An Integrated Framework for Heterogeneous Aquifer Structure Inversion with Concurrent-Single-Image Generative Adversarial and Deep Octave Convolution Dense Residual Networks

Chuanjun Zhan[1], Zhenxue Dai[1,2,*], Javier Samper[3], Liang Chen[4]

[1]*College of Construction Engineering, Jilin University, Changchun, China*
[2]*Engineering Research Center of Geothermal Resources Development Technique and Equipment, Ministry of Education, Jilin University, Changchun, China*
[3]*University of La Coruña, A Coruña, Spain*
[4]*Beijing Research Institute of Uranium Geology, Beijing, China*

Corresponding author's email: dzx@jlu.edu.cn

Abstract:

Accurate identification of heterogeneous aquifer structures is essential for obtaining reliable predictions of groundwater flow and solute transport. Aquifer structures built with only sparse borehole data often maintain large uncertainty. Here we present an integrated inversion framework, which combines a concurrent-single-image generative adversarial network (ConSinGAN), a deep octave convolution dense residual network (DOCRN), and an iterative ensemble smoother for identifying heterogeneous structures.ConSinGAN is able to use a low-dimensional noise vector with a single training sample to represent accurately aquifer heterogeneous structure within a short training time. Simultaneously, the octave convolution layer and the multi-residual connection enable DOCRN to carry out the mapping from the heterogeneous structure to the state field (hydraulic head and concentration distributions), while increasing approximation accuracy, and reducing the GPU memory. ConSinGAN and DOCRN networks are systematically integrated into an iterative ensemble smoother to incorporate available observed data for structure inversion. The performance of the integrated framework is illustrated with two synthetic contaminant transport experiments in 2D and 3D. We show that ConSinGAN can generate a series of

heterogeneous models with geostatistical characteristics similar to those of the training sample. Its training is more than 10 times faster than that of a multi-sample-based generative adversarial network. In addition, DOCRN provided accurate state fields for heterogeneous structures in 2D and 3D. The integrated inversion framework obtained realistic heterogeneous structures. The integration of ConSinGAN and DOCRN reduces significantly reduce the computation time of the inversion process.

1. Introduction

Groundwater flow and solute transport models are widely used in hydrogeological applications, such as contaminant transport, CO_2 geological sequestration, groundwater resource management, and geological radioactive waste disposal (Linde et al., 2015; Paniconi & Putti, 2015). Reliable model prediction depends on a realistic characterization of aquifer heterogeneous structures (Dai et al., 2020). Such structures are commonly identified with geostatistical methods (Dai et al., 2007; Deutsch & Journel, 1992) and from conditional facies data obtained from spare boreholes and outcrops.

Geostatistical methods can be divided into two-point and multi-point methods. The two-point statistical method is typically used to calculate the spatial covariance or the variogram of the stratigraphic facies, which are then used to interpolate the distribution of spatial heterogeneity of the site based on these statistical variables. To overcome the issue of some geological models which are non-Gaussian and highly non-linear, Carle and Fogg (1996) proposed a kriging geological modeling method for indicators based on transition probability. Compared with the covariance approach, this method allows incorporating subjective information (such as the estimation of facies extension lengths), and the sedimentary trend. In recent years, this method has been widely employed to identify heterogeneous facies (Dai et al., 2018; Song et al., 2019). The multi-point statistical method relies on a training image to predict complex heterogeneous aquifer structures (Li et al., 2016; Tahmasebi, 2017). This method is

based on extracting first the spatial distribution pattern of the facies from an image, and then the pattern is matched with conditional data. This method, however, behaves poorly in non-stationary strongly-heterogeneous formations because the image pattern matches incompletely the true facies distribution. In addition, both two-point and multi-point statistics rely primarily on borehole data, and because such data are limited and costly, an inversion method combined with prior knowledge and measured hydraulic head and concentration data are often needed to obtain accurate representations of aquifer structures.

The inversion of aquifer heterogeneous structures has received extensive attention in recent years (Canchumuni et al., 2019; Liu et al., 2019; Soares et al., 2020; Xu et al., 2021). For example, Xu and Gómez-Hernández (2018) used the ensemble Kalman filter to identify the heterogeneous conductivity field by assimilating hydraulic head and concentration data. Laloy et al. (2016) used a sequential geostatistical resampling algorithm to identify the conductivity field based on multi-point statistics. However, in the process of heterogeneous structure inversion using observations, particularly for large-scale aquifers, the inversion parameters are usually high dimensional. Most inversion methods usually require performing multiple forward simulations to obtain convergent solutions (Dai et al., 2014; Dai & Samper, 2004). To alleviate these problems, parameterization methods and surrogate models are often used for the inversion of heterogeneous structures (Mo et al., 2020).

The parameterization method uses low-dimensional vectors to represent high-dimensional aquifer structures. Commonly used parameterization methods include zonation (Harp et al., 2008), level-set function (Song et al., 2019), principal component analysis (PCA) and its variants (Emerick, 2017; Gao et al., 2015; Sarma & Chen, 2009), K-singular value decomposition (Kim et al., 2018), and discrete cosine transformation (Jung et al., 2017). In recent years, the development of deep learning techniques has yielded also encouraging results (Yu et al., 2020), especially in image processing. 2D and 3D aquifer structures can be rendered as images accordingly, and many scholars have resorted to deep learning techniques of image processing for parameterizing

geological structures. Generative adversarial network (GAN) and variational auto-encoder (VAE) are the two most popular methods. Mosser et al. (2017) used GAN to reconstruct 3D porous media structure. Zhang et al. (2019) generated realistic 3D reservoir facies models by using GAN. Chan and Elsheikh (2017) used WGAN (Wasserstein GAN) to parameterize the channelized geological structures together with Bayesian methods to obtain the a posteriori distribution. Laloy et al. (2018) used spatial GAN as the parameterization method to alleviate the computational cost of Markov chain Monte Carlo Bayesian inversion of geological structures. Laloy et al. (2017) used VAE for the inversion of a binary facies structure. Canchumuni et al. (2017) used VAE to parameterize the binary facies model for history matching. Mo et al. (2020) reported an adversarial autoencoder method by combining VAE and GAN to parameterize a non-Gaussian hydraulic conductivity field.

The use of deep learning methods for aquifer model parameterization is limited by the large number of required training samples and the instability of the training process because the quality of the training results may not improve when the training time is increased.

Surrogate models are computationally fast and provide similar results as high-fidelity forward simulations (Keating et al., 2016). In the inversion of the aquifer heterogeneous structure, surrogate models are used to find complex relationships between structures and state variables such as concentrations and hydraulic heads. As the size of both the heterogeneous structure and state field is often large, and the parameters involved are commonly high-dimensional, these relationships tend to be highly nonlinear. It is difficult for some commonly used surrogate methods, such as polynomial chaos expansion (Fajraoui et al., 2011; Jia et al., 2018), kriging (Zhou et al., 2018), and Gaussian processes (Zhang et al., 2016), to obtain optimum results. Owing to their powerful nonlinear mapping capabilities, deep learning techniques have achieved impressive results in building surrogate models with high-dimensional inputs and outputs. Sun (2018) used GAN to map the parameter and state field. A residual recurrent network was used by Kani and Elsheikh (2019) to build a surrogate model of

multiphase flow simulation. Mo et al. (2020) related the hydraulic conductivity field to the concentration and hydraulic head fields with deep residual dense convolution networks (DRDCN). Although these methods have produced inspiring results, there are still some differences between the high-fidelity and the surrogate models. Further, the deep learning-based surrogate model usually needs a large amount of graphics processing unit (GPU) memory during training, the requirement of which may be difficult to meet. It is a limiting factor in both image processing and geoscience; however, with the rapid development of deep learning techniques in recent years, some state-of-the-art methods can alleviate these problems.

Here we present a novel inversion framework which improves GAN based on a single-sample training to parameterize aquifer heterogeneous structures. After training, a series of heterogeneous structures with similar facies distribution patterns as the training sample are generated. This proposed parameterization network is denoted here as concurrent-single-image generative adversarial network (ConSinGAN). It is capable of decreasing significantly the training times through a parallel multi-stage learning strategy with a single training sample. In addition, a novel deep octave convolution residual network (DOCRN) is proposed for the surrogate model. The DOCRN uses octave convolution instead of the traditional convolution layer. It decomposes the training samples into two parts: high- and low-frequency information, which significantly reduces the GPU memory required for surrogate model training. DOCRN uses residual dense blocks (Rakotonirina & Rasoanaivo, 2020) as the basic component of the network. The inclusion of more dense residual structures improves the accuracy of DOCRN results. Simultaneously, Gaussian noise is added to the output of each residual dense block to alleviate model overfitting and instability. Finally, ConSinGAN and DOCRN modules are integrated into an iterative local updating ensemble smoother algorithm (ILUES, Zhang et al., 2018) to form an integrated ConSinGAN-DOCRN-ILUES inversion framework. The proposed method has been used to identify 2D and 3D aquifer heterogeneous properties.

Section 2 presents the ConSinGAN-DOCRN-ILUES inversion framework. The statistical analyses of the unconditional simulation results of ConSinGAN are described in Section 3. Section 4 presents the unconditional simulation results. Section 5 presents the surrogate model performance and the inversion results. Section 6 discusses the advantages and limitations of the proposed inverse framework. Finally, the conclusions are highlighted in Section 7.

2. Problem Formulation

This study deals with contaminant transport in a steady-state flow field. The governing equation is given by (Bear, 1979):

$$\frac{\partial C}{\partial t} = \mathrm{div}(\mathbf{D} \cdot \nabla \mathrm{C}) - \mathrm{div}(\mathbf{u}\mathrm{C}) + S \tag{1}$$

where C ($M \cdot L^{-3}$) is the solute concentration, t (T) is time, **D** (L) is the hydrodynamic dispersion tensor, **u** (L/T) is the pore water velocity and S ($M \cdot L^{-3} \cdot T^{-1}$) is the sink/source. The solute transport equation was solved numerically with TOUGHREACT (Xu et al., 2006).

In this paper, we focus on the inversion identification of the aquifer structure with high heterogeneous by using observation data. The objective function of the inversion problem is the difference between the simulated and measured hydraulic heads and concentrations. Here the aquifer structure is parameterized with a low-dimensional vector. Therefore, the inversion process consists on finding a series of optimal vectors, which are used for aquifer parameterization to obtain realistic aquifer structures.

3. Integrated Inversion Framework

3.1 ConSinGAN Architecture

The proposed GAN architecture is based on ConSinGAN (Hinz et al., 2021). This type of architecture is capable of training the generator to produce realistic images with just an image. A multi-stage parallel training strategy is used to improve the quality of

the generated image in less time; however, in its initial form, the dimensions of the input latent vector were identical to those of the object image. Thus, for the inversion of a large-size aquifer structure, the high dimensionality of the input vector results in a huge computational burden, thus preventing convergence of the inversion process. To overcome these problems, two noise-map blocks were used for mapping the low-dimensional vector to the high-dimensional one with the same shape as the object aquifer. Noise-map 1 block was used in the first stage of training, and noise-map 2 block was used in the follow-up stages (Figure 1). In addition, ConSinGAN was extended to obtain 3D heterogeneous structures.

Figure 2 shows the modified ConSinGAN architecture. ConSinGAN first learns to map a vector to a low-resolution image (Figure 2, training *Stage 0*). Once completed, the size of the generated image is increased via the upsample layer. To improve diversity, an input vector is also added to the upscaled image through the noise-map 2 block, and this mixed image is used as the input for the next training stage. Besides Stage 0, a residual connection is added to connect the input image to the output of each training stage. This process is repeated N times until an image with the desired resolution is generated.

The discriminator architecture is the same as that of SinGAN (Shaham et al., 2019), and the initial version of ConSinGAN. It includes several user-defined convolution blocks, each containing two operations (convolution and activation), and the LeakyRule activation function is used for both generator and discriminator. In this multi-stage training process, the generator and discriminator focuses on global features at the low stage and finer textures at the higher stage.

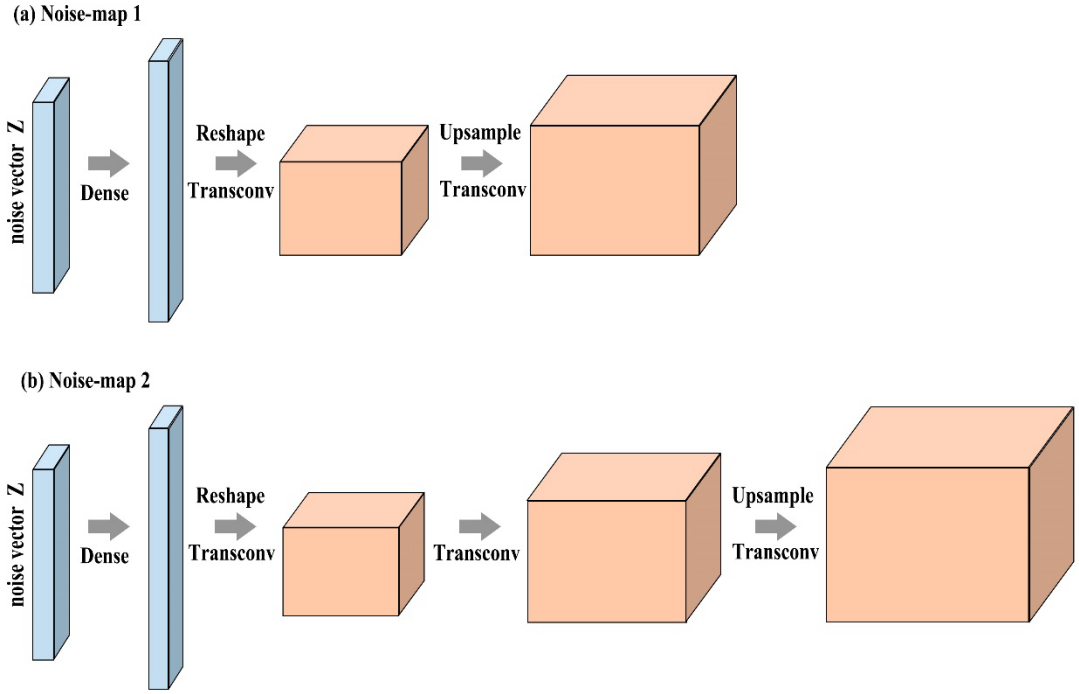

Figure 1. Block structures for the 3D test cases of (a) noise-map 1, and (b) noise-map 2. Dense refers to a fully connected layer; Transconv contains three operations: Transposed convolution (with stride = 2), Batch Normalization, and Activation; The Upsample operation is used to transform the size of the 3D noise vector into the same as the object model at the current stage.

