Multipoint-based	Intuitive spatial continuity model using training image More complicated reservoir heterogeneity can be obtained than variogram-based models Highly flexible to condition to well data	Limited integration of geological information from the training image with reproduction of stationary connectivity, geometry, and stacking patterns. Incapable to exhibit the variability and uncertainty of geological inference, especially in 3D
Object-based	Geometric parametrization can use analog data directly	Limited integration of geological information from simple parametric geometry in absence of details in stacking patterns Conditioning to well observations only in relatively sparse well data settings.
Rule-based	Direct integration of process mimicking rules along with geometric Parametrization More realistic reservoir heterogeneity uncertainty models Provide insight into heterogeneity structure that was not known in advance	Limited ability to condition to observations Input parameters are not directly available from observed structure Large computational cost is required for more complicated heterogeneity

Source: Revised from Pyrcz et al. (2015).

through the interactions of the model rules. This phenomenon is defined as *emergent heterogeneity* by Pyrcz et al. (2015). In addition, emergent heterogeneity has helped understand geological deposition processes such as the preserved geometry of lateral elements in a fluvial system from FLUMY (Cojan et al., 2005) and ALLUVSIM (Pyrcz et al., 2009). Emergent heterogeneity in rule-based models provides realistic subsurface heterogeneity without a tedious, explicit parametric representation of the result (Pyrcz et al., 2015). Table 1 shows a comparison of the strengths and weaknesses of each reservoir modeling method.

The rule-based models are widely emerging in applications for both of deep-water (Deptuck et al., 2008; Pyrcz et al., 2005, 2015; Sullivan et al., 2000; Zhang et al., 2009) and fluvial clastic reservoirs (Cojan et al., 2005; Pyrcz, 2004; Pyrcz et al., 2009). There is, however, a remaining obstacle to its broad application, robust conditioning to local well data (Pyrcz et al., 2015). There have been multiple studies to resolve the local data conditioning of rule-based models (Bertoncello et al., 2013; Michael et al., 2010; Pyrcz, 2004). Pyrcz (2004) iterates the simulation of the next depositional event and selects the result that minimized the mismatch with the local data and then adds a conditional stochastic residual for precisely match the local data. Michael et al. (2010) propose the hybrid method with the combination of rule-based modeling methods and conventional geostatistical methods to solve the local conditioning problem, but this may require large computational time to find the optimum surface. Bertoncello et al. (2013) select the most sensitive model parameters of the rule-based model by sensitivity analysis and apply the sequential optimization scheme to minimize data mismatch. However, none of these methods are sufficiently robust and straightforward for broad application to subsurface modeling projects. More specifically, these methods are limited to sparse well data cases, with their performance degraded as the data density increases (Pyrcz et al., 2015).

Generative adversarial network (GAN) is a framework for training generative models through an adversarial learning process, where two neural networks (i.e. a generative model and a discriminative model) are trained concurrently (Goodfellow et al., 2014). GAN poses an advantage in generating realistic images with high resolution in comparison to other methods. Radford et al. (2015) add the convolution layer in GAN to capture the geometric configuration of images more efficiently. Wu et al. (2016) expand the application of GAN to 3D objects and demonstrate learning a 3D structure with a 3D kernel filter. Moreover, Yeh et al. (2016) suggest semantic image inpainting algorithm with GAN that restores the missing content in the 2D image by conditioning the available data. Given a trained GAN, the missing content is estimated through navigating the latent image manifold and minimizing semantic loss that is the sum of perceptual and conceptual loss.

GAN and semantic image inpainting have been applied to geoscience and reservoir engineering for reservoir modeling. Chan and Elsheikh (2017) represent a geological model with latent parameters using GAN and Laloy et al. (2018) apply GAN to train with geostatistical training images for geostatistical inversion. Moreover, Zhang et al. (2019) apply GAN to train multiple-point statistics (MPS) models and use semantic image inpainting for data conditioning. However, previous investigations are limited to conventional geostatistical methods in acquiring training data. As such, conditioning data for those models do not necessarily require complicated, deep learning algorithms. For example, data conditioning for MPS (even 3D model) is readily available in Geostatistical Software Library by Deutsch and Journel (1998). Moreover, most of the previous methods mainly investigated the categorical facies models that do not contain realistic petrophysical properties.

Our contribution is applying GAN and semantic image inpainting to conditional rule-based models with local data conditioning data. The GAN is trained with multiple rule-based models to extract geological features and to generate latent reservoir manifolds representing a space of possible models. After training GAN, rule-based model volumes near the well locations are removed; these are referred to as masked volumes. Local well data are assigned at the well locations and the remainder of the mask volumes are filled with the optimum GAN's realization based on consistency with the surrounding volumes and precise well data information while retaining the complicated subsurface features.

Method

In this section, we introduce our rule-based model for the deep-water lobe system (greater detail on this model is available in Jo and Pyrcz, 2019). An alternative approach is available from Cojan et al. (2005). To support the broad use of this model, we introduce GAN and semantic image inpainting for robust well data conditioning.

Aggradational rule-based modeling

The applied rule-based model is designed for deep-water lobe depositional systems. A lobe complex consists of multiple lobe complexes, and each lobe complex consists of several lobe elements (Abreu et al., 2003; Deptuck et al., 2008; Sullivan et al., 2004; Zhang et al., 2016). Figure 1 shows a block diagram of the deep-water lobe system, as well as its hierarchical structure. A hierarchical structure implies a scale-dependent framework that subdivides deposits based on the distinct changes in characteristics such as geometry, orientation, and trend. The lobe geometry and depositional stacking patterns are controlled by various processes, including turbidity flow properties, frequency of flows, gradient change, and seafloor morphology (Deptuck et al., 2008). Moreover, the lobe geometry and stacking pattern are suspected of having a significant impact on subsurface flow due to the unique superposition of low permeability lobe margin and high permeability inner lobe in the resulting unique stacking pattern (Mutti and Normark, 1987).

Our rule-based model is controlled by three geological inputs to fit a range of hierarchical structures of naturally occurring systems, including lobe element geometry, compositional exponent to constrain the stacking pattern, and hierarchical trend model that controls the distribution of petrophysical properties within the lobe complex (Pyrcz et al., 2005).

The typical aspect ratio of a lobe element (ratio of length to width) varies from one to three (Deptuck et al., 2008; Zhang et al., 2016). In this study, we assume the isotropic lobe (with aspect ratio equal to one) to assess a generalized outcome by removing the effect of sediment flow direction. Figure 2 shows the geometry of individual lobe geometry, similar to Xie et al. (2001). Therefore, we can parameterize each lobe element with horizontal radius and maximum thickness.

The rule-based model starts with a predefined initial bathymetry. For each step of the simulation, a lobe element is stacked on the resultant surface of the previous iteration. The location of the lobe element is drawn from a probability map that is inversely proportional to the topography of the current compositional surface. The compositional exponent controls the tendency of a compensation stacking pattern by controlling the probability map. If the compositional exponent is zero, the rule-based model produces a random lobe element

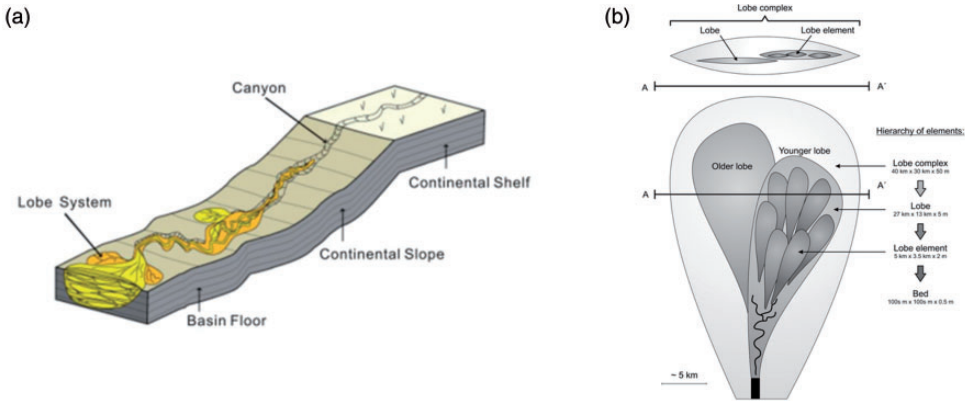

Figure 1. Deep-water aggradational lobe system. (a) Block diagram of lobe system (Bouma and Stone, 2000) and (b) its hierarchical structure (Groenenberg et al., 2010). Sediment flows from right to left in the block diagram (a), and it flows from bottom to top in the hierarchical diagram (b).