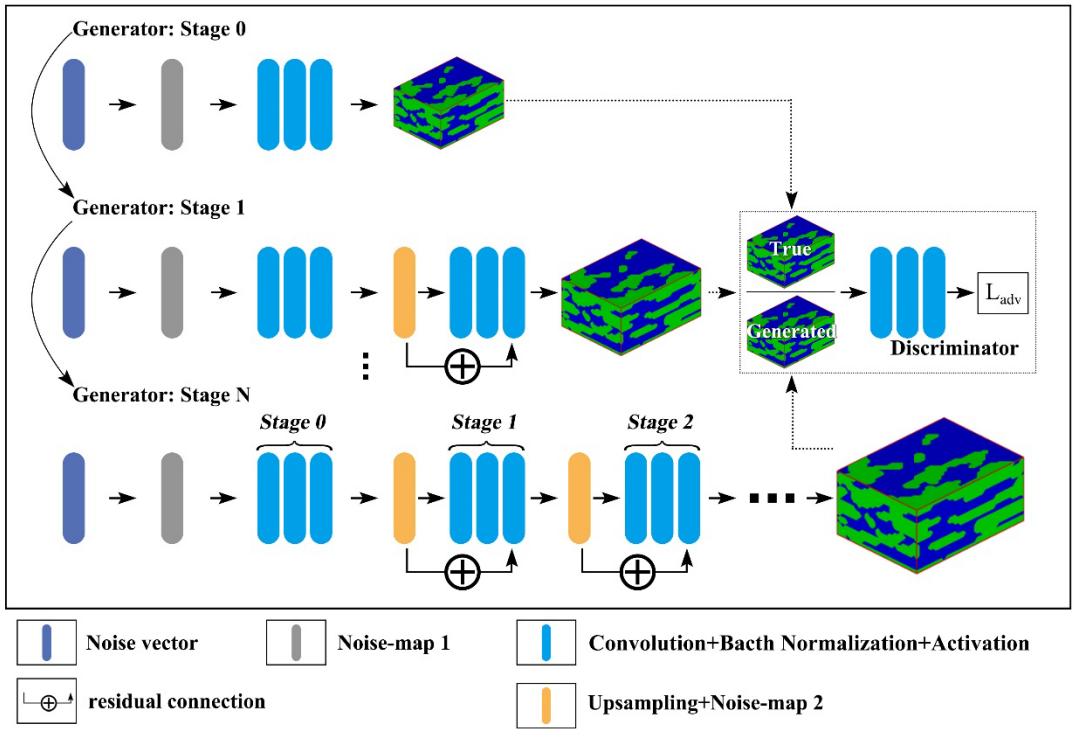

Figure 2. Modified ConSinGAN architecture (modified from Hinz et al. (2021)).

X denotes the true heterogeneous structure (training sample), $\hat{X}$ is the estimated

heterogeneous structure obtained by the generator $\hat{X} = G(Z)$. Z is the noise vector selected from a specific distribution, with a user-defined dimension. Like most other GAN, the generator and discriminator are learned consecutively with opposing targets.

Discriminator $D(.)$ is trained to distinguish $\hat{X}$ from X as much as possible. Simultaneously, the generator $G(.)$ learns the patterns of X, and generates $\hat{X}$ to fool the discriminator into labeling $\hat{X}$ as a true heterogeneous structure. Mathematically, at any given stage N, this training process can be translated into the following minimization-maximization loss function (Eq. 2; Hinz et al. (2021)):

$$\min_{G_N} \max_{D_N} \mathcal{L}_{wgan-gp}(G_N, D_N) + \delta \mathcal{L}_{rec}(G_N) \quad (2)$$

where $\mathcal{L}_{wgan-gp}(G_N, D_N)$ is the WGAN-GP loss function (Gulrajani et al., 2017) shown in Eqs (3 and 4), $\mathcal{L}_{rec}(G_N)$ is the reconstruction loss, and δ is the reconstruction loss weight defined by users to balance the similarity between the true and estimated structures.

The WGAN-GP loss function can significantly improve the stability and speed of GAN training:

$$\mathcal{L}_{wgan-gp}(D_N) = -\mathrm{E}_{x\sim P_{train}} D(x) + E_{x\sim P_{gen}} D(x) + \lambda E_{\tilde{x}\sim P_{\tilde{x}}}[\|\nabla_x D(\tilde{x})\|_2 - 1]^2 \quad (3)$$

$$\mathcal{L}_{wgan-gp}(G_N) = -E_{x\sim P_{gen}} D(x) \quad (4)$$

where P_{train} is the training data distribution, P_{gen} denotes the generated data distribution, and $P_{\tilde{x}}$ is a random sample distribution that uniformly samples along straight lines between P_{train} and P_{gen}, λ is the penalty coefficient, and $\nabla_x D(\tilde{x})$ represents the gradient of discriminator. Both terms of $\mathrm{E}_{x\sim P_{train}} D(x) + E_{x\sim P_{gen}} D(x)$ are the Wasserstein distance used to measure the space between the distributions of X and $\hat{X}$. The introduction of these terms alleviates the problem of gradient vanishing (slowing down the parameter updating speed during GAN training). $\lambda E_{\tilde{x}\sim P_{\tilde{x}}}[\|\nabla_x D(\tilde{x})\|_2 - 1]^2$ is the gradient penalty term used to ensure that the gradient of $D(.)$ is neither too large nor too small, and further prevent gradient explosion and vanishing.

The reconstruction loss $\mathcal{L}_{rec}(G_N)$ was used to measure the similarity between $\hat{X}$

and *X*. The larger the value of δ, the less diverse is $\hat{X}$, and the larger the similarity to *X*. The reconstruction loss function at a given stage *N* is given by:

$$\mathcal{L}_{rec}(G_N) = \|G_N(z_{rec}) - X_N\| \quad (5)$$

where z_{rec} is a reconstruction noise vector randomly selected from a user-defined distribution before training, and remains fixed throughout training process; and X_N is the downsampled version of the original training image at a given stage *N*.

At each stage, adversarial training is completed in two consecutive steps: First, the parameters of *G*(.) is fixed to train *D*(.) three times to minimize Eq. (3), and subsequently *G*(.) is trained three times with fixed *D*(.) parameters to minimize the sum of Eqs (4) and (5). After training, the generator *G*(.) creates a series of realistic heterogeneous structures with different noise vector inputs.

3.2 DOCRN for Surrogate Modeling

A surrogate model for the forward simulation is an image-to-image translation process to establish a complex nonlinear relationship between the heterogeneous aquifer structures and the state field. For the 3D test case, this process is given by:

$$Y^{L*W*H*D_{OUT}} = F(X^{L*W*H*D_{IN}}) \quad (6)$$

where X is the heterogeneous structure and Y represents the state field; *L*, *W*, and *H* are the length, width, and height of the 3D heterogeneous structure, respectively; D_{IN} and D_{OUT} represent the dimensions of the input and output parameters at each grid, respectively; and *F* denotes the complex nonlinear relationship between input and output parameters. he o performance of the surrogate model is improved by integrating a novel convolution structure, “octave convolution,” and a novel residual block, “residual in residual dense residual block (RRDRB)”, into the DOCRN surrogate model.

3.2.1 Octave Convolution

The information of a natural image can be decomposed into low- and high-frequency information. Low-frequency information is usually the global structure,

whereas more detailed information is generally included in the high-frequency. In neural network training processes, some redundant low-frequency information can be compressed without affecting the learning ability (Chen et al., 2019). The traditional convolution layer is replaced by an octave convolution layer to reduce this redundancy and improve the processing of information with different frequencies. The architecture of the octave convolution is shown in Figure 3. The first octave convolution is used to decompose the input feature map into high- and low-frequency features according to:

$$Y^H = Conv(X) \tag{7}$$

$$Y^L = Conv(Avepool(X)) \tag{8}$$

In contrast, the last octave convolution was used to recombine the high- and low-frequency features into the output feature according to:

$$Y = Conv(X^H) + Upsample(Conv(X^L)) \tag{9}$$

The middle octave convolution operations are given by:

$$Y^H = Conv(X^H) + Upsample(Conv(X^L)) \tag{10}$$

$$Y^L = Conv\big(Avepool(X^H)\big) + Conv(X^L) \tag{11}$$

where *Conv* contains the following three operations: convolution (with stride = 1), batch normalization, and activation, used to change the channel dimension of the feature map without altering its spatial dimensions. The spatial dimensions are altered by the *Upsample* and *Avepool* operations, where *Upsample* is used to expand the spatial dimension of the feature map, and *Avepool* is used to reduce it.

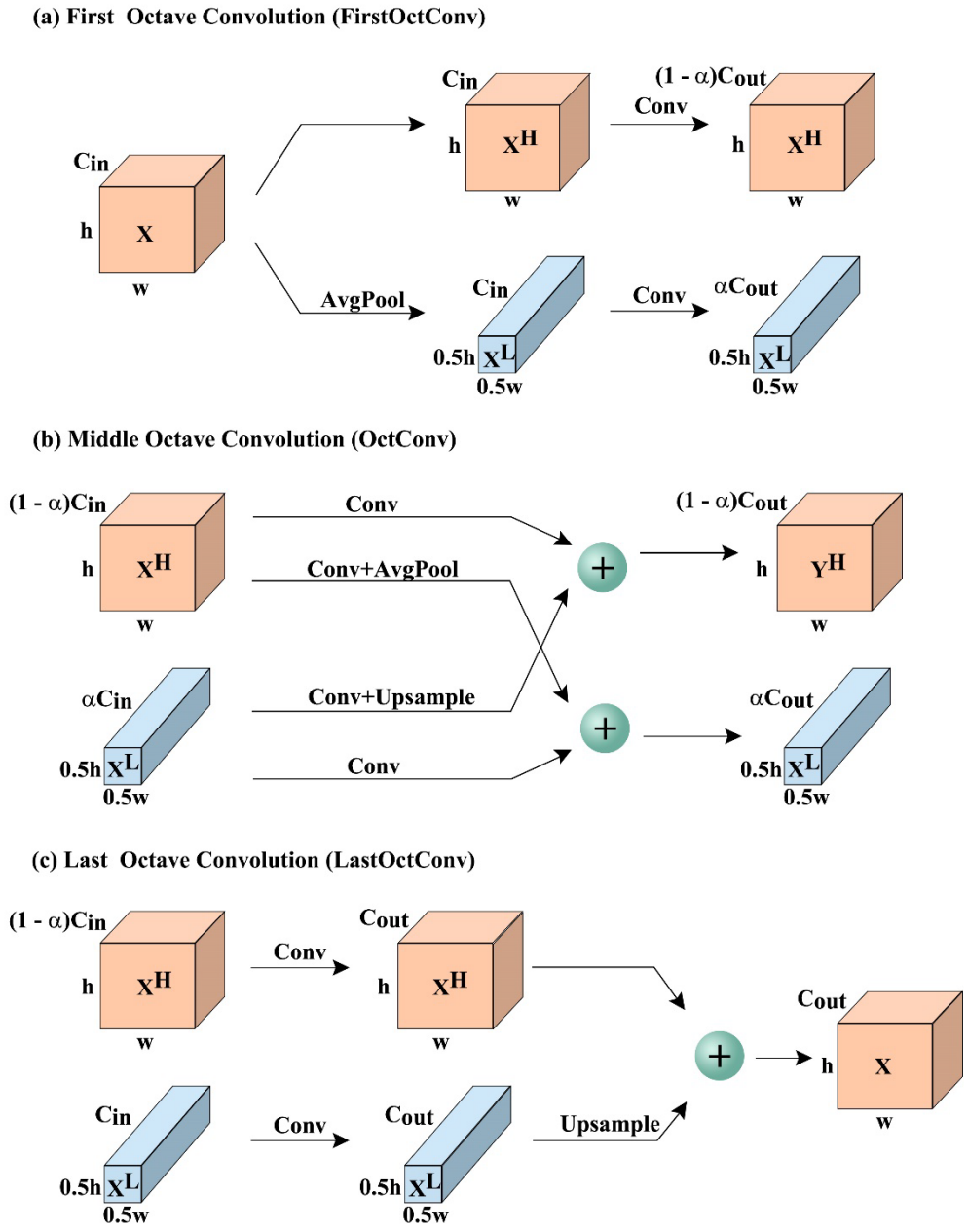

Figure 3. (a) First, (b) middle, and (c) last octave convolution architecture (modified from Chen et al., (2019)). X and Y denote the input and output feature maps, respectively; the superscripts H and L denote the corresponding high- and low-frequency features, respectively; w and h are the spatial dimensions of the images; c_{in} and c_{out} are the channel dimensions; and α is a user-defined parameter representing the ratio of input feature channels used in low-frequency feature maps. Here α is taken equal to 0.25.

3.2.2 Octave Residual in Residual Dense Residual Block

RRDRB was first proposed for super-resolution image generation (Rakotonirina

& Rasoanaivo, 2020). It contains three residual dense blocks (RDBs), which add a residual connection every two layers (blue lines in Figure 4(a)) and enable learning the feature map patterns at multiple levels. This multilevel residual learning increases the network capacity, and ensures that the network can learn a more complex, non-linear relationship between the heterogeneous structure and state field. Concurrently, a crucial role of the residual learning strategy is to alleviate the gradient vanishing problem. To this end, the quality of the translated images using an RDB tends to be better than that of simple dense block, SDB (Rakotonirina and Rasoanaivo, 2020).

To improve training stability, all outputs from the RDBs were multiplied by a residual scale β ($\beta = 0.2$ in the present study; Szegedy et al. (2017)). Similar to the method of adding noise to human face generation tasks, Gaussian noise was multiplied by a learning scale parameter γ ($\gamma = 0.1$ in the present study) and added to the scaled RDB outputs to improve translated image details.

The octave residual dense block (OctRDB) was based on the aforementioned RDB, while replacing the convolution layer with an octave convolution layer. Batch normalization (BN) layers were also used to improve the generalization ability of the surrogate model. For OctRRDRB, besides replacing RDB with OctRDB, the first and last octave convolution layers were added to the head and the tail of RRDRB (Figure 4).

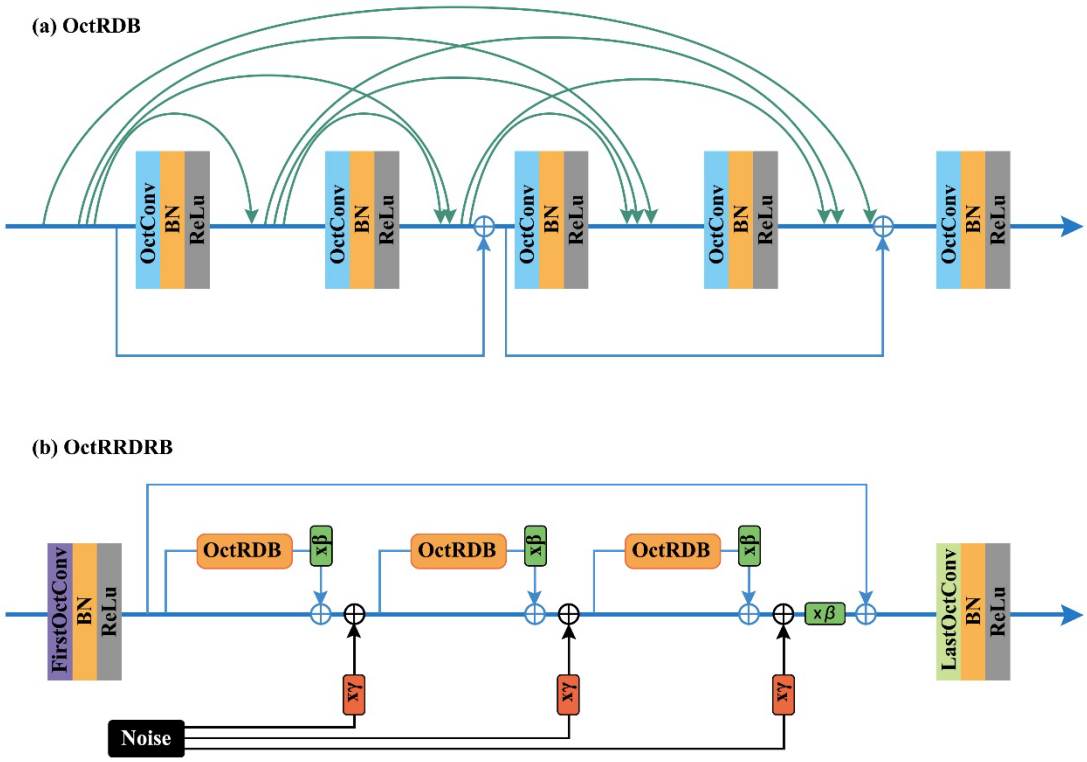

Figure 4. (a) OctRDB contains five layers, a residual connection is added between every two layers, *BN* denotes the Batch Normalization operation and *Relu* is the activation function. (b) OctRRDRB contains three OctRDBs, β is the residual scale, and γ denotes the learning scaling parameter (modified from Rakotonirina & Rasoanaivo, 2020).

3.2.3 DOCRN Networks based on OctRRDRB

OctRRDRB was integrated into the DOCRN surrogate model. DOCRN (Figure 5) adopts a similar network distribution structure to DRDCN developed by Mo et al. (2020). This type of network structure, which resembles a hourglass, enables the network to learn pattern information from coarse to fine scales. DOCRN has four OctRRDRBs, and each OctRRDRB has 17 convolution layers. Contrary to the traditional convolution layer which repeatedly learns some low-frequency information and requires a lot of computational resources, the OctRRDRBs proposed here alleviates this problem. Compared to residual in residual dense block (RRDB), additional residual connection structures and Gaussian noise have contributed to the translation of images with more realistic textures and details (Rakotonirina & Rasoanaivo, 2020). It should be noted that, except for the OctRRDRBs, all other blocks (e.g., Conv1 and Conv2) still

employ traditional convolutional layers.

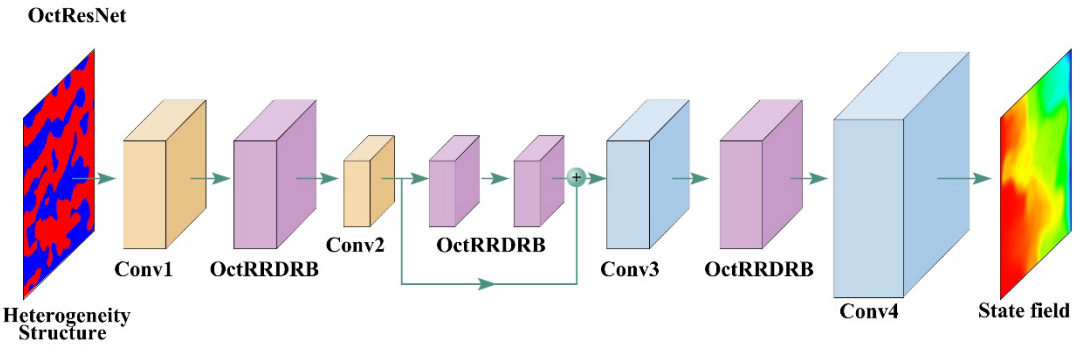

Figure 5. Architecture of DOCRN. The yellow blocks (Conv1, Conv2) have the following three operations: Batch Normalization, Activation and Convolution (with stride = 2). The blue blocks (Conv3, Conv4) involve the following four operations: Batch Normalization, Activation, Upsample and Convolution (with stride = 1).

3.3 ILUES for Inverse Modeling

The purpose of heterogeneous structure inversion (Eq. 12) is to find a series of noise vectors *Z*, which can be fed into ConSinGAN to obtain results ($\hat{X}$) that are similar to the true heterogeneous structure (X):

$$\hat{\mathrm{X}} = ConSinGAN(Z) \tag{12}$$

Similar to Mo et al. (2019), Song et al. (2019) and Zhang et al. (2018) ILUES is used here for inversion.

In practice, one cannot obtain the actual aquifer heterogeneous structure, as only a limited amount of borehole data, hydraulic head, and concentration data are collected. Therefore, the difference between the generated and true structures can be evaluated by using these measurements.

It is assumed that the relationship between the observations and the forward model output for a heterogeneous aquifer can be expressed by:

$$d_{obs} = d^f + e \tag{13}$$

where d_{obs} are the measurements, $d^f = f(G(m))$ is the simulated result of the forward model, $f(.)$, $G(.)$ is the trained generator, m is the noise vector and e is a $N_f * 1$ observational error vector with a mean of zero and a covariance of $C_D = E[ee^T]$.

The ILUES process includes two main steps: First, an ensemble M^0 of the N_{ES}

initial parameter samples is generated from a specific prior distribution (Eq. 14):

$$\mathrm{M}^0 = [m_1^0, m_2^0, \dots, m_{N_{ES}}^0] \tag{14}$$

Second, after the l-th iteration ($l = 1,2, \dots, N_{iter}$), the *nth* realization of the model parameter ensemble m_n^{l+1} is updated via an ensemble smoother scheme by incorporating the measurements (Emerick and Reynolds, 2013):

$$m_n^{l+1} = m_n^l + C_{MF}^l(C_{FF}^l + \widehat{C_D})^{-1}[d_{obs} + \sqrt{N_{iter}} * e_n^l - f(G(m_n^l))] \tag{15}$$

where $\widehat{C_D} = N_{iter} * C_D$ is the inflated covariance matrix, $e_n^l = C_D^{1/2} * r_{N_f}$ ($r_{N_f} \sim N(0, I_{N_f})$) is the *n*-th random realization of the measurement error, C_{MF}^l is the cross-covariance between the parameters after the *l*-th iteration and the simulation results, and C_{FF}^l is the auto-covariance of the forward model predictions. The expressions of C_{MF}^l and C_{FF}^l are given by:

$$C_{MF}^l \approx \frac{1}{N_{ES}-1}\sum_{n=1}^{N_{ES}}[(m_n^l - \overline{m^l})(d_n^{f,l} - \overline{d^{f,l}})^T] \tag{16}$$

$$C_{FF}^l \approx \frac{1}{N_{ES}-1}\sum_{n=1}^{N_{ES}}[(d_n^{f,l} - \overline{d^{f,l}})(d_n^{f,l} - \overline{d^{f,l}})^T] \tag{17}$$

where *T* denotes the transpose and $d_n^{f,l} = f(G(m_n^l))$ is the forward model predictions.

ILUES adopts a local updating scheme in Eq. (15) to avoid problems resulting from the prior or posterior distribution of unknown parameters. The local ensemble of the parameter sample was identified by using the following distance metric:

$$J(\mathrm{m}) = \frac{J_F(m)}{J_F^{MAX}} + \frac{J_M(m)}{J_M^{MAX}} \tag{18}$$

where $J_F(m) = [d^f - d_{obs}]C_D^{-1}[d^f - d_{obs}]$ represents the distance between the simulation results and the measurements, $J_M(m) = [m - m_j^s]C_{MM}^{-1}[m - m_j^s]$ measures the distance between the parameters and the local ensemble sample $m_j^s (j = 1,2, \dots, N_{ES})$, and J_F^{MAX} and J_M^{MAX} are the maximum values of $J_F(m)$ and $J_M(m)$, respectively. Based on the values of *J*, a roulette wheel selection operator (Lipowski & Lipowska, 2012) was used to select the local ensemble of parameters. The selection probability of the *n*-th parameter was determined by $p_n = \frac{\beta_i}{\sum_{j=1}^{N_{ES}} \beta_j}, i = 1,2, \dots, N_{ES}$,

where $\beta_i = \frac{1}{J(m_i)}$ (Mo et al., 2019). More can be found in Zhang et al. (2018) and Mo et al. (2019).

3.4 The Integrated ConSinGAN-DOCRN-ILUES Inversion Framework

The ConSinGAN heterogeneous structure parameterization model and the DOCRN surrogate model are integrated with the ILUES inverse module. The major steps of the integrated ConSinGAN-DOCRN-ILUES inversion framework for heterogeneous structure inversion include:

1. Training the generator ConSinGAN, *G*(.), and the DOCRN surrogate model, *Fs*(.), based on the training samples.
2. Generating the initial ensemble of noise vectors $M^0 = [m_1^0, m_2^0, \dots, m_{N_{ES}}^0]$ from an user-defined distribution.
3. Obtaining the heterogeneous structure ensemble $X^i = [\hat{x}_1^i, \hat{x}_2^i, \dots, \hat{x}_{N_{ES}}^i]$ by $\hat{X}_j^i = G(m_j^i)$.
4. Runing the surrogate model to obtain the simulation result ensemble $d^i = [\hat{d}_1^i, \hat{d}_2^i, , \dots, \hat{d}_{N_{ES}}^i]$ from $\hat{d}_j^i = Fs(\hat{x}_j^i)$.
5. Updating the ensemble of noise vectors by assimilating observation data with ILUES.
6. Repeating steps 3-5 with the updated noise vector ensemble $M^{i+1} = [m_1^{i+1}, m_2^{i+1}, \dots, m_{N_{ES}}^{i+1}]$ until the prescribed maximum number of iterations is reached.
7. Obtaining the posterior heterogeneous structure ensemble $X^{post} = [\hat{x}_1^{post}, \hat{x}_2^{post}, \dots, \hat{x}_{N_{ES}}^{post}]$ from $\hat{X}_i^{post} = G(m_i^{post})$.

3.5 Spatial Statistical Model for Assessing Generation Quality

To evaluate the generator's ability to obtain heterogeneous structures, geostatistical analysis was performed on the ConSinGAN's unconditional generation results. The main focus is the evaluation of the geometry and the spatial correlation structure of the facies. The following three properties are evaluated:

(1) The volume proportion of each facies, p_i, which is calculated from the proportion of the grid number of the *i*-th facies, N_i, and the total grid number:

$$p_i = \frac{N_i}{\sum_j^K N_j} \tag{19}$$

where *K* is the number of different facies.

(2) The mean length of the *i*-th facies in the θ direction, $\bar{L}_i^\theta$, which is obtained by counting each continuous facies block's length in different directions according to:

$$\bar{L}_i^\theta = \frac{\sum_m^{M_i} L_{i,m}^\theta}{M_i} \tag{20}$$

where $L_{i,m}^\theta$ and M_i are the length and the number of the *m*-th continuous block for the *i*-th facies in the θ direction, respectively.

(3) The transition probability, $t_{ij}(\boldsymbol{h}_\varphi)$, which measures the probability that the *j* –th facies is found at a point located a distance $\boldsymbol{h}_\varphi$ from another point with the *i* –th facies (Carle & Fogg, 1997; Ritzi et al., 2004). Its expression is given by:

$$t_{ij}(\boldsymbol{h}_\varphi) = \Pr\{I_i(\boldsymbol{x}+\boldsymbol{h}_\varphi) = 1 \text{ and } I_j(\boldsymbol{x}) = 1\}/\Pr\{I_j(\boldsymbol{x}) = 1\} \tag{21}$$

where $I_i(x)$ and $I_j(x)$ are indicator functions. $I_i(x) = 1$ if facies *i* is at location *x*, and $I_i(x) = 0$ otherwise.