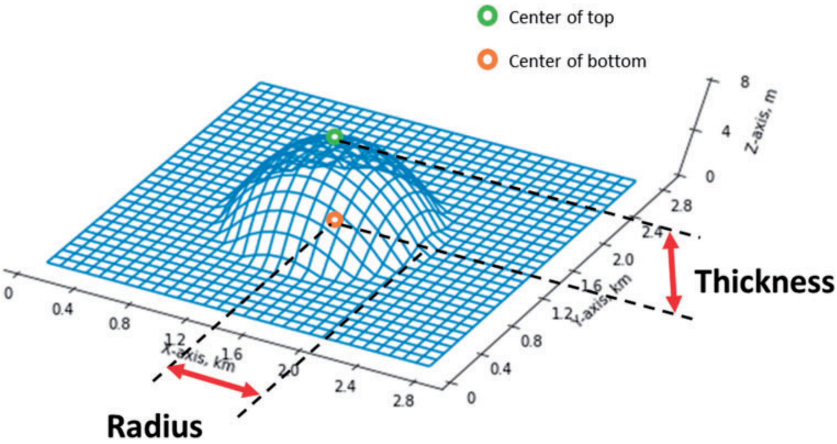

Figure 2. Lobe element geometry. In this study, paraboloid lobe geometry is assumed with 750 m radius and 4 m thickness. Source: Modified from Xie et al. (2001).

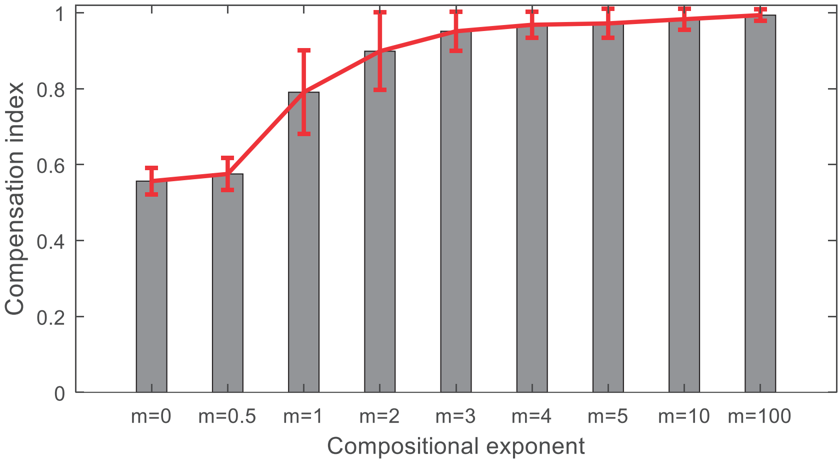

Figure 3. Compensation index with different compositional exponent (Jo and Pyrcz, 2019). Compensation index denotes the degree of compensational stacking, the tendency of sediment to preferentially deposit in topographic low (Straub et al., 2009). The compensation index is 0.5 for random stacking and 1.0 for perfect compensational stacking pattern.

stacking pattern (Straub et al., 2009). Whereas, if the compositional exponent is above five, a perfect compensational pattern will be observed among the lobe elements in the model (see Figure 3).

Once the lobe element is deposited on the current compositional surface, the topography is updated. Consequently, this modifies the probability map for the lobe location in the next iteration. The model sequentially adds lobe elements until the reservoir volume is filled. We then assign rock properties (e.g. porosity and permeability) based on hierarchical trends anchored to the compositional surfaces (Pyrcz et al., 2005). For example, the center of lobe

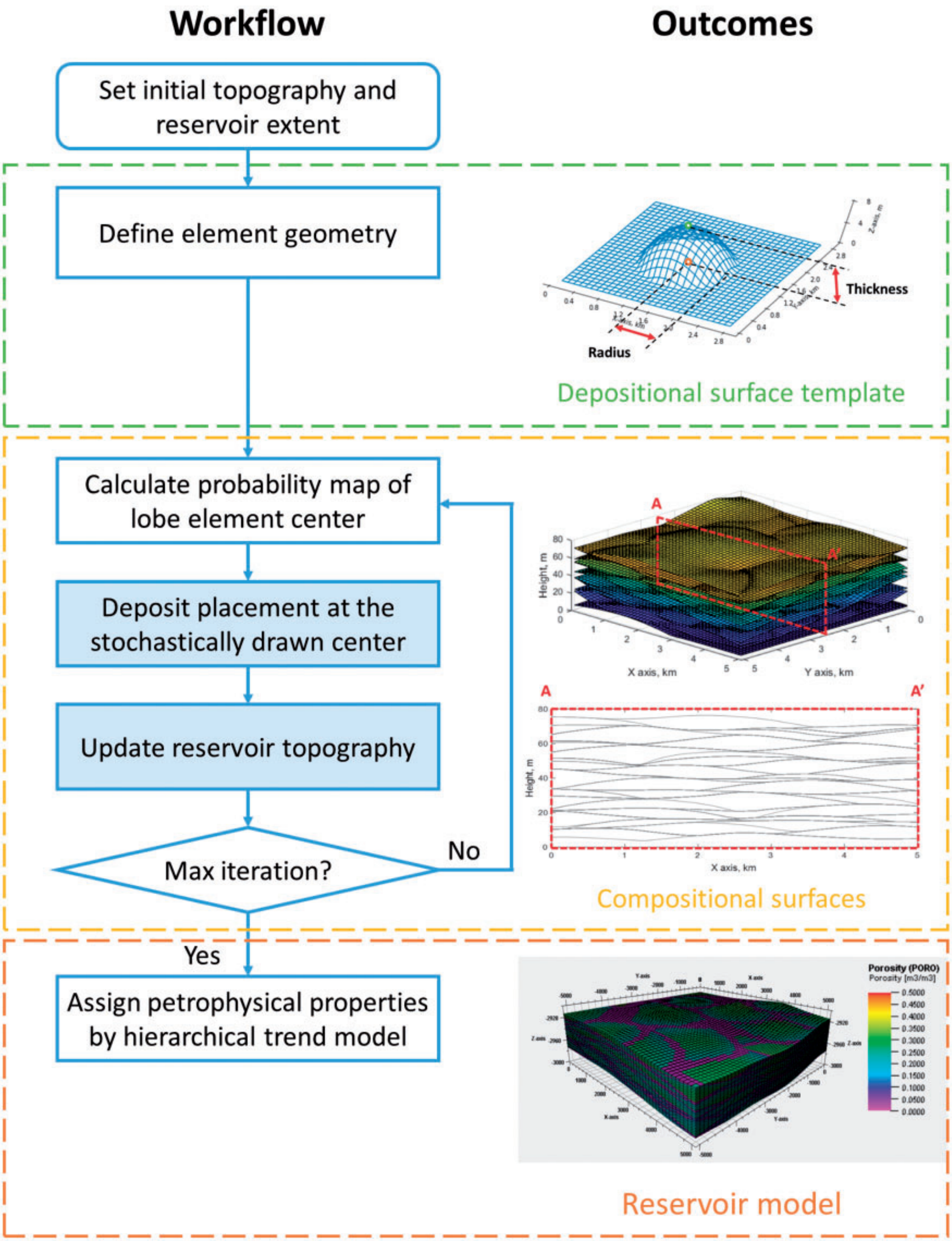
Workflow
Outcomes
Set initial topography and reservoir extent
Define element geometry
Thickness
Radius
Depositional surface template
Calculate probability map of lobe element center
Deposit placement at the stochastically drawn center
Update reservoir topography
Max iteration?
No
Yes
A
A'
Height, m
X axis, km
Y axis, km
Compositional surfaces
Assign petrophysical properties by hierarchical trend model
Porosity (PORO)
Porosity [m3/m3]
Reservoir model