The transition probability rate, r_{ij} is the slope of $t_{ij}(\boldsymbol{h}_\varphi)$ at $\boldsymbol{h}_\varphi = 0$ (Carle and Fogg, 1997):

$$r_{ij} = \frac{\partial t_{ij}(0)}{\partial \boldsymbol{h}_\varphi} \tag{22}$$

It can also be expressed in terms of the facies volume proportion and the mean

length according to (Dai et al., 2019):

$$r_{ij} = \frac{p_i}{\bar{L}_i^{\theta}(1-p_j)} \tag{23}$$

4. Unconditional Simulation Results

4.1 2D Test Case

The 2D test case considered a 100×80 binary training sample generated by cropping randomly the 340×200×80 training image representing the hydrofacies of an alluvial aquifer in the Maules Creek Valley, Australia (Figure 6). The aquifer contains two facies with contrasting permeabilities equal to 10^{-10} m^2 for facies 1 and 10^{-12} m^2 for facies 2. The corresponding ASCII file can be found at http://www.trainingimages.org.

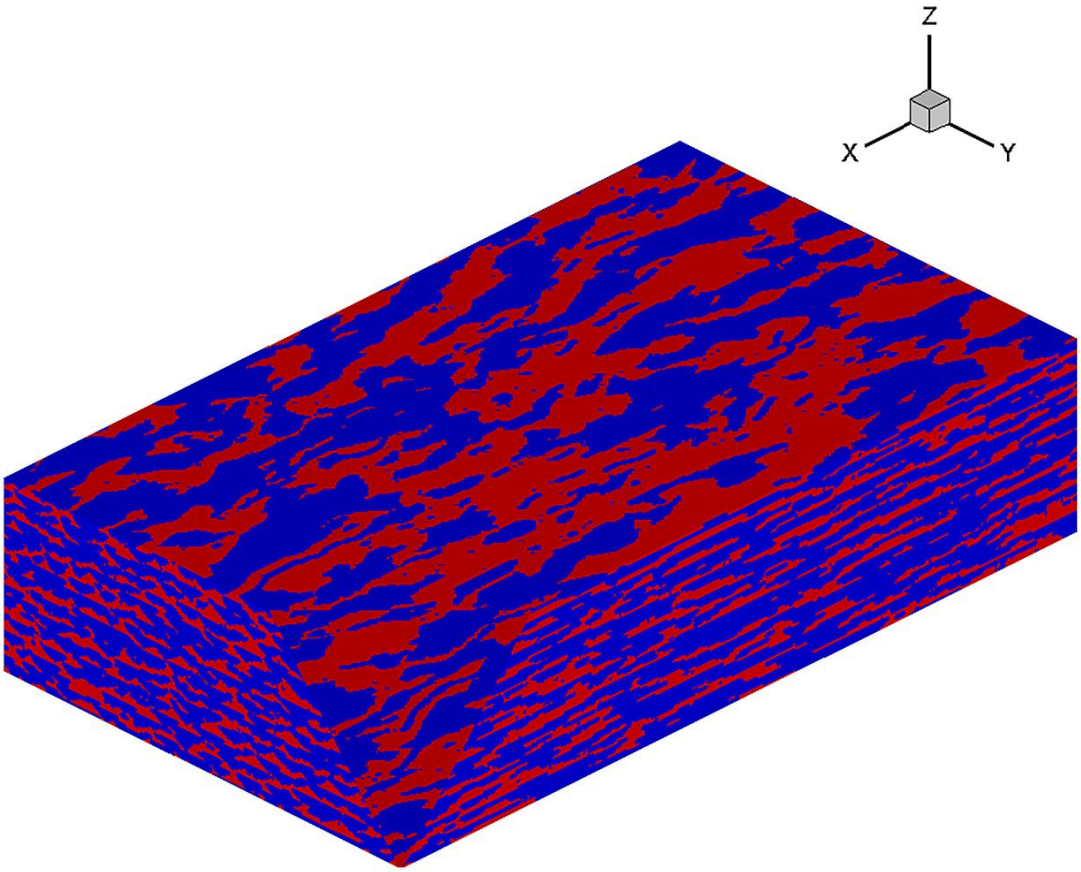

Figure 6. The 340×200×80 training image used for the 2D and 3D test cases.

The random noise vector in the ConSinGAN network was generated from a standard normal distribution, and the dimension of it was set 100. The training stage of the ConSinGAN in the 2D test case was 5, the reconstruction loss weight was 10, and the learning rates of the generator and discriminator were equal to 0.0001 and 0.00005, respectively. With these super-parameters, the ConSinGAN network was trained for 1000 epochs based on a single training sample (Figure 7(a)). The training process took ~280 s on an NVIDIA GeForce GTX 1060 GPU, whereas generating a single 100×80 realization with the trained network took 0.05 s on an Intel i9-10850K CPU.

The reconstructed results and some selected realizations are shown in Figure 7 (b–f). Visually, the reconstruction had nearly the same structure as the training sample, and the trained ConSinGAN was capable of generating realistic heterogeneous structures based on random noise vectors. 10,000 realizations were performed to evaluate quantitatively the trained ConSinGAN network's ability to reproduce the facies volume proportions, the mean lengths in the X and Y directions and the transition probability rates. Figures 8–10 show the frequency distributions of these statistical results. The training sample statistics are marked with arrows in these figures.

The statistical results show that the variation in different realizations compared to the training samples was limited. The statistical results of the volume proportion, mean length and transition probability rate of the heterogeneous structure generated by the trained ConSinGAN network are similar to those of the training model's results and generally present a normal distribution. By comparing the position of the training model's results in the corresponding normal distribution, the volume proportion of facies and mean length in the X-direction are well captured by the ConSinGAN network. However, the mean length in the Y direction, and the transition probability rate distribution are relatively more offset.

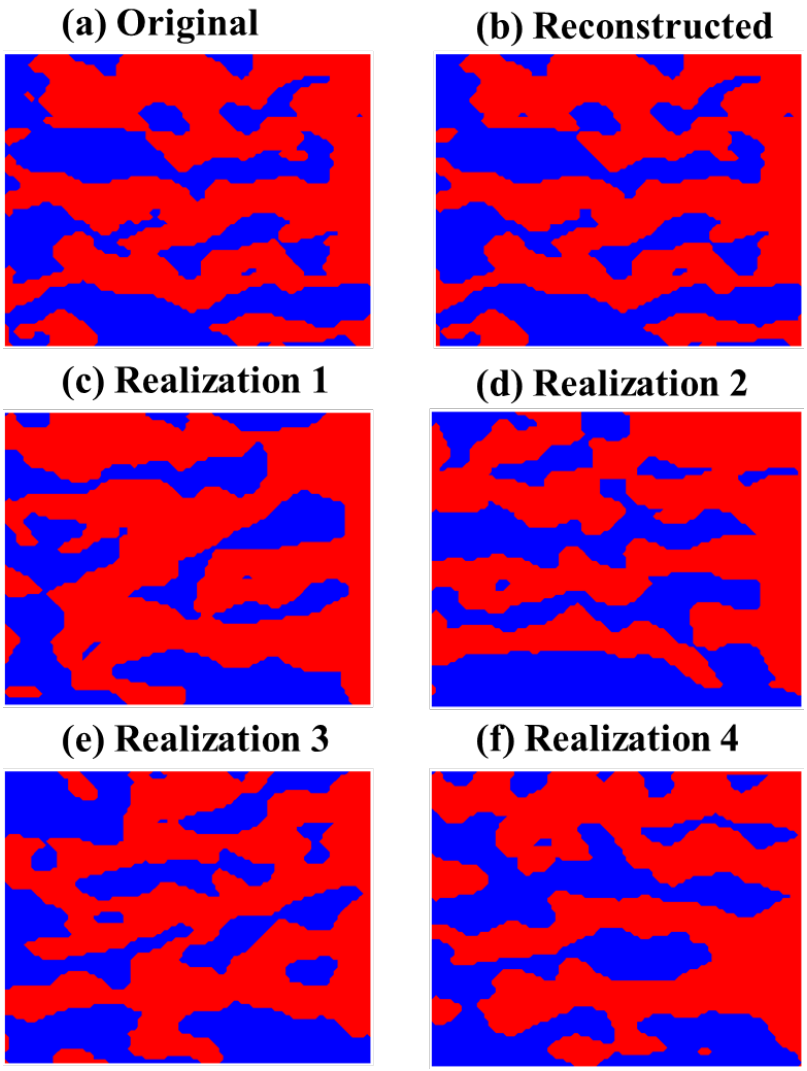

Figure 7. Unconditional simulation results for the (a) original training sample, (b) reconstructed, and (c-f) some selected realizations 1–4 of 2D test case. Blue blocks represent facies 1, and red blocks represent facies 2.

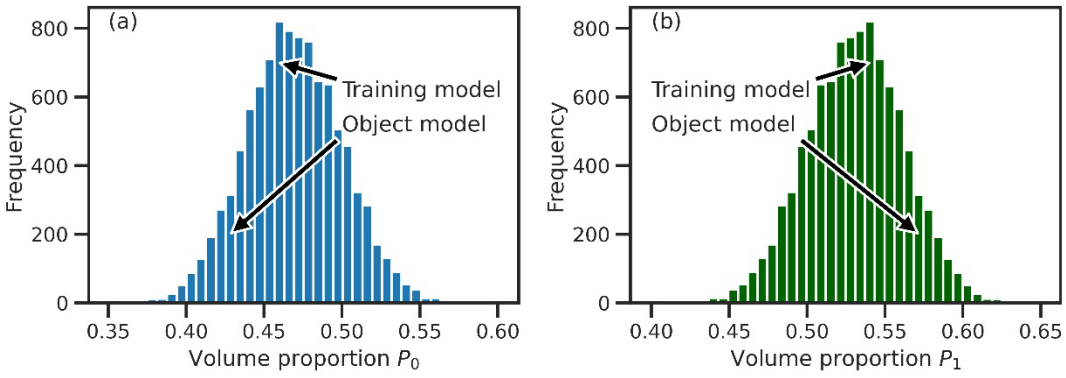

Figure 8. Volume proportion frequency distribution for (a) Facies 1 and (b) Facies 2 of 2D test case.

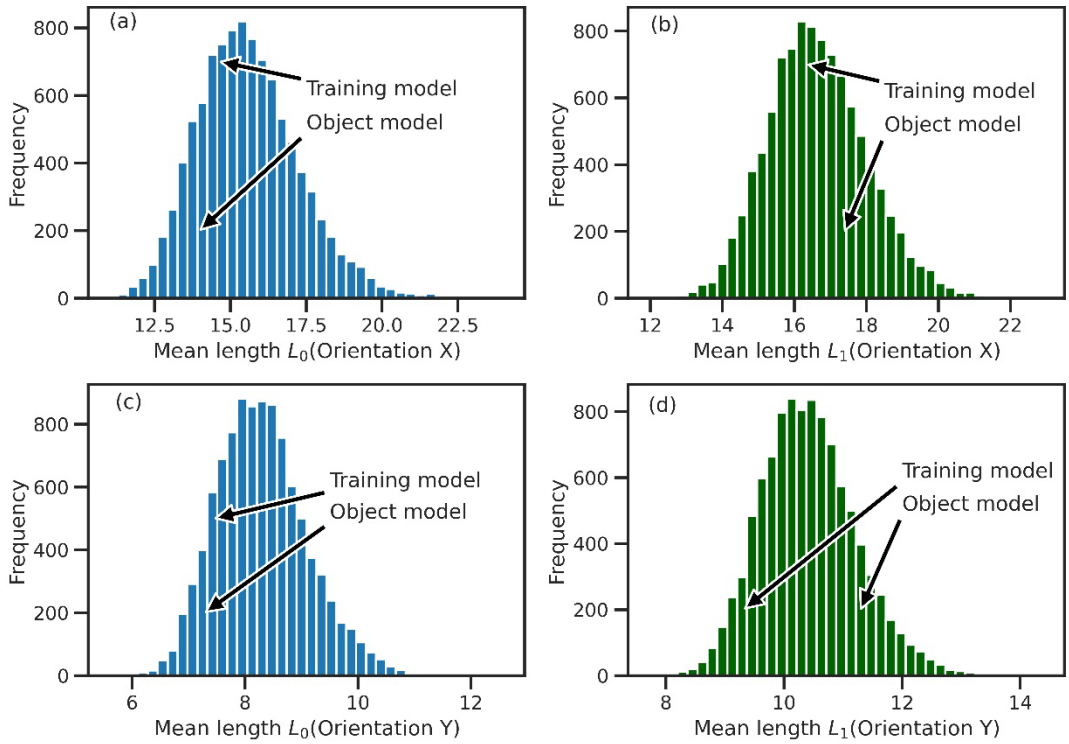

Figure 9. Frequency distributions of mean lengths in the (a, b) X and (c, d) Y directions for Facies 1 and 2 of the 2D test case.

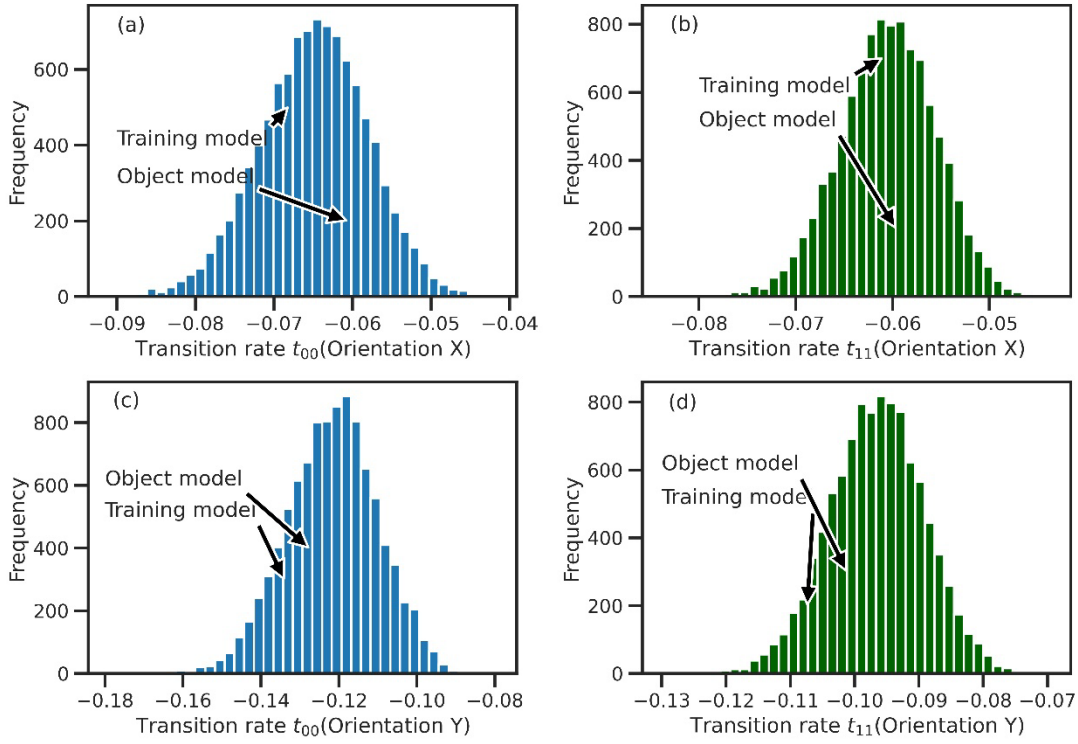

Figure 10. Frequency distribution of auto-transition probability rates in the (a, b) X and (c, d) Y

directions for Facies 1 and 2 of the 2D test case.