Figure 4. Flow chart of rule-based model and corresponding outcomes at each step (Jo and Pyrcz, 2019).

element tends to have sandy rocks, whereas their surface consists of shale drapes (i.e. fining-up trend), and the coarsening-up trend at the lobe complex scale due to system progradation. We use a hierarchical trend model to consider these two trends in different scales (Pyrcz, 2004). Figure 4 represents the entire workflow and outcomes from each step of the rule-based model.

We set the lobe element stacking pattern to perfect compensation of lobe elements of 750 m radius and 4 m maximum thickness. The hierarchical trend is set to reduce porosity toward the margins and tops of the lobe elements, following deep-water lobe complex observations from western Gulf of Mexico (Sullivan et al., 2004), east Corsica (Deptuck et al., 2008), and offshore West Africa (Zhang et al., 2016). See Figure 5 for an example realization of this rule-based model.

GAN

GAN consists of two neural network models: discriminator and generator (Denton et al., 2015; Goodfellow et al., 2014). Radford et al. (2015) improve the performance of GAN by adding a convolutional neural network and suggesting a better hyperparameter set. The generative model maps a latent random vector (usually consists of 100 elements) to a 3D image (in our case, it is a rule-based lobe). Note that the latent random variables are not directly observed, but are inferred through the trained generative model. We can, therefore, represent high-dimensional geometric features in the rule-based model by low-dimensional latent random variables. The discriminative model maps an input image, from either the generator or training image set, to the likelihood that the image is real. Both generative and discriminative models are trained adversarially to optimize the following loss function

$$\min_{\boldsymbol{G}} \max_{\boldsymbol{D}} V(G,\ D) = E_{\boldsymbol{h}}\left[\log(D(\boldsymbol{h}))\right] + E_z[\log(1 - D(G(z))] \tag{1}$$

where G and D are generative and discriminative models, respectively. The $\boldsymbol{h}$ denotes the sample from the training image set, and z stands for latent random variables from the latent space.

Training GAN is like a competition between a forger and a detective. At the beginning of the training, the forger may generate random noises that allow the detective to discriminate fake images easily. However, based on the feedback of the loss function, the forger grows more sophisticated to generate increasingly realistic images (more similar to the training set) than the previous iterations. In the following iteration, the detective may incorrectly identify a fake image, thereby reinforcing its performance. In such a way, both detective and forger are trained alternatively. This training process continues until the GAN reaches an equilibrium, where all salient information in the training data is used to build realistic synthetic models.

Figure 6 shows the schematic diagram of training GAN. The discriminative model maps a 3D input image to a likelihood that the input is sampled from the rule-based models. The input 3D image is from either the storage of rule-based models or the realization from the generator. The generative model projects a latent random vector that consists of 100 elements to a 3D image. The latent random variables are sampled from the standard Gaussian distribution (with a mean of zero and variance of one). After training GAN, the latent random variables enable us to navigate the range of possible models. In other words, we can generate different realizations by inputting different latent random vectors, and each realization has the necessary, realistic features of training data. To train the GAN, 40,000 rule-based models are used over 30,000 epochs with 40 batch sizes, requiring about 8 h on a desktop computer with an Nvidia M6000 GPU. For deployment, the trained GAN generates approximately 1000 realizations per second using the GPU.

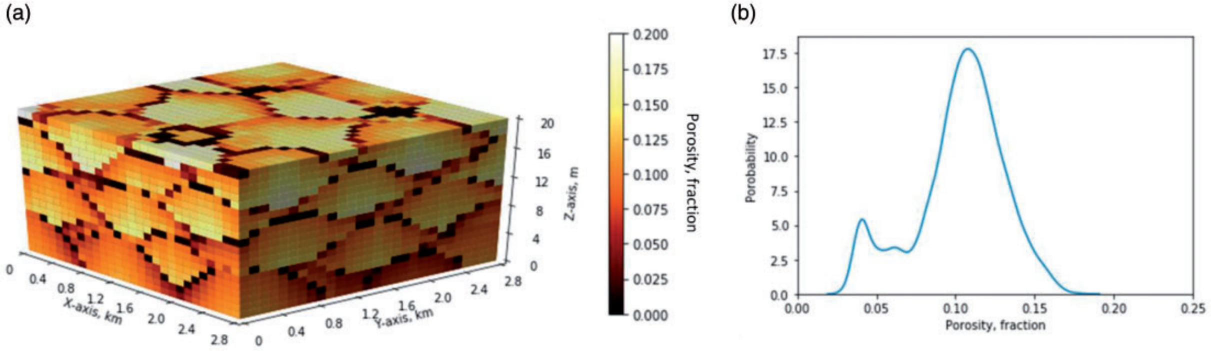

Figure 5. Example of rule-based model. (a) 3D porosity model and (b) porosity distribution of rule-based model.

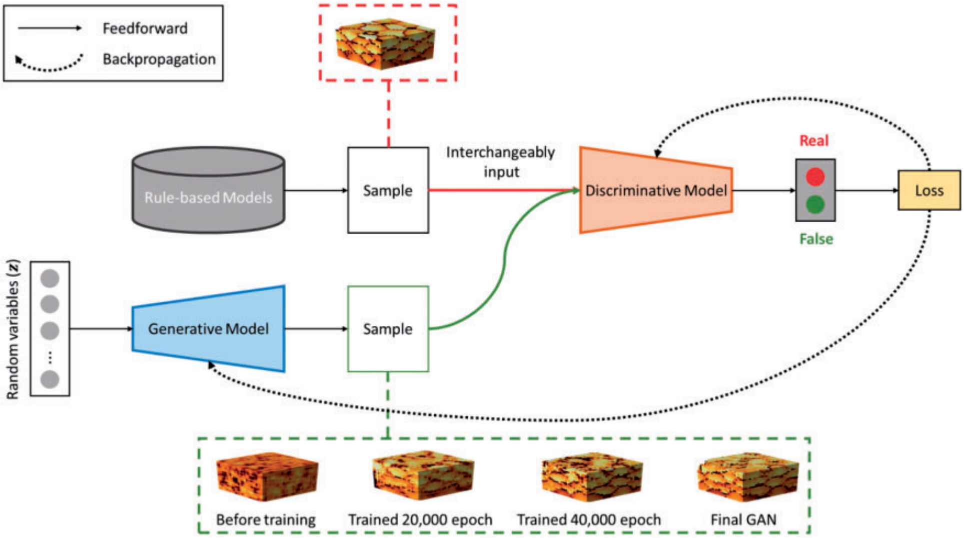

Figure 6. Schematic diagram of GAN for rule-based model. Red dashed box indicates an example of rule-based model and green dashed box indicates realizations of the generative model at each training epoch. The more GAN is trained, the more sophisticated realizations are generated.

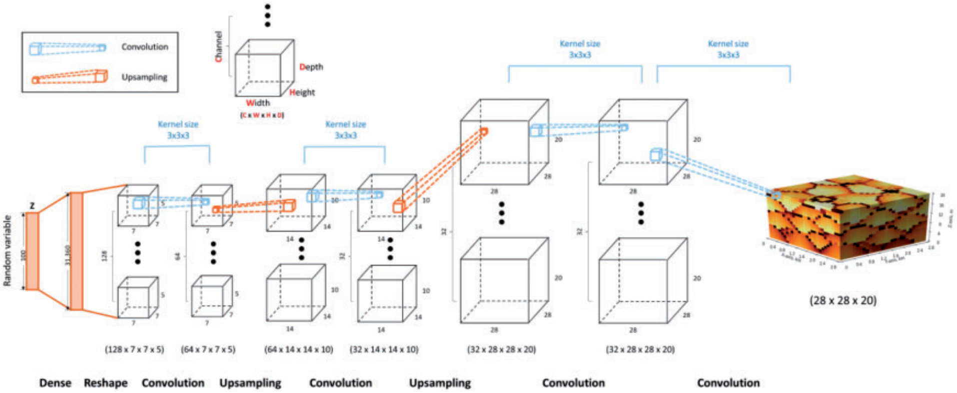

Figure 7. Structure of generative model.

Figure 7 shows the neural network structure of the generative model. The generative model maps (fully connected) a vector of 100 elements in the input layer to a hidden layer of 31,360 elements. After reshaping the hidden layer, a feature map of $7 \times 7 \times 5$ (i.e. width × length × heightt) size with 128 channels is built. For the remainder of the hidden layers, the size of the feature map increases while the number of channels decreases. At the last feature map, the generative model outputs a 3D model of $28 \times 28 \times 20$ size that is identical to the size of the rule-based model. If the extent of a subsurface model is changed, the dimensions of feature maps and the number of channels must be updated accordingly. The details of the discriminative model are provided in Appendix 1.

Well data conditioning through semantic image inpainting

Image inpainting refers to the algorithms to restore lost or corrupt parts of the image data. Yeh et al. (2016) consider two pieces of information for the inpainting task: contextual and perceptual information. For a corrupted image with missing regions, the missing voxels (or pixels in a 2D image) should be inferred based on the neighboring voxels (or pixels), or conceptual information, and the restored parts should be "realistic" by themselves so that they have similar features to the training data or perceptual information. Semantic inpainting integrates both conceptual and perceptual information to determine the missing parts of an image by minimizing the following semantic loss function (Yeh et al., 2016)

$$L_{\text{semantic}} \equiv L_{conceptual}(z) + \lambda \cdot L_{\text{perceptual}}(\mathbf{z}) \tag{2}$$

$$L_{conceptual} = \|\boldsymbol{M} \odot \mathrm{G}(z) - \boldsymbol{M} \odot \boldsymbol{Y}\|_1 \tag{3}$$

$$L_{\text{perceptual}} = \log(1 - \mathrm{D}(\mathrm{G}(\mathbf{z}))) \tag{4}$$

where $\boldsymbol{M}$ is the mask matrix that consists of elements 0 for missing portions and 1 for the rest, $\odot$ denotes the element-wise multiplication, and λ is a hyperparameter to control the trade-off between perceptual and conceptual loss. For this work, we set λ as 0.001 following Yeh et al. (2016). Note that λ is selected from trial and error, a typical process for hyperparameter tuning. We also tune λ and select 0.001 through ocular inspection. The $\boldsymbol{Y}$ stands for the original, corrupted image and $\mathbf{z}$ is the latent random variables to map to a new image, $\mathrm{G}(\mathbf{z})$. Remember that $\mathbf{z}$ is not directly measured but are inferred through the trained generator, $\mathrm{G}(\mathbf{z})$.

With the trained GAN and semantic image inpainting scheme, this study proposes a novel workflow to solve well data conditioning problem of the rule-based model. After generating a rule-based model, our proposed method places voids in the adjacent volumes of well locations, which should be as large as maximum spatial correlation length to provide sufficient freedom to match well data, and conceptual and perceptual information. Well data (from either well logs or core descriptions), assumed to be error-free, are assigned to the center of voids along the well trajectories. Once the semantic loss is defined, the latent vector is optimized by gradient descent to find the $\mathbf{z}_{\mathbf{optimum}}$ that will generate the best GAN's realization, consistent with both well data and the surrounding volumes. The $\mathrm{G}(\mathbf{z}_{\mathbf{optimum}})$ restores the masked volumes of the rule-based model to reconstruct the entire reservoir model. Figure 8 represents the overall workflow for well data conditioning in the rule-based model.

Result

Realizations by GAN

We generate 40,000 rule-based models and utilize them as training data for the GAN. The rule-based models satisfy a perfect compensational stacking pattern and consist of isotropic,

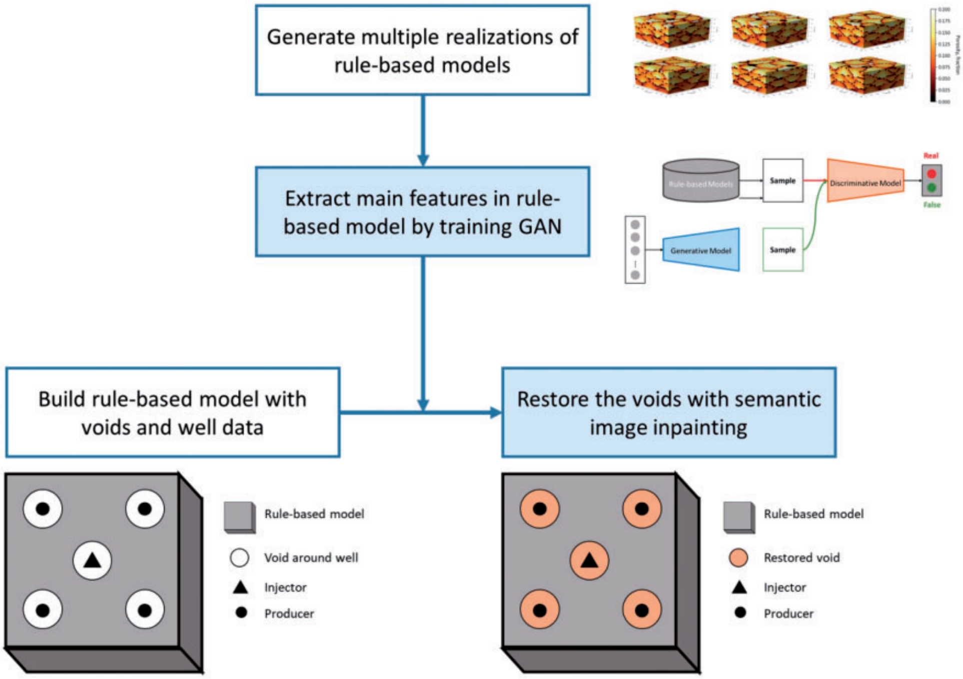
Generate multiple realizations of rule-based models
Extract main features in rule-based model by training GAN
Rule-based Models
Sample
Discriminative Model
Real
False
Generative Model
Sample
Build rule-based model with voids and well data
Restore the voids with semantic image inpainting
Rule-based model
Void around well
Injector
Producer
Rule-based model
Restored void
Injector
Producer

Figure 8. Workflow for well data conditioning. GAN: generative adversarial network.

uniform lobe geometry. The size of each training data is 28 × 28 × 20 grid blocks in x-, y-, and z-direction, respectively, where each grid block is 100 × 100 × 0.5 m. Considering the lobe radius is 750 m, and the maximum lobe thickness is 4 m, the grid block dimension is suitable to describe details of geometric features in the rule-based models. Figure 9 shows visual comparisons between training rule-based models and realizations from GAN in 3D. Figure 9(a) presents five models out of 40,000 training models (i.e. rule-based models), where we can observe compositional lobe boundaries, as well as hierarchical trends in porosity. The complex subsurface heterogeneity, lobe boundaries and those hierarchical trends, from the training models are reproduced in the GAN realizations, as shown in Figure 9(b). Table 2 shows the dimensionality comparison between original rule-based model and GAN. GAN diminishes the original dimensionality by 99.4%.