4.2 3D Test Case

A 40×30×20 binary training sample (Figure 11(a)) was similarly generated for the 3D test case by cropping randomly the 340×200×80 training image shown in Figure 6. The Gaussian noise vector's dimension is identical to that of the 2D test case. The training stage of ConSinGAN in the 3D test case is 3, the reconstruction loss weight is 10, and the learning rates of the generator and discriminator are equal to 0.0005 and 0.00005, respectively. The ConSinGAN network was trained for 1500 epochs. The training time was ~600 s on the used GPU, and the single image generation time was ~0.2 s using the CPU mentioned above.

Figure 11 (b–f) show the reconstructed results and some selected realizations. The structure of the training sample was reconstructed well, and realistic heterogeneous structures were obtained. The volume proportion of the two facies and the mean lengths and transition probability rates along the X, Y, and Z directions were calculated from 10,000 3D realizations. Figures 12–14 show the frequency distributions of these statistical results.

Similar to the 2D test case, the statistical properties of the generated realizations are similar to those of the training model's results. The ConSinGAN network of the 3D test case captures the volume proportion of the facies better than in the 2D test case. For other variables, such as the mean length and transition probability ratio of most directions, the average value of their distribution was relatively more offset.

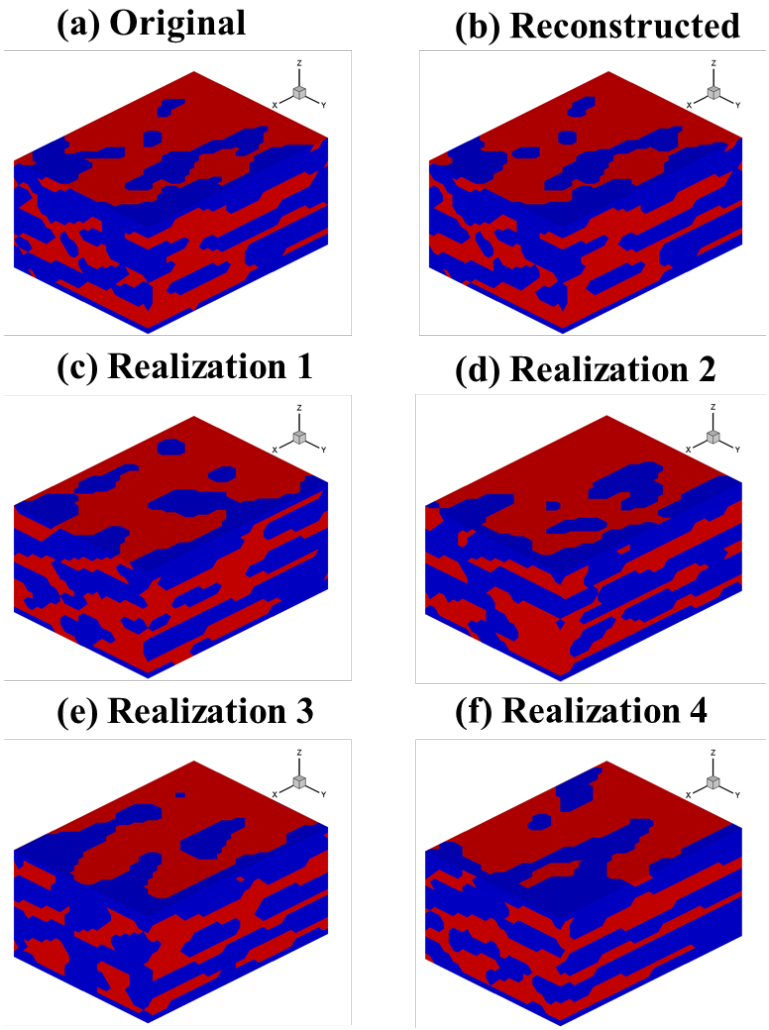

Figure 11. Unconditional simulation results for the (a) original training sample, (b) reconstructed, and (c-f) some selected realizations 1–4 of the 3D test case.

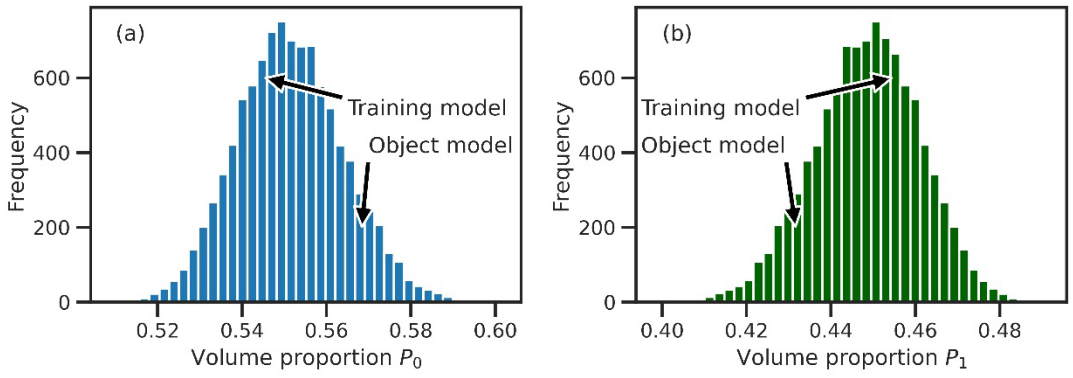

Figure 12. Volumetric proportion frequency distribution for (a) Facies 1 and (b) Facies 2 of the 3D test case.

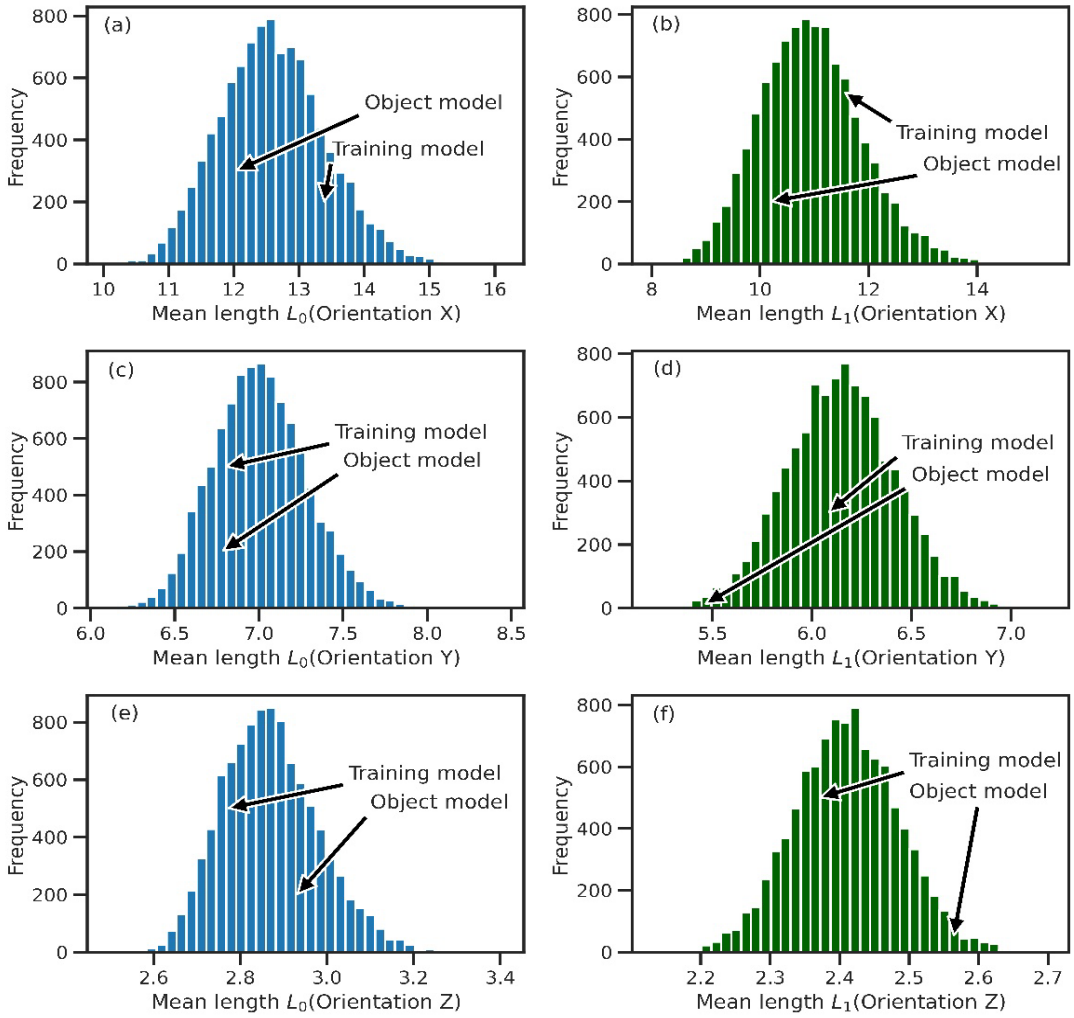

Figure 13. Frequency distributions of mean lengths in the (a, b) X, (c, d) Y, and (e, f) Z directions for Facies 1 and 2 of the 3D test case.

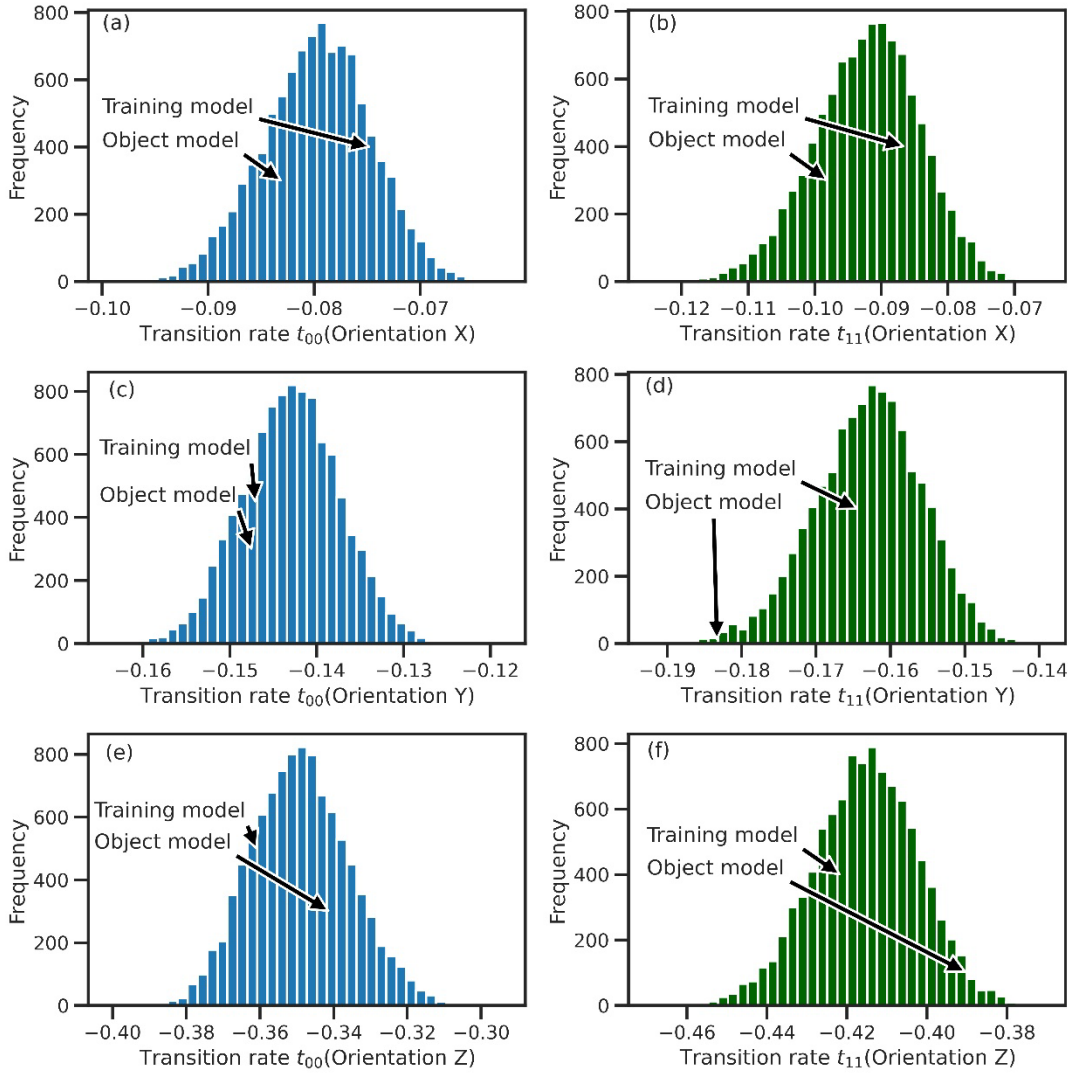

Figure 14. Frequency distribution of auto-transition probability rates in the (a, b) X, (c, d) Y, and (e, f) Z directions for Facies 1 and 2 in the 3D test case.

5. Inversion Framework Evaluation

5.1 Surrogate Model Training Results

5.1.1 2D Surrogate Model

The 2D inversion case considered a hypothetical scenario in which a contaminant source was released in a steady-state flow field. The size of the domain was 100 (L)×80 (L), and was uniformly discretized into 8000 grids of a unit grid size. Dirichlet conditions were adopted for the left and right boundaries with prescribed hydraulic

heads equal to 1 (L) and 0 (L), respectively. The upper and lower boundaries are impervious.

As shown in Figure 15, there was a 3 (L)×7 (L) contaminated area (yellow) near the left boundary. This area continuously released NaCl at a rate of 0.1 $M \cdot L^{-3} \cdot T^{-1}$. A total of 80 monitoring wells (Figure 15, the green blocks) were evenly distributed across the entire area, and seven different time concentrations (0.5, 1, 2, 3, 4, 6, and 12 T) were collected, along with hydraulic head data corresponding to the final time (12 T). The permeability of the two facies within the domain was assigned as 1×10^{-10} m^2 and 1×10^{-12} m^2, respectively. Based on the different heterogeneous structures generated by the trained ConSinGAN, TOUGHREACT (Xu et al., 2006) was used to obtain the training sample set by simulating this hypothetical scenario. In this 2D test case, 10,000 samples were generated for DOCRN training. The training efficiency of network learning was improved by working with dimensionless concentrations, $\hat{c}_k$ which are calculated from:

$$\hat{c}_k = \frac{c_k - c_{min}}{c_{max} - c_{min}} \tag{24}$$

where c_k is the concentrations at grid k, and c_{min} and c_{max} are the minimum and maximum concentrations within the entire training sample set, respectively.

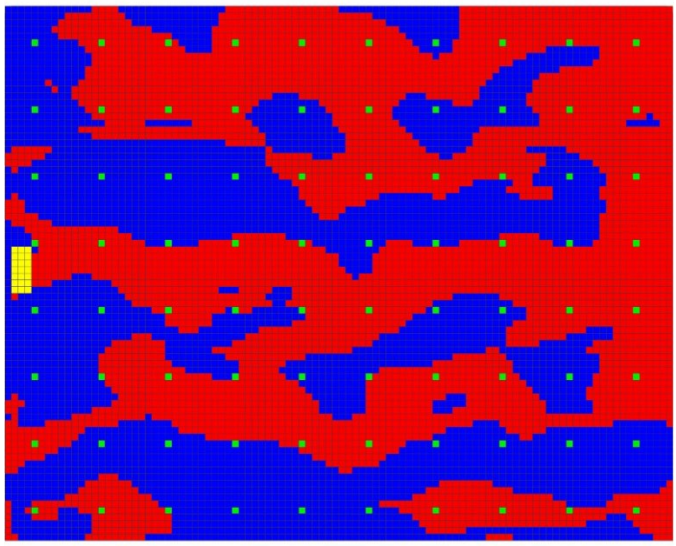

Figure 15. 2D surrogate model's locations of contaminated area (yellow) and observation points (green blocks).

The number of samples was 10,000, 80% were used for training, and the remaining 20% for validation. The batch size of the training process was 128, and an Adam optimizer was used. To prevent overfitting in the later training stage, the following exponential function was adopted to implement a learning rate decay strategy:

$$R_N = R_0 * e^N \tag{25}$$

where $R_0 = 0.0005$ is the initial learning rate, R_N is the updated learning rate in the *N*-th epoch, and *e* is the multiplicative factor of learning rate decay (*e* = 0.99 was used in the present study).

The following regularized L_1 loss function was used in DOCRN training:

$$\mathcal{L}_1 = \frac{1}{K}\sum_{k=1}^{K}\|y_k - \tilde{y}_k\|_1 + \frac{w}{2}\theta^T\theta \tag{26}$$

where y_k is the simulated results by TOUGHREACT at grid *k*, $\tilde{y}_k$ is the predicted value of the DOCRN, *K* is the number of grids, w is the regularization coefficient which is equal to 10^{-5}), and θ denotes the vector of network trainable parameters.

The DOCRN training was conducted on an NVIDIA Tesla V100s GPU for 500 epochs. To evaluate the quality of the DOCRN predictions, the DRDCN developed by Mo et al. (2020) was also trained with the same super-parameters setting on the same GPU. In addition, was also calculated to evaluate The accuracy of the surrogate model was evaluated with the following e_{RMSE} root-mean-square error (RMSE) criterion:

$$e_{RMSE} = \sqrt{\frac{1}{K}\sum_{k=1}^{K}\|y_k - \tilde{y}_k\|_2^2} \tag{27}$$

The L_1 loss, RMSE of the validation set, and GPU memory use were compared between DOCRN and DRDCN (Table 1). All three values were lower for DOCRN; and more specifically, *L1* and RMSE of DOCRN were ~20% less, and the GPU memory cost was decreased by 11%, ultimately favoring DOCRN's performance over DRDCN. The training time of DOCRN was ~5 h, whereas the trained network could generate the concentration and hydraulic head distribution fields within 0.2 s. As shown in figure 16, regardless of the concentration fields or the head field, the results of the DOCRN network and TOUGHREACT were extremely similar, and the differences were mainly distributed along the edges of the contaminant plume. Therefore, it was concluded that

DOCRN could accurately map the heterogeneous structure to the corresponding state field.

Table 1. Comparison of DOCRN and DRDCN results for the 2D test case.

Network	L_1 loss	RMSE	GPU use
DOCRN	0.014553	0.035512	20.48 GB
DRDCN	0.018212	0.042736	24.00 GB

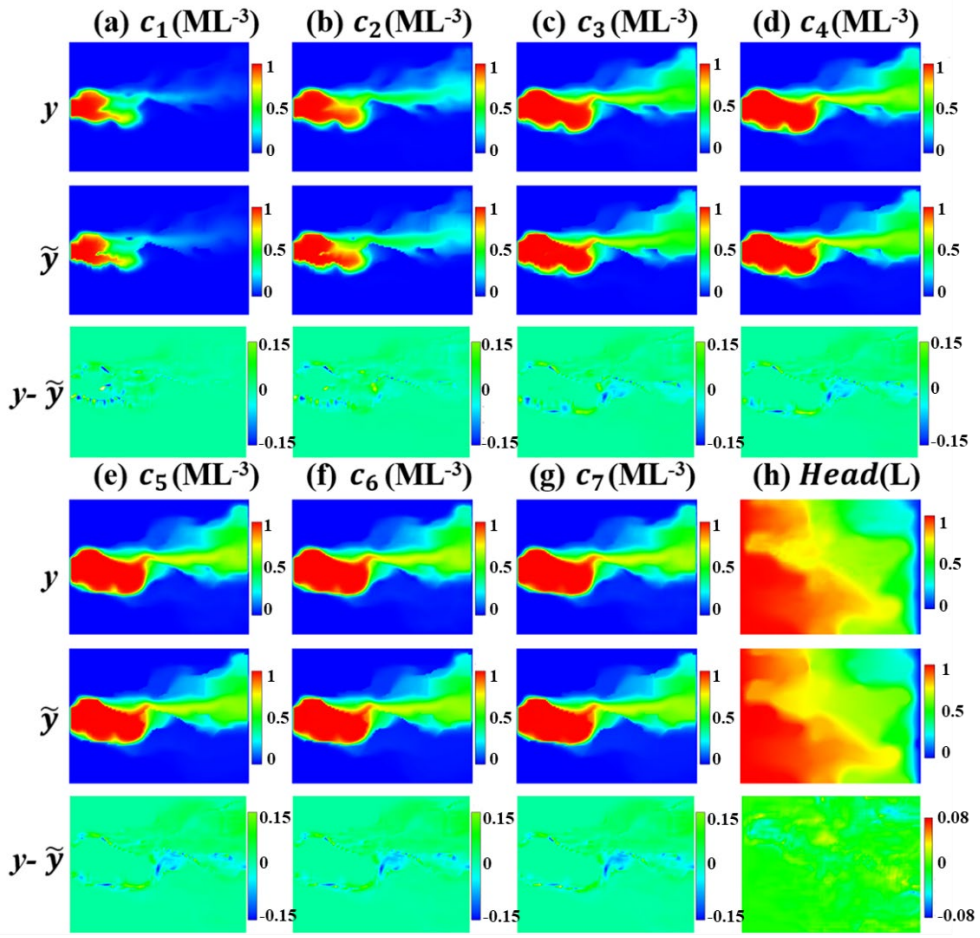

Figure 16. Comparison of the concentration fields of a simulated contaminant plume at (a–g) seven different times, c_1–c_7, and (h) the hydraulic head field for the 2D test case. y is the surrogate model's results, $\tilde{y}$ represents the TOUGHREACT simulation results, and $y - \tilde{y}$ denotes the prediction errors of the surrogate model.

5.1.2 3D Surrogate Model

The model domain of the 3D test case is 40 (L)×30 (L)×20 (L). It was uniformly discretized with unit grid size. Hydraulic heads were prescribed to 1 (L) and 0 (L) at the right and left boundaries, respectively. The remaining boundaries were imprevious. Figure 17 shows the location of the contaminant source (yellow area near right boundary) and the observation points (green blocks). The 3 (L) × 3 (L) × 2 (L) contaminated area had a continuous release rate of NaCl (0.05 $M \cdot L^{-3} \cdot T^{-1}$). There are 48 monitoring wells (light green blocks) to collect observation data at 4 different depths. The collected data type, monitoring frequency and facies permeabilities are similar to those of the 2D test case. 5000 samples were generated for DOCRN training in the 3D test case. Concentration data were normalized according to Eq. (24).

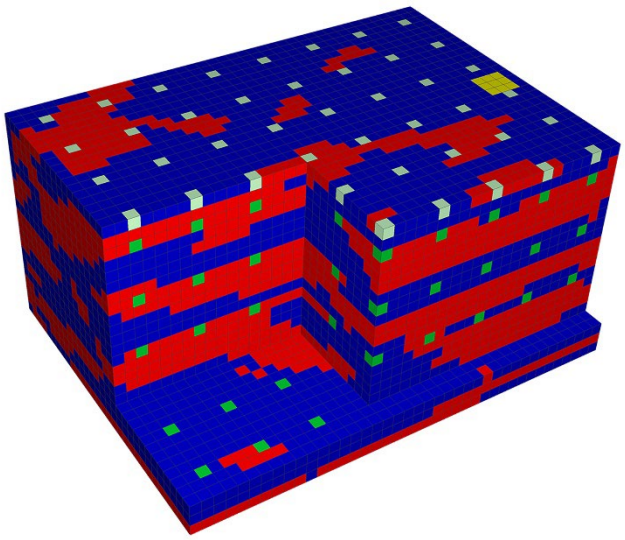

Figure 17. 3D surrogate model's locations of contaminated area (yellow) and observation points (green blocks). Light green blocks represent the locations of monitor wells.

All super-parameters were similar to those of the 2D test case. The save for the batch size was 64. The L_1 loss, RMSE of the validation set, and the GPU memory use of DOCRN and DRDCN are listed in Table 2. Comparatively, the L_1 and RMSE values of DOCRN were ~10% smaller, and the GPU memory cost was 22% smaller than those of the DRDCN. Therefore, it can be concluded that similar to the 2D test case, the

DOCRN outperforms DRCRN in the 3D test case. Because the size of the 3D heterogeneous structure was much larger than that of the 2D test case, the training of DOCRN took nearly 15 h, while the concentration and hydraulic head distribution could still be generated by the trained network within 0.2 s. Some randomly selected results of the validation set are shown in Figure 18. In the 3D test case, the trained DOCRN network could still obtain a state field close to the TOUGHREACT simulation results. Therefore, it can be concluded that DOCRN can handle effectively the heterogeneous structure inversion in 3D test cases.

Table 2. Comparison of the DOCRN and DRDCN results for the 3D test case.