We check the GAN models' ability to learn the hierarchical trends of the training models by visualizing the cross-sectional views of GAN realization. Figure 10(a) is a 3D view of GAN realization, and Figure 10(b) is a fence diagram for the same model. Distinct shale drapes of lobe are observed in internal vertical layers in Figure 10(b). Moreover, these lobe element boundaries have realistic geometries and continuity and the hierarchical trends are identical to the training models.

Since the GAN takes a vector of continuous latent random variables (of 100 elements) to generate 3D images, a perturbation on these elements will generate a continuous set of subsurface realizations. Figure 11 shows the gradual change of 3D images between

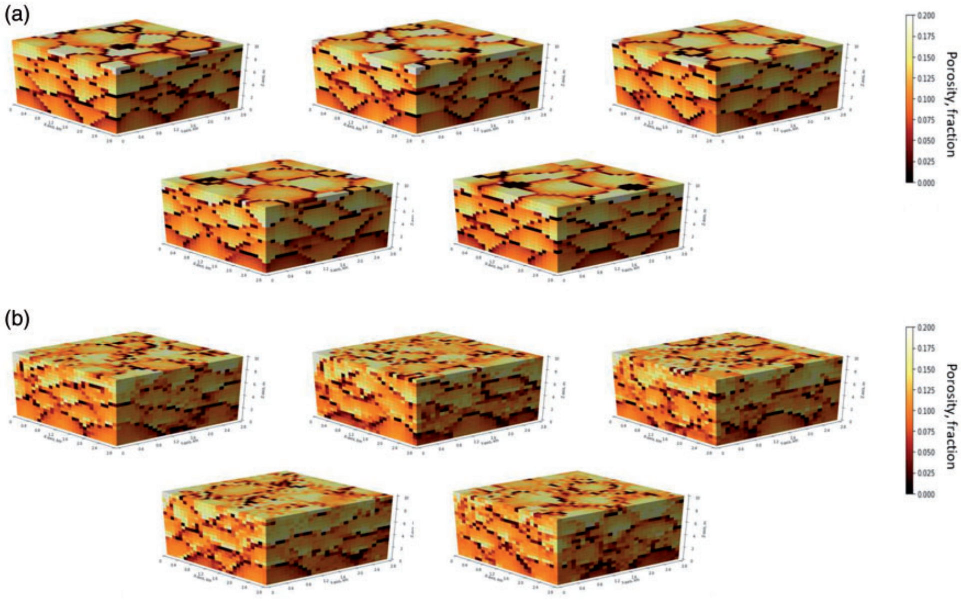

Figure 9. Comparison between rule-based model and unconditional realizations of GAN. (a) 3D rule-based model for lobe system (5 out of 40,000) and (b) unconditional realizations of the trained GAN.

Table 2. Dimensionality comparison between original rule-based model and GAN.

Method	*Dimensionality*	*Ratio (%)*
Rule-based model (original)	15,680	100
GAN-generator	100	0.63

GAN: generative adversarial network.

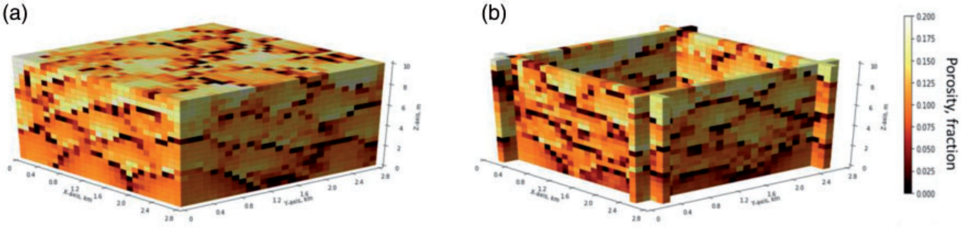

Figure 10. Cross-section of GAN realization (a) 3D view and (b) fence diagram of GAN realization.

realizations of $G(z_1)$ and $G(z_2)$. GAN realizations ($G(z)$) are continuously changed by the continuous change of latent random variables z enabling us to explore the latent reservoir manifold and efficiently apply a gradient descent scheme to the optimization problem of minimizing mismatch at the well locations.

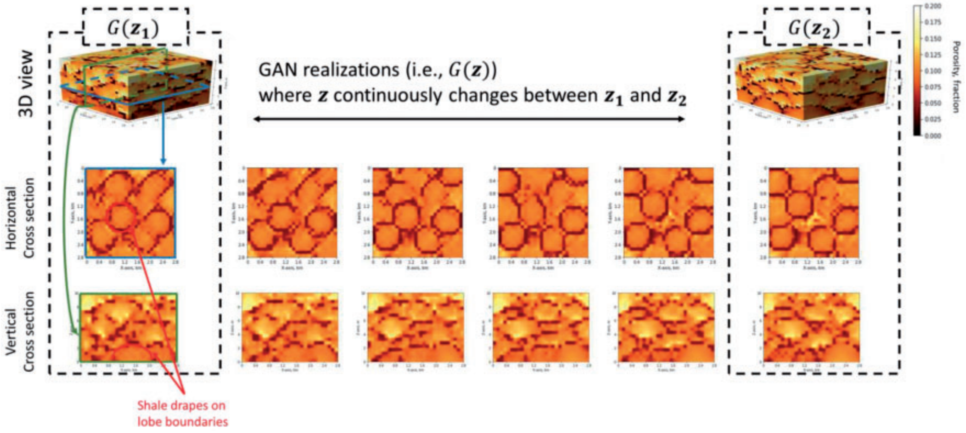

Figure 11. Continuous change of GAN realizations between two random variables z_1 and z_2. GAN: generative adversarial network.

Validation of GAN realizations

On top of visually inspecting GAN realizations, we measure the difference of porosity distribution between rule-based models and GAN realizations by projecting an ensemble of them into 2D space. The rule-based model and GAN realization consist of 15,680 grid cells, so those models represent 15,680 dimensions natively. We implement a discrete cosine transform (DCT) to compress the dimension of the model from 15,680 to 100. Subsequently, we reduce their dimension from 100 to 2 by multidimensional scaling (MDS). In other words, both rule-based models and GAN realizations are represented in 2D model dissimilarity space by using DCT and MDS consecutively.

Figure 12 shows a comparison between rule-based models and GAN realizations in 2D space. The red points are the projections of 100 GAN realizations, and the blue points are the projections of 100 rule-based models. The x- and y-axis of each 2D plot do not have any physical meaning, but they preserve approximately the pairwise difference between all of the models. Each 2D plot shows how much the rule-based model and GAN realization are similar or different from each other at the corresponding training step, along with a representative GAN and rule-based.