Network	L_1 **loss**	**RMSE**	**GPU Use**
DOCRN	0.011755	0.028675	14.08GB
DRDCN	0.012776	0.030862	18.24GB

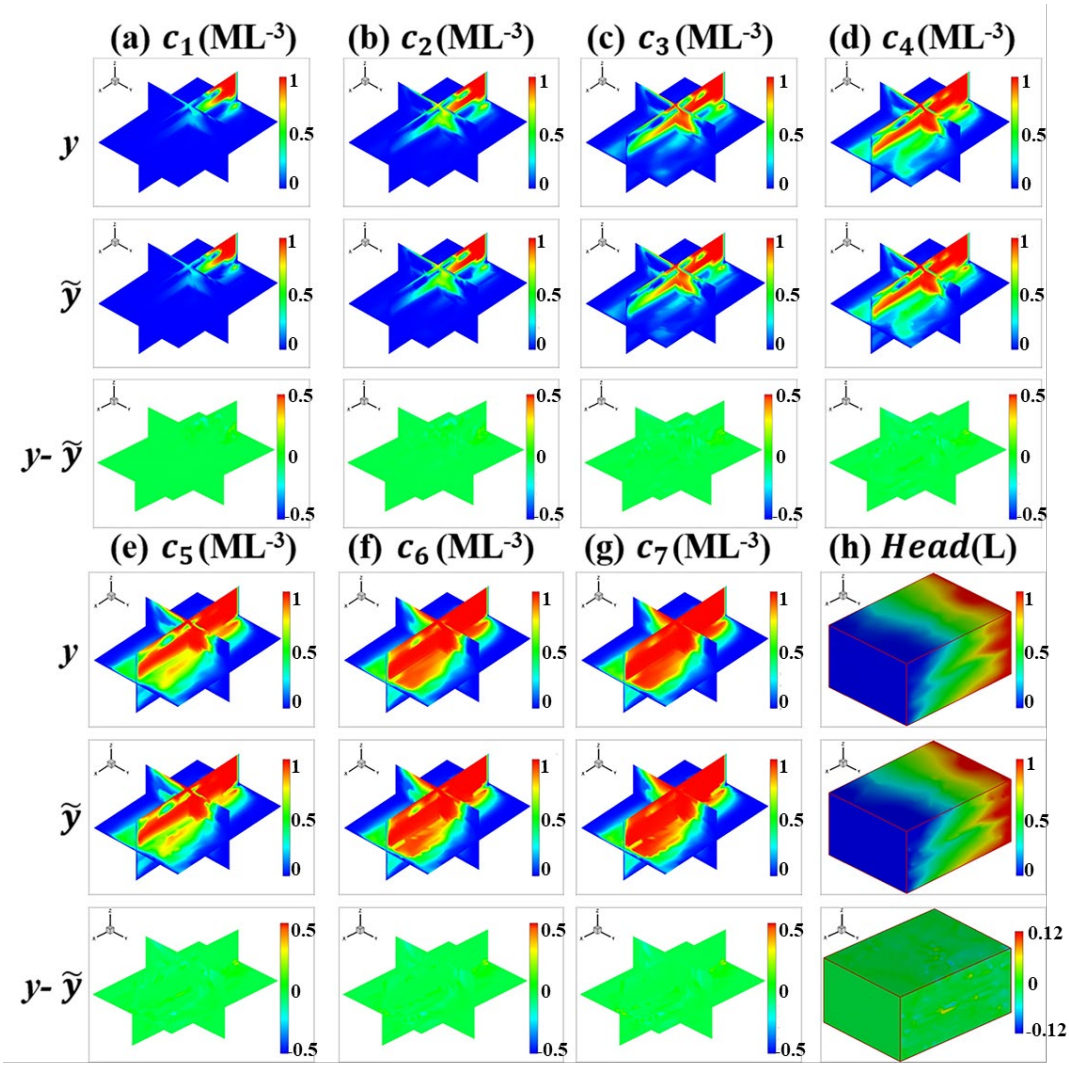

Figure 18. Comparison of the concentration fields of a simulated contaminant plume at (a–g) seven different times, c_1–c_7, and (g) the hydraulic head field based on some randomly selected heterogeneous structure for the 3D test case. y is the surrogate model's results, $\tilde{y}$ is the TOUGHREACT simulation results, and $y - \tilde{y}$ denotes the prediction errors of the surrogate model.

5.2 Inversion results

After training the DOCRN, the ILUES was used to invert the heterogeneous structure. In the 2D and 3D test cases, the concentration data of 80 and 192 observation points at eight different state fields were collected, respectively (Figures 14 and 16). Therefore, 80×8 = 640, and 192×8 = 1536 observation data were used for the inversion of 2D and 3D heterogeneous structures, respectively. To increase the inversion challenge, no conditional data were used (e.g., borehole data), and a 5% observation

error for each observation data was added. The object aquifer (Figure 19) was generated by feeding a randomly selected 100 dimension noise vector **Z** from the standard normal distribution *N* (0,1) into the ConSinGAN network. The statistics of the geometric features and the spatial correlation structure of the object aquifer were calculated. The differences between the object and the training aquifer structure are readily apparent, further increasing the challenge of inversion.

The large ensemble size and iteration number could enable the ILUES to obtain relatively good results (Zhang et al., 2018), but also increase the computational cost. To balance the inversion accuracy and computational cost, the ensemble size $N_e = 3000$, and the iteration number $N_{iter} = 30$ were used in both the 2D and 3D test cases. The local ensemble factor $\alpha = 0.1$ was used here. Therefore, both 2D and 3D test cases inversions required 30 × 3000=90,000 iterations. Although requiring many iterations, the use of the DOCRN surrogate model can significantly reduce inversion time. The inversion of the 2D heterogeneous structure took ~2.4 h, and that of the 3D test case took ~6 h.

After the inversion process was completed, 3000 posterior noise vectors were fed into the trained ConSinGAN network to obtain 3000 corresponding heterogeneous structures. The facies with the largest number of occurrences in each grid were regarded as the grid's posterior facies. The inversion results show that both 2D and 3D posterior models capture the underlying structure of the object model, and only some local features display differences (Figure 19). It is concluded that the ConSinGAN-DOCRN-ILUES framework can obtain accurate and realistic posterior aquifer structure without conditional data when observation errors are not large.

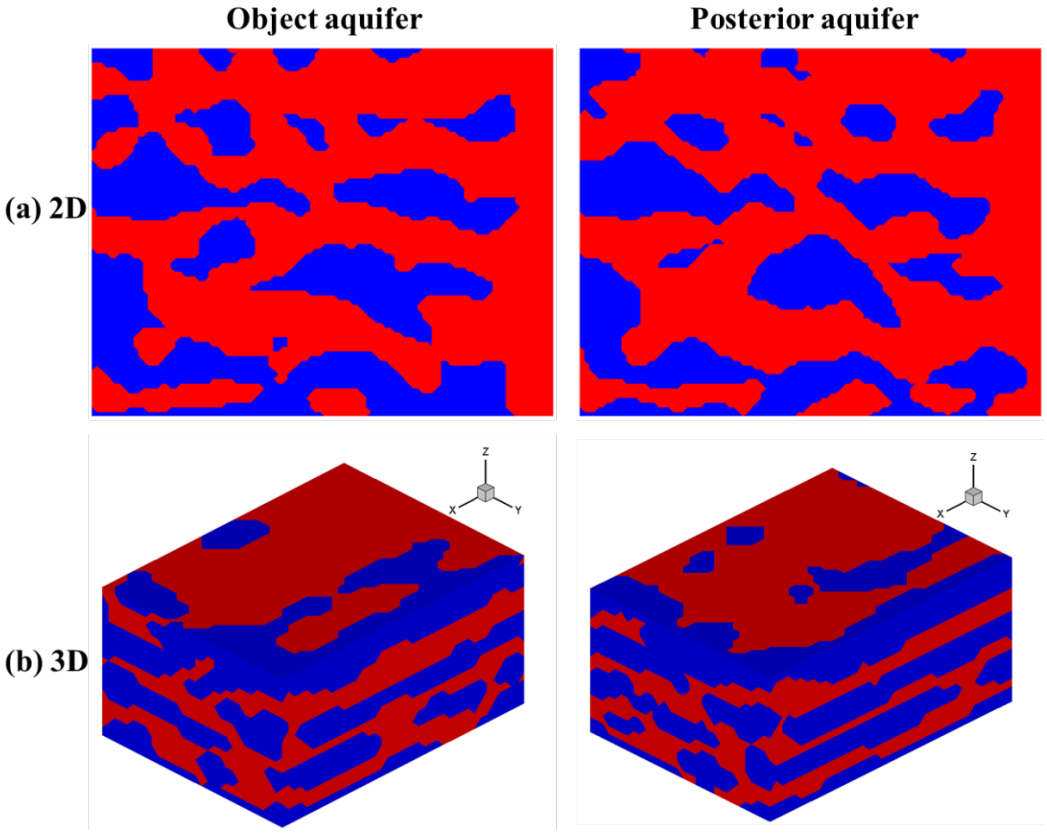

Figure 19. The object and posterior aquifer using the ConSinGAN-DOCRN-ILUES framework in the (a) 2D and (b) 3D test case.

6. Discussion

The results demonstrated that each module of the ConSinGAN-DOCRN-ILUES inversion framework works wells. The posterior aquifer captured the main structure of the true model, with only some observable differences in the local features.

Compared to the multi-sampled-based neural network (Chan & Elsheikh, 2019; Laloy et al., 2018; Liu et al., 2019), the training speed of the proposed ConSinGAN was significantly increased. For example, the training of the ConSinGAN for a 3D heterogeneous structure containing 24,000 grid cells required only 600 s, whereas for a multi-sample-based neural network, generating a heterogeneous structure with a similar

size required several hours (Laloy et al., 2018; Mo et al., 2020). Based on a single training sample, ConSinGAN generated a series of heterogeneous structures with geometric features and spatial correlation structures similar to those of the training sample.

The prediction accuracy of the DOCRN surrogate model was significantly improved by integrating the octave convolution layer and RRDRB, and the octave convolution layer significantly reduced the occupied GPU memory by DOCRN during the training process. This can enable the inversion of heterogeneous structures with larger size when GPU memory is limited. It also permitted for an increase in the batch size used in the training process, which can further improve training the efficiency and the prediction accuracy. Furthermore, the additional residual connection structure and Gaussian noise helped the DOCRN to obtain more accurate prediction results without dramatically increasing the network complexity.

The proposed inversion framework has some limitations. The generated results of ConSinGAN show a diversity poorer than that of the results generated by the multi-sample-based GANs. Therefore, the ConSinGAN is more suitable when the true aquifer structure has statistical characteristics similar to those of the training sample. For cases with large differences between the true model and the training sample, a lower reconstruction loss weight (δ) in the loss function of ConSinGAN can be used. In practical applications, multiple δ values can be tested during the inversion framework. In addition, the local features of the heterogeneous structure are not well identified by the integrated inversion framework employed here, possibly due to: (1) The low dimensionality of the noise vector, making each independent variable of the noise vector influence a large region of the generated heterogeneous structure, and slight changes in each variable could result in significant changes; and (2) There are slight difference between the predicted and simulated DOCRN results of the forward model. The large measurement error and high heterogeneity of the aquifer model may have led to differences between the posterior structure and the true model. Here, the Bayesian inversion method may consider better the observation error and the surrogate model

prediction deviation (Laloy et al., 2018; Zhu & Zabaras, 2018). Further, the uncertainty analysis of the inversion results can also be conducted. However, the Bayesian methods require much longer inversion times.

Besides heterogeneous structure inversion, the inversion framework module proposed here has the potential also for other purposes. For example, the ConSinGAN provides an efficient way to quantify aquifer structure uncertainty owing to its rapid generation speed. The DOCRN and OctRRDRB proposed here can be used also for contaminant source identification, monitoring network optimization, and other applications which require many forward model simulations. The octave convolution layer is suitable for many high-dimensional geoscience applications because the introduction of this layer can effectively reduce the computational burden of network training.

With the continuous development of deep learning techniques, the proposed ConSinGAN-DOCRN-ILUES inversion framework could be further improved, especially for single-sample-based GANs. Such networks were first proposed by Shaham et al. (2019) only recently, and it is expected that more powerful generation capabilities for these methods will be proposed soon, potentially enabling parameterization of heterogeneous structures. In addition, deep learning can also be applied as an inversion method. In the inversion problem discussed here, the neural network can be used to determine directly the relationship between the observations and the input latent vectors. Then, the input noise vector can be directly predicted by the trained neural network based on the observations. This is promising when adequate training data are available, and the neural network capability is sufficiently strong. In this context, it would be unnecessary to use computationally-demanding inversion methods such as the ensemble smoother and the Bayesian inversion.

7. Conclusions

A novel integrated inversion framework for the identification of complex

heterogeneous structures has been presented. The inversion framework combines a concurrent-single-image generative adversarial network (ConSinGAN) for modeling heterogeneous parameterization, a deep octave convolution dense residual network (DOCRN) for surrogate modeling, and an iterative ensemble smoother. The results of the testing of the inversion framework with two synthetic contaminant transport experiments show that

1. The integrated inversion framework was able to identify the heterogeneous structures with a clear reduction of computational time compared to state-of-the-art deep learning methods.
2. The ConSinGAN overcame the dependence of traditional generative adversarial networks on multiple training samples. ConSinGAN was trained with a single aquifer model sample and required much less time. Furthermore, the generated heterogeneous structures showed geometric features and spatial correlation structures similar to those of the training sample. This powerful learning and generating ability greatly enhance the efficiency of the ensemble-based inversion.
3. DODCN outperformed other deep learning networks with similar structures such as DRDCN. DOCRN obtained more accurate predictions and required less GPU memory thanks to the octave convolution layer, the dense residual connections and the additional Gaussian noise. DOCRN enabled the surrogate model-based inversion framework to obtain approximation results of the inversion framework based on the forward model, but significantly improved inversion speed.
4. The integration of the ConSinGAN and DOCRN networks enabled the efficient generation of realistic posterior heterogeneous structures by accounting for indirect observations.

The inversion framework can be further improved by integrating hydraulic head, solute concentration and other types of measured data for aquifer structure inversion such as

core logs, borehole geophysical data and expert knowledge to obtain reliable training samples of ConSinGAN. Although our inversion framework has been tested with binary facies, it can be readily extended to the inversion for multiple facies or continuous parameter structures by replacing the training sample of ConSinGAN.

Acknowledgements

This work was funded by the National Key R&D Program of China (No.2018YFC1800904), the National Natural Science Foundation of China [NSFC: 41772253], Jilin University through an innovation project awarded to the corresponding author [45119031A035], JLU Science and Technology Innovative Research Team [JLUSTIRT 2019TD-35] and partially supported by the Graduate Innovation Fund of Jilin University awarded to the first author (101832020CX233). We thank the ILUES and original ConSinGAN developers for sharing their codes (https://github.com/cics-nd/cnn-inversion;https://github.com/tohinz/ConSinGAN).

References

Bear, J. (1979). Hydraulics of Groundwater. McGraw-Hill, *New York*, p. 569.

Canchumuni, S. W. A., Emerick, A. A., & Pacheco, M. A. (2017). Integration of Ensemble Data Assimilation and Deep Learning for History Matching Facies Models. Paper presented at the OTC Brasil. https://doi.org/10.4043/28015-MS

Canchumuni, S. W. A., Emerick, A. A., & Pacheco, M. A. C. (2019). Towards a robust parameterization for conditioning facies models using deep variational autoencoders and ensemble smoother. *Computers & Geosciences, 128*, 87-102. https://doi.org/10.1016/j.cageo.2019.04.006

Carle, S. F., & Fogg, G. E. (1996). Transition probability-based indicator geostatistics. *Mathematical Geology, 28*(4), 453-476. doi:10.1007/BF02083656

Carle, S. F., & Fogg, G. E. (1997). Modeling Spatial Variability with One and Multidimensional Continuous-Lag Markov Chains. *Mathematical Geology, 29*(7), 891-918. https://doi.org/10.1023/A:1022303706942

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. https://doi.org/10.1007/s10596-019-09850-7

Chen, Y., Fan, H., Xu, B., Yan, Z., Kalantidis, Y., Rohrbach, M., & Feng, J. (2019). Drop an octave: Reducing spatial redundancy in convolutional neural networks with octave convolution. Paper presented at the Proceedings of the IEEE/CVF International Conference on Computer Vision. https://doi.org/10.1109/ICCV.2019.00353

Dai, Z., Middleton, R., Viswanathan, H., Fessenden-Rahn, J., Bauman, J., Pawar, R., & McPherson, B. (2014). An Integrated Framework for Optimizing CO2 Sequestration and Enhanced Oil Recovery. *Environmental Science & Technology Letters, 1*(1), 49-54. https://doi.org/10.1021/ez4001033

Dai, Z., & Samper, J. (2004). Inverse problem of multicomponent reactive chemical transport in porous media: Formulation and applications. *Water Resources Research, 40*(7). https://doi.org/10.1029/2004WR003248

Dai, Z., Wolfsberg, A., Lu, Z., & Ritzi Jr., R. (2007). Representing aquifer architecture in macrodispersivity models with an analytical solution of the transition probability matrix. *Geophysical Research Letters, 34*(20). https://doi.org/10.1029/2007GL031608

Dai, Z., Zhan, C., Dong, S., Yin, S., Zhang, X., & Soltanian, M. R. (2020). How does resolution of sedimentary architecture data affect plume dispersion in multiscale and hierarchical systems? *Journal of Hydrology, 582*, 124516. https://doi.org/10.1016/j.jhydrol.2019.124516

Dai, Z., Zhan, C., Soltanian, M. R., Ritzi, R. W., & Zhang, X. (2019). Identifying spatial correlation structure of multimodal permeability in hierarchical media with Markov chain approach. *Journal of Hydrology, 568*, 703-715. https://doi.org/10.1016/j.jhydrol.2018.11.032

Dai, Z., Zhang, Y., Bielicki, J., Amooie, M. A., Zhang, M., Yang, C., & Stauffer, P. (2018). Heterogeneity-assisted carbon dioxide storage in marine sediments. *Applied Energy, 225*, 876-883. https://doi.org/10.1016/j.apenergy.2018.05.038

Deutsch, C. V., & Journel, A. G. (1992). Geostatistical software library and user's guide. *New York, 119*(147).

Emerick, A. A. (2017). Investigation on principal component analysis parameterizations for history matching channelized facies models with ensemble-based data assimilation. *Mathematical Geosciences, 49*(1), 85-120.

Emerick, A. A., & Reynolds, A. C. (2013). Ensemble smoother with multiple data assimilation. *Computers & Geosciences, 55*, 3-15. https://doi.org/10.1016/j.cageo.2012.03.011

Fajraoui, N., Ramasomanana, F., Younes, A., Mara, T. A., Ackerer, P., & Guadagnini, A. (2011). Use of global sensitivity analysis and polynomial chaos expansion for interpretation of nonreactive transport experiments in laboratory-scale porous media. *Water Resources Research, 47*(2). https://doi.org/10.1029/2010WR009639

Gao, G., Vink, J. C., Chen, C., Alpak, F. O., & Du, K. (2015). Enhanced Reparameterization and Data-Integration Algorithms for Robust and Efficient History Matching of Geologically Complex Reservoirs. Paper presented at the

SPE Annual Technical Conference and Exhibition. https://doi.org/10.2118/175039-MS

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., & Courville, A. (2017). Improved training of wasserstein gans. arXiv preprint arXiv:1704.00028.

Harp, D. R., Dai, Z., Wolfsberg, A. V., Vrugt, J. A., Robinson, B. A., & Vesselinov, V. V. (2008). Aquifer structure identification using stochastic inversion. *Geophysical Research Letters, 35*(8). https://doi.org/10.1029/2008GL033585

Hinz, T., Fisher, M., Wang, O., & Wermter, S. (2021). Improved techniques for training single-image GANs. Paper presented at the Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. arXiv preprint arXiv:2003.11512v2

Jia, W., McPherson, B., Pan, F., Dai, Z., & Xiao, T. (2018). Uncertainty quantification of CO2 storage using Bayesian model averaging and polynomial chaos expansion. *International Journal of Greenhouse Gas Control, 71*, 104-115. https://doi.org/10.1016/j.ijggc.2018.02.015

Jung, H., Jo, H., Kim, S., Lee, K., & Choe, J. (2017). Recursive update of channel information for reliable history matching of channel reservoirs using EnKF with DCT. *Journal of Petroleum Science and Engineering, 154*, 19-37. https://doi.org/10.1016/j.petrol.2017.04.016

Kani, J. N., & Elsheikh, A. H. (2019). Reduced-Order Modeling of Subsurface Multi-phase Flow Models Using Deep Residual Recurrent Neural Networks. *Transport in Porous Media, 126*(3), 713-741. doi:10.1007/s11242-018-1170-7

Keating, E. H., Harp, D. H., Dai, Z., & Pawar, R. J. (2016). Reduced order models for assessing CO2 impacts in shallow unconfined aquifers. *International Journal of Greenhouse Gas Control, 46*, 187-196. https://doi.org/10.1016/j.ijggc.2016.01.008

Kim, S., Min, B., Lee, K., & Jeong, H. (2018). Integration of an Iterative Update of Sparse Geologic Dictionaries with ES-MDA for History Matching of Channelized Reservoirs. *Geofluids, 2018*, 1532868. doi:10.1155/2018/1532868

Laloy, E., Hérault, 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. https://doi.org/10.1002/2017WR022148

Laloy, E., Hérault, R., Lee, J., Jacques, D., & Linde, N. (2017). Inversion using a new low-dimensional representation of complex binary geological media based on a deep neural network. *Advances in Water Resources, 110*, 387-405. https://doi.org/10.1016/j.advwatres.2017.09.029

Laloy, E., Linde, N., Jacques, D., & Mariethoz, G. (2016). Merging parallel tempering with sequential geostatistical resampling for improved posterior exploration of high-dimensional subsurface categorical fields. *Advances in Water Resources, 90*, 57-69. https://doi.org/10.1016/j.advwatres.2016.02.008

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. https://doi.org/10.1002/2015WR018378

Linde, N., Renard, P., Mukerji, T., & Caers, J. (2015). Geological realism in hydrogeological and geophysical inverse modeling: A review. *Advances in Water Resources, 86*, 86-101. https://doi.org/10.1016/j.advwatres.2015.09.019

Lipowski, A., & Lipowska, D. (2012). Roulette-wheel selection via stochastic acceptance. *Physica A: Statistical Mechanics and its Applications, 391*(6), 2193-2196. https://doi.org/10.1016/j.physa.2011.12.004

Liu, Y., Sun, W., & Durlofsky, L. J. (2019). A Deep-Learning-Based Geological Parameterization for History Matching Complex Models. *Mathematical Geosciences, 51*(6), 725-766. https://doi.org/10.1007/s11004-019-09794-9

Mo, S., Zabaras, N., Shi, X., & Wu, J. (2019). Deep Autoregressive Neural Networks for High-Dimensional Inverse Problems in Groundwater Contaminant Source Identification. *Water Resources Research, 55*(5), 3856-3881. https://doi.org/10.1029/2018WR024638

Mo, S., Zabaras, N., Shi, X., & Wu, J. (2020). Integration of Adversarial Autoencoders With Residual Dense Convolutional Networks for Estimation of Non-Gaussian Hydraulic Conductivities. *Water Resources Research, 56*(2), e2019WR026082. https://doi.org/10.1029/2019WR026082

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. https://doi.org/10.1103/PhysRevE.96.043309

Paniconi, C., & Putti, M. (2015). Physically based modeling in catchment hydrology at 50: Survey and outlook. *Water Resources Research, 51*(9), 7090-7129. https://doi.org/10.1002/2015WR017780

Rakotonirina, N. C., & Rasoanaivo, A. (2020). ESRGAN+ : Further Improving Enhanced Super-Resolution Generative Adversarial Network. Paper presented at the ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). https://doi.org/10.1109/ICASSP40776.2020.9054071.

Ritzi, R. W., Dai, Z., Dominic, D. F., & Rubin, Y. N. (2004). Spatial correlation of permeability in cross-stratified sediment with hierarchical architecture. *Water Resources Research, 40*(3). https://doi.org/10.1029/2003WR002420

Sarma, P., & Chen, W. H. (2009). Generalization of the Ensemble Kalman Filter Using Kernels for Nongaussian Random Fields. Paper presented at the SPE Reservoir Simulation Symposium. https://doi.org/10.2118/119177-MS

Shaham, T. R., Dekel, T., & Michaeli, T. (2019). Singan: Learning a generative model from a single natural image. Paper presented at the Proceedings of the IEEE/CVF International Conference on Computer Vision. https://doi.org/10.1109/ICCV.2019.00467

Soares, M. F., Centeno, L. N., Timm, L. C., Mello, C. R., Kaiser, D. R., & Beskow, S. (2020). Identifying Covariates to Assess the Spatial Variability of Saturated Soil Hydraulic Conductivity Using Robust Cokriging at the Watershed Scale. *Journal of Soil Science and Plant Nutrition, 20*(3), 1491-1502. https://doi.org/10.1007/s42729-020-00228-8

Song, X., Chen, X., Ye, M., Dai, Z., Hammond, G., & Zachara, J. M. (2019).

Delineating Facies Spatial Distribution by Integrating Ensemble Data Assimilationand Indicator Geostatistics With Level-Set Transformation. *Water Resources Research, 55*(4), 2652-2671. https://doi.org/10.1029/2018WR023262

Sun, A. Y. (2018). Discovering State-Parameter Mappings in Subsurface Models Using Generative Adversarial Networks. *Geophysical Research Letters, 45*(20), 11,137-111,146. https://doi.org/10.1029/2018GL080404

Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. (2017). Inception-v4, inception-resnet and the impact of residual connections on learning. Paper presented at the Proceedings of the AAAI Conference on Artificial Intelligence.

Tahmasebi, P. (2017). HYPPS: A hybrid geostatistical modeling algorithm for subsurface modeling. Water Resources Research, 53(7), 5980-5997. https://doi.org/10.1002/2017WR021078

Xu, T., & Gómez-Hernández, J. J. (2018). Simultaneous identification of a contaminant source and hydraulic conductivity via the restart normal-score ensemble Kalman filter. Advances in Water Resources, 112, 106-123. https://doi.org/10.1016/j.advwatres.2017.12.011

Xu, T., Gómez-Hernández, J. J., Chen, Z., & Lu, C. (2021). A comparison between ES-MDA and restart EnKF for the purpose of the simultaneous identification of a contaminant source and hydraulic conductivity. Journal of Hydrology, 595, 125681. https://doi.org/10.1016/j.jhydrol.2020.125681

Xu, T., Sonnenthal, E., Spycher, N., & Pruess, K. (2006). TOUGHREACT—A simulation program for non-isothermal multiphase reactive geochemical transport in variably saturated geologic media: Applications to geothermal injectivity and CO2 geological sequestration. *Computers & Geosciences, 32*(2), 145-165. https://doi.org/10.1016/j.cageo.2005.06.014

Yu, Q., Xiong, Z., Du, C., Dai, Z., Soltanian, M. R., Soltanian, M., & Song, Z. (2020). Identification of rock pore structures and permeabilities using electron microscopy experiments and deep learning interpretations. *Fuel, 268*, 117416. https://doi.org/10.1016/j.fuel.2020.117416

Zhang, J., Li, W., Zeng, L., & Wu, L. (2016). An adaptive Gaussian process-based method for efficient Bayesian experimental design in groundwater contaminant source identification problems. *Water Resources Research, 52*(8), 5971-5984. https://doi.org/10.1002/2016WR018598

Zhang, J., Lin, G., Li, W., Wu, L., & Zeng, L. (2018). An Iterative Local Updating Ensemble Smoother for Estimation and Uncertainty Assessment of Hydrologic Model Parameters With Multimodal Distributions. *Water Resources Research, 54*(3), 1716-1733. https://doi.org/10.1002/2017WR020906

Zhang, T.-F., Tilke, P., Dupont, E., Zhu, L.-C., Liang, L., & Bailey, W. (2019). Generating geologically realistic 3D reservoir facies models using deep learning of sedimentary architecture with generative adversarial networks. *Petroleum Science, 16*(3), 541-549. https://doi.org/10.1007/s12182-019-0328-4

Zhou, J., Su, X., & Cui, G. (2018). An adaptive Kriging surrogate method for efficient joint estimation of hydraulic and biochemical parameters in reactive transport

modeling. *Journal of Contaminant Hydrology, 216*, 50-57. https://doi.org/10.1016/j.jconhyd.2018.08.005

Zhu, Y., & Zabaras, N. (2018). Bayesian deep convolutional encoder–decoder networks for surrogate modeling and uncertainty quantification. *Journal of Computational Physics, 366*, 415-447. https://doi.org/10.1016/j.jcp.2018.04.018

Appendix A: List of Acronyms

Acronyms	Description
SinGAN	Single-image generative adversarial network
ConSinGAN	Concurrent-single-image generative adversarial network
DOCRN	Deep octave convolution dense residual network
DRDCN	Deep residual dense convolution network
ILUES	Iterative local updating ensemble smoother
GAN	Generative adversarial network
VAE	Variational auto-encoder
WGAN	Wasserstein generative adversarial network
WGAN-GP	Wasserstein generative adversarial network with gradient-penalty
GPU	Graphics processing unit
CPU	Central processing unit
FirstOctConv	First Octave Convolution
OctConv	Middle Octave Convolution
LastOctConv	Last Octave Convolution
RDB	Residual dense block
SDB	Simple dense block
RRDB	Residual in residual dense block
RRDRB	Residual in residual dense residual block
OctRDB	Octave residual dense block
OctRRDRB	Octave residual in residual dense residual block
BN	Batch normalization