If the GAN is successful at extracting high dimensional geometric features of the rule-based model and puts those features in its realization, the GAN realizations should map over the same extent as the training models. At the beginning of training, GAN only generates random noise so that red points show maximum dissimilarity from blue points. As we step through training the GAN, the dissimilarity (or distance) between two clusters reduces. Moreover, after training GAN for 30,000 epochs, two clusters become adjacent to each other. Finally, after training for 40,000 epochs, there is no distinct difference between the two. In addition, the representative GAN realizations illustrate the learning of complex subsurface heterogeneity. We trained GAN for 50,000 epochs in this study.

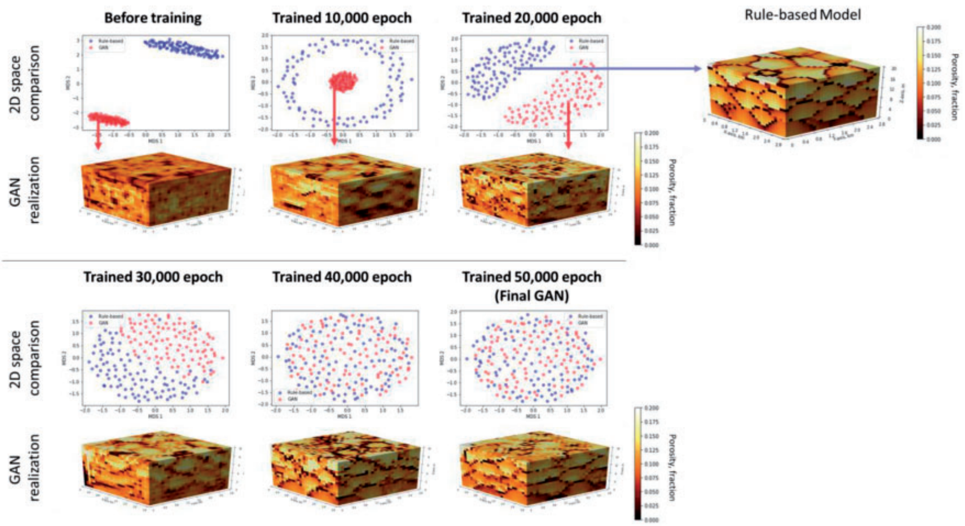

Figure 12. Comparison between rule-based models and GAN realizations in 2D MDS space. Before the training is carried out, the difference between blue and red cluster is maximum, and, as GAN is trained, the distance between two clusters reduces until both rule-based and GAN generated models are indistinguishable in MDS space. GAN: generative adversarial network.

Well data conditioning

After generating a rule-based model, we place the masks near well locations and assign local well data at the center of the masks along the well trajectories. In this demonstration, we assume the inverted five-spot well pattern that locates an injector at the center and surrounding producers in each of the four corners. As such, five wells (i.e. one injector and four producers) exist in a rule-based model, and local well data are acquired in these well locations. The radius of a masked area is assumed to be 1 km, which should be adjusted depending on the maximum correlation lengths and the density of the well data to avoid discontinuities in the resulting model. Figure 13 illustrates the process for mask construction in a rule-based model.

The masked areas should be filled by the GAN's realization, $G(z)$, and the related random variables, z, are determined by minimizing the semantic loss in equation (2). Due to the continuous responses of $G(z)$, gradient descent is available to find the optimum z. After finding $z_{optimum}$, the masked volumes of the rule-based model is replaced by $G(z_{optimum})$. Figure 14(a) shows the 3D view of the reconstructed model. Figure 14(a) and (b) show the incomplete and reconstructed rule-based models, repectively. The proposed method assures the spatial continuity of the reservoir model is preserved in both horizontal and vertical layers. More specifically, the shale drapes of lobe elements are well-connected, and the stacking pattern does not change significantly. In addition to the aforementioned ocular inspection, the probability that the reconstructed model is from the training data set (i.e., real rule-based models) is over 99% when the optimum GAN realization is put into the discriminative model. Therefore, these results verify that GAN and semantic image inpainting provide a solution for well data conditioning of rule-based subsurface models.

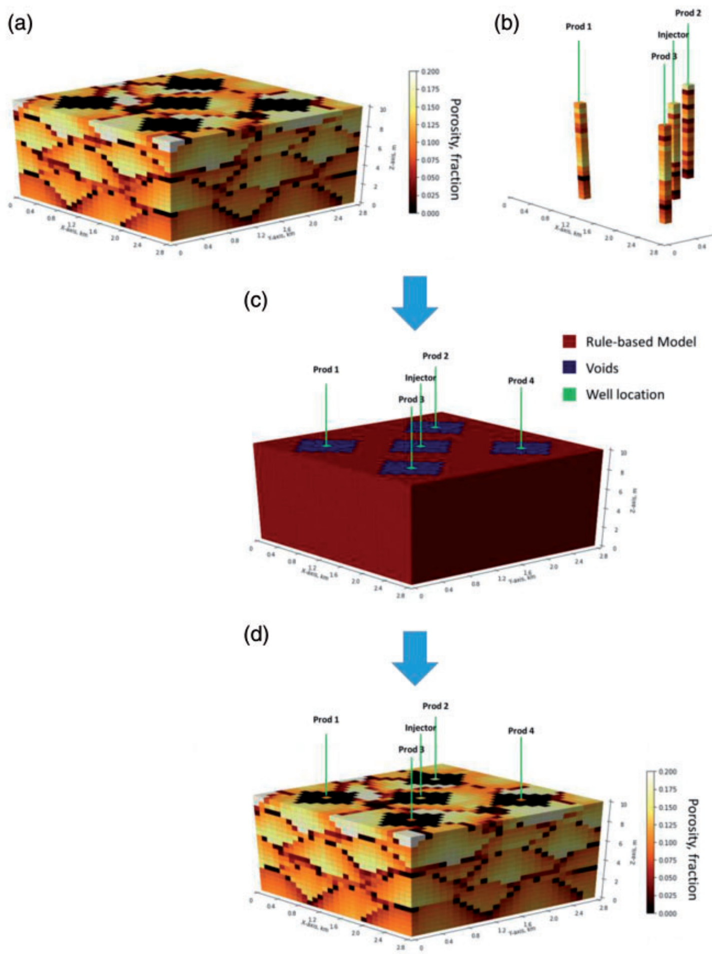

Figure 13. Rule-based model and mask. (a) Rule-based model with mask around well location, (b) well locations and local well data, (c) schematic 3D view of rule-based model with hard data and void, and (d) rule-based model with hard data and void.

Discussion

With a trained GAN, this study demonstrates that GAN can generate realistic 3D lobe reservoirs that conserve geological features in rule-based models. Since realizations of GAN follow a different distribution, cumulative distribution function mapping is required to conserve the original bimodal distribution of reservoir properties. Continuous characteristic of GAN realization enables us to use gradient descent optimization for the best match with well data. Through the semantic image inpainting algorithm, the proposed method successfully solves well conditioning problems for 3D rule-based models allowing for robust use for modeling deep-water lobe complexes. *Rule-based modeling with GAN enables direct geologic information integration as well as spatial data conditioning in a computationally efficient way. As a result, our workflow can reproduce a more realistic subsurface model than the*

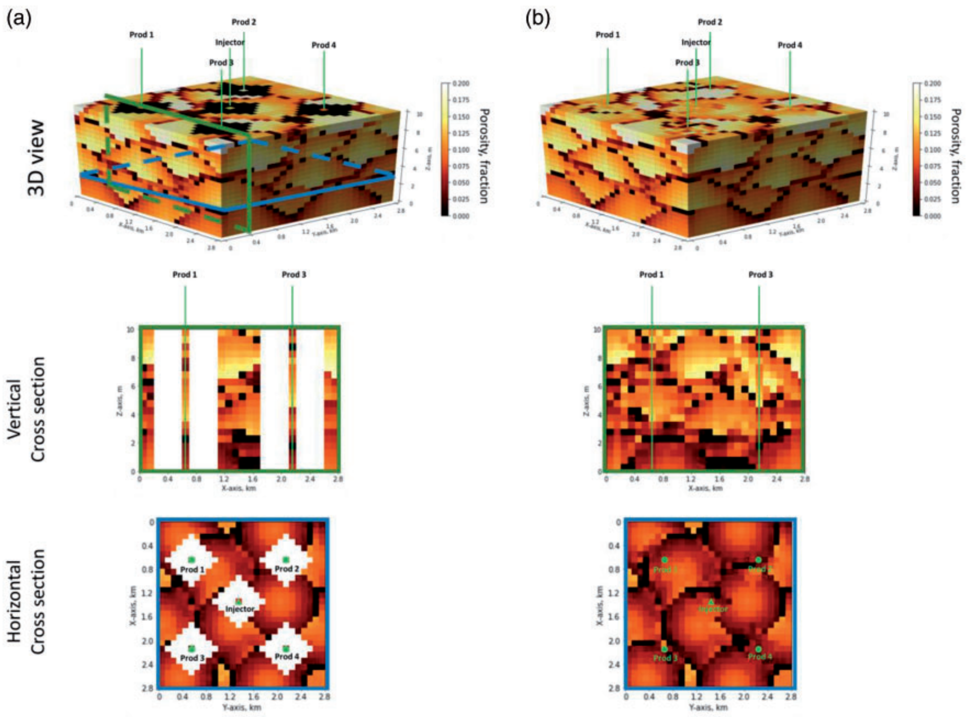

Figure 14. Reconstructed reservoir model and void area. (a) Rule-based model before restoring voids and (b) after restoring voids through semantic image inpainting.

conventional modeling process, such as variogram-, multipoint-, and object-based methods, without a significant increase in computation cost.

Conclusion

In this paper, a deep learning-based data conditioning method is proposed for rule-based models. Compared to existing methods based on conventional geostatistics, the proposed method has more flexibility in preserving complicated subsurface heterogeneity while conditioning to well data for subsurface models with computational efficiency given a trained GAN. This study demonstrates that our generative model can render sharp and realistic 3D reservoir models in an efficient and practical workflow for use in the common practice of subsurface modeling.

Acknowledgements

The authors would like to thank the two reviewers for their critical feedback and their help to improve the quality of the manuscript.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was funded by the DIRECT consortium in the Cockrell School of Engineering, The University of Texas at Austin.

ORCID iD

Honggeun Jo https://orcid.org/0000-0002-1184-0190

References

Abreu V, Sullivan M, Pirmez C, et al. (2003) Lateral accretion packages (LAPs): An important reservoir element in deep water sinuous channels. *Marine and Petroleum Geology* 20(6–8): 631–648.

Bertoncello A, Sun T, Li H, et al. (2013) Conditioning surface-based geological models to well and thickness data. *Mathematical Geosciences* 45(7): 873–893.

Bouma AH and Stone CG (2000) *Fine-Grained Turbidite Systems: AAPG Memoir 72*. Tulsa, OK: AAPG.

Chan S and Elsheikh AH (2017) Parametrization and generation of geological models with generative adversarial networks. *Computational Geosciences* 23: 925–952.

Cojan I, Fouché O, Lopéz S, et al. (2005) Process-based reservoir modelling in the example of meandering channel. In: Leuangthong O and Deutsch CV (eds) Geostatistics Banff 2004. *Quantitative Geology and Geostatistics*. Vol. 14. Dordrecht: Springer.

Denton E, Chintala S, Szlam A, et al. (2015) Deep generative image models using a Laplacian pyramid of adversarial networks. arXiv preprint arXiv:1506.05751.

Deptuck ME, Piper DJ, Savoye B, et al. (2008) Dimensions and architecture of late Pleistocene submarine lobes off the northern margin of East Corsica. *Sedimentology* 55(4): 869–898.

Deutsch CV and Journel A (1998) *GSLIB: Geostatistical Software Library and User's Guide*. 2nd ed. New York: Oxford University Press.

Goodfellow I, Pouget-Abadie J, Mirza M, et al. (2014) Generative adversarial nets. In: *Advances in Neural Information Processing Systems*, pp.2672–2680. Red Hook: Curran Associates, Inc.

Groenenberg RM, Hodgson DM, Prélat A, et al. (2010) Flow–deposit interaction in submarine lobes: Insights from outcrop observations and realizations of a process-based numerical model. *Journal of Sedimentary Research* 80(3): 252–267.

Hassanpour MM, Pyrcz MJ and Deutsch CV (2013) Improved geostatistical models of inclined heterolithic strata for McMurray formation, Alberta, Canada. *AAPG Bulletin* 97(7): 1209–1224.

Jo H and Pyrcz MJ (2019) Robust rule-based aggradational lobe reservoir models. *Natural Resources Research* 29: 1193–1213.

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

Mariethoz G and Caers J (2014) *Multiple-Point Geostatistics*. Chichester: John Wiley & Sons, Ltd.

Michael HA, Li H, Boucher A, et al. (2010) Combining geologic-process models and geostatistics for conditional simulation of 3-D subsurface heterogeneity. *Water Resources Research* 46(5): 1–20.

Mutti E and Normark WR (1987) Comparing examples of modern and ancient turbidite systems: Problems and concepts. In: Leggett JK and Zuffa GG (eds) *Marine Clastic Sedimentology*. Dordrecht: Springer, pp.1–38.

Pyrcz MJ (2004) *Integration of geologic information into geostatistical models*. PhD Thesis, University of Alberta, Canada.

Pyrcz MJ, Boisvert JB and Deutsch CV (2009) ALLUVSIM: A program for event-based stochastic modeling of fluvial depositional systems. *Computers & Geosciences* 35(8): 1671–1685.

Pyrcz MJ, Catuneanu O and Deutsch CV (2005) Stochastic surface-based modeling of turbidite lobes. *AAPG Bulletin* 89(2): 177–191.

Pyrcz MJ and Deutsch CV (2014) *Geostatistical Reservoir Modeling*. 2nd ed. New York: Oxford University Press.

Pyrcz MJ, Sech RP, Covault JA, et al. (2015) Stratigraphic rule-based reservoir modeling. *Bulletin of Canadian Petroleum Geology* 63(4): 287–303.

Pyrcz MJ and Strebelle S (2006) Event-based geostatistical modeling: Application to deep-water systems. In: Slatt RM, Rosen NC, Bowman M, et al. (eds) *Reservoir Characterization: Integrating Technology and Business Practices: 26th Annual*. Tulsa, OK: SEPM Society for Sedimentary Geology, pp.893–922.

Radford A, Metz L and Chintala S (2015) Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434.

Straub KM, Paola C, Mohrig D, et al. (2009) Compensational stacking of channelized sedimentary deposits. *Journal of Sedimentary Research* 79: 673–688.

Santos JE, Xu D, Jo H, et al. (2020) PoreFlow-Net: A 3D Convolutional Neural Network to Predict Fluid Flow through Porous Media. *Advances in Water Resources* 138: 103539.

Sullivan MD, Foreman JL, Jennetteet DC, et al. (2004) An integrated approach to characterization and modeling of deep-water reservoirs, Diana Field, Western Gulf of Mexico. In: Grammer GM, Harris PM and Eberli GP (eds) *Integration of Outcrop and Modern Analogs in Reservoir Modeling: AAPG Memoir 80*. Tulsa, OK: AAPG, pp.215–234.

Sullivan MD, Jensen GN, Goulding FJ, et al. (2000) Architectural analysis of deep-water outcrops: Implications for exploration and development of the Diana Sub-Basin, Western Gulf of Mexico. In: Weimer P (ed) *Deep-Water Reservoirs of the World: 20th Annual*. Tulsa, OK: SEPM Society for Sedimentary Geology, pp.1010–1031.

Wu J, Zhang C, Xue T, et al. (2016) Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling. arXiv preprint arXiv:1610.07584.

Xie Y, Cullick AS and Deutsch CV (2001) Surface-geometry and trend modeling for integration of stratigraphic data in reservoir models. In: *SPE Western Regional Meeting*, Bakersfield, California, USA, 26–30 March.

Yeh RA, Chen C, Lim TY, et al. (2016) Semantic image inpainting with deep generative models. In: *The 30th IEEE conference on computer vision and pattern recognition*, Honolulu, Hawaii, USA, 21–26 July 2017, pp.6882–6890. Los Alamitos, CA: Institute of Electrical and Electronics Engineers Inc.

Zhang J, Wu S, Fan T, et al. (2016) Research on the architecture of Submarine-Fan Lobes in the Niger Delta Basin. *Journal of Palaeogeography* 5(3): 185–204.

Zhang T, Tilke P, Dupont E, et al. (2019) Generating geologically realistic 3D reservoir facies models using deep learning of sedimentary architecture with generative adversarial networks. *Petroleum Science* 16: 541–549.

Zhang X, Pyrcz MJ and Deutsch CV (2009) Stochastic surface modeling of deep-water depositional systems for improved reservoir models. *Journal of Petroleum Science and Engineering* 68(1–2): 118–134.

Appendix 1. Model description

In this section, we explain the architecture of our GAN. As aforementioned, GAN consists of two models: a generative and discriminative model. The generator creates new realizations, whereas the discriminator calculates the likelihood of the input image to be true. The accuracy of the discriminative model is utilized to train both the generative and discriminative models by optimizing the loss function (equation (1)).

We design our GAN using 3D convolutional layers, as shown in Figure 15, to take advantage of the spatial correlation in all dimensions. Below we will provide a detailed summary of each of the components of these two models.

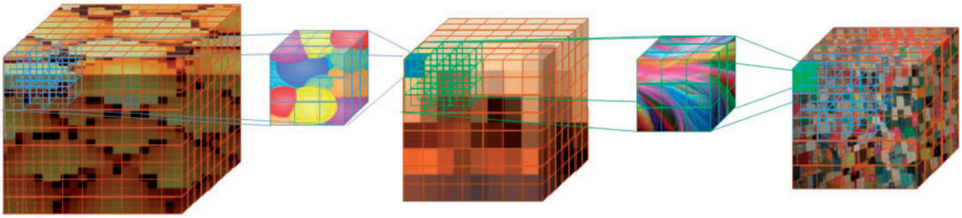

Figure 15. Schematic of three subsequent convolution operations using a 3 × 3 × 3 filter and a stride (kernel distance of where the next convolution operation is performed) of two. The input is a 3D deep-water lobe reservoir section, then the network is trained to create a more compact representation, while retaining relevant features of the original image. Although the image loses the structure to the human eye (as one can see after the third convolution), it retains the most significant information to the network. This operation allows to capture local and global spatial relationships by convolving over the output of the previous convolutional block. It is also cheaper to train because it has a smaller number of parameters (smaller filters) compared to a fully connected network. Source: Modified from Santos et al. (2020).

Table 3. Architecture of the discriminative model.

Layer	*Type*	*Output dimension (channel, width, height, depth)*	*Kernel size*	*Stride*	*Activate function*	*Associated parameters*
1	Convolution	(32, 14, 14, 10)	4 × 4 × 4	2	Leaky ReLU	Alpha = 0.2
2	Dropout	(32, 14, 14, 10)	–	–	–	Rate = 0.25
3	Convolution	(64, 7, 7, 5)	4 × 4 × 4	2	Leaky ReLU	Alpha = 0.2
4	Zero padding	(64, 8, 8, 6)	–	–	–	–
5	Batch Normalization	(64, 8, 8, 6)	–	–	–	Momentum = 0.8
6	Dropout	(64, 8, 8, 6)	–	–	–	Rate = 0.25
7	Convolution	(128, 4, 4, 3)	4 × 4 × 4	2	Leaky ReLU	Alpha = 0.2
8	Batch Normalization	(128, 4, 4, 3)	–	–	–	Momentum = 0.8
9	Dropout	(128, 4, 4, 3)	–	–	–	Rate = 0.25
10	Convolution	(128, 2,2, 2)	4 × 4 × 4	2	Leaky ReLU	Alpha = 0.2
11	Batch Normalization	(128, 2, 2, 2)	–	–	–	Momentum = 0.8
12	Dropout	(128, 2, 2, 2)	–	–	–	Rate = 0.25
13	Flatten	1024	–	–	–	–
14	Dense	1	–	–	Sigmoid	–

ReLU: the Rectified Linear Unit.

Discriminative model

The discriminative model takes the input image of 28 × 28 × 20 dimensions and passes it through a series of layers described in Table 3. In the *convolutional layers*, the input tensor is convoluted with n-filters, where each of these has a side length specified by the user. The result of this operation is a new feature map per filter. The convolution operation is carried out with a shift of several voxels called *stride*. Leaky ReLU was chosen as the activation function to introduce non-linearities. This function has the benefit of propagating gradients better. The *dropout layers*, "disconnect" a percentage of the neurons to prevent over-fitting,

Table 4. Architecture of the generative model.

Layer	*Type*	*Output dimension (channel, width, height, depth)*	*Kernel size*	*Stride*	*Activate function*	*Parameters*
1	Dense	31,360	–	–	ReLU	–
2	Reshape	(128, 7, 7, 5)	–	–	–	–
3	Upsampling	(128, 14, 14, 10)	–	–	–	–
4	Convolution	(64, 14, 14, 10)	3 × 3 × 3	1	ReLU	–
5	Batch Normalization	(64, 14, 14, 10)	–	–	–	Momentum = 0.8
6	Upsampling	(64, 28, 28, 20)	–	–	–	–
7	Convolution	(32, 28, 28, 20)	3 × 3 × 3	1	ReLU	–
8	Batch Normalization	(32, 28, 28, 20)	–	–	–	Momentum = 0.8
9	Convolution	(16, 28, 28, 20)	3 × 3 × 3	1	ReLU	–
9 = >10	Batch Normalization	(16, 28, 28, 20)	–	–	–	Momentum = 0.8
10 = >11	Convolution	(1, 28, 28, 20)	5 × 5 × 5	1	Tanh	–

ReLU: the Rectified Linear Unit.

is applied in the training process. For the last activation layer, we selected a sigmoid, which provides a binary score depending on whether the model interprets that the example was drawn from the training model set or not.

Generative model

The generative model creates a new realization by mapping the latent random vector that represents the range of possible models through a series of layers with trainable parameters (described in Table 4) to the output space (reservoir model sample). Based on the score of the discriminative model, the parameters in the generative model are trained accordingly. The input for the generative model is a standard Gaussian vector (size of 100), the latent random variables, which is then reshaped to have a 3D structure that propagates through the series of layers in Table 4. In the generative model, we use *upsampling* layers, which repeat the input by a factor of two in all the coordinate directions (with no trainable parameters, making it cheaper and avoiding artifacts). The last layer consists of a hyperbolic tangent to add an extra non-linearity and to constrain the output from 0 to 1.

The generative model has 3,892,641 trainable parameters, while the discriminative includes 1,708,001.