Leveraging Deep Learning with Progressive Growing GAN and Ensemble Smoother with Multiple Data Assimilation for Inverse Modeling

Michael Tetteh[a], **Liangping Li**[a,*], **Arden Davis**[a]

[a]*Department of Geology and Geological Engineering, South Dakota School of Mines and Technology, Rapid City, 57701, USA*

Abstract

The incorporation of hard data in geostatistical modeling is crucial for enhancing the accuracy of interpolating or stochastically estimating subsurface spatial features. The hard data at specified points in the model domain serve as a guide in optimizing the unknown parameters to follow the patterns of the hard data. Recently, a novel approach to solving hydrogeologic/reservoir modeling problems has emerged by using deep generative models, specifically generative adversarial networks (GANs), to generate realistic and diverse images of channelized aquifers. This subsequently can be coupled with inverse models to solve parameter estimation problems. This study focused on using an improved GAN, called a progressive growing generative adversarial network (PGGAN), conditioned with hard data to perform parameter estimation of complex facies models by coupling an ensemble smoother with multiple data assimilation (ES-MDA). First, the PGGAN was trained to an image with 128×128 resolution. The trained PGGAN was used to generate hydraulic conductivity fields when fed an ensemble of latent variables and hard data. The ES-MDA then was used to update the latent variable with the help of hydraulic head data obtained from the groundwater model. The approach was tested on synthetic hydraulic conductivity data. Results show that this approach was able to perform efficient estimation of an unknown facies model domain. Additionally, the proposed method was applied to a different test case of a facies model exhibiting different statistical characteristics. The results were satisfactory, demonstrating that the method is not constrained to the particular hydraulic conductivity fields introduced in the generator's training.

Keywords: Generative adversarial networks, Ensemble smoother, Facies modeling, Deep learning.

---

*Corresponding author

Email address: Liangping.Li@sdsmt.edu (Liangping Li)

Contents

1 Introduction **3**

2 Methodology **7**

2.1 Forward Modeling . . . 7

2.2 Progressive Growing of GANs . . . 8

2.3 Training of PGGAN conditioned to hard data . . . 9

2.4 Ensemble Smoother with Multiple Data Assimilation . . . 10

3 Synthetic Examples **13**

3.1 Cases . . . 14

3.1.1 Case 1: Conditioning PGGAN on hard data only . . . 14

3.1.2 Case 2: Coupling PGGAN and ES-MDA conditioned to only hydraulic head data . . . 14

3.1.3 Case 3: Coupling PGGAN and ES-MDA conditioned to hard data and head data . . . 14

3.1.4 Methods of Evaluation . . . 15

3.2 Results . . . 17

3.2.1 Analysis of PGGAN conditioned to hard data . . . 17

3.2.2 Influence of head conditioning data . . . 18

3.2.3 Random image as reference field . . . 18

4 Reference field outside the training data **19**

5 Conclusion **20**

1. Introduction

The spatial distribution of rock properties within an aquifer/reservoir medium significantly influences the movement and transfer of fluids. Early research relied on a limited amount of data for defining the static properties of subsurface reservoirs or aquifers, like facies types, facies proportions, and facies orientations. Similarly, it also aimed at simulating dynamic data, such as the hydraulic head of an aquifer. Consequently, various geostatistical approaches were developed for creating geological models. Early stages of geomodeling used two-point statistics (i.e., mean and covariance) to capture the statistical properties of the data distribution (Cressie, 1990). Two-point statistics are limited in their ability to depict the variables following a Gaussian distribution, a presumption that does not always align with the complex nature of hydrological systems (Dupont et al., 2018). In addition, two-point statistics do not capture higher-order moments or non-linear relationships between variables, which can be important in hydrologic modeling. Multiple-point statistics (MPS) have been proposed to model the spatial patterns of variables at a larger scale in order to overcome the limitations of two-point statistics (Caers and Zhang, 2004). By conditioning the simulation on a set of training images or patterns, MPS can capture higher-order moments and non-linear relationships between variables (Mahmud and Baker, 2014). While the application of MPS in hydrological modeling has been explored in numerous papers, addressing the computational complexities associated with high-dimensional space remains an ongoing research area (e.g., Tahmasebi, 2018; Zuo et al., 2022).

Deep learning is creating a revolution in science and related industries. Deep learning is a mathematical model that automatically learns a new parametric representation of the data that are fed to them and, in some cases, enables inverse mapping from the learned representation to the original data space (Goodfellow and Courville, 2016). Generative adversarial networks (GANs), developed by Goodfellow et al. (2014), are a type of deep learning model that has gained popularity in recent years because of their ability to generate high-quality images and videos that are often indistinguishable from real-world samples. GANs have numerous applications, including art, fashion, gaming, and virtual reality, and are quickly becoming an indispensable tool in computer vision and machine learning research, where they currently are used widely in hydrogeology and petroleum engineering for aquifer/reservoir modeling for parameterization of geological facies (Laloy et al., 2018; Bao et al., 2022). For example, Dubrule and Blunt (2017) employed a GAN to create 3D images of porous media, while Laloy et al. (2017) utilized a variational autoencoder (VAE) to develop a low-dimensional parameterization of binary facies models with latent variable for inverse modeling with Markov Chain Monte Carlo method. In a subsequent study, Laloy et al. (2018), the same researchers expanded on their initial work by incorporating spatial GANs. Canchumuni et al. (2017) applied an autoencoder to represent binary facies values using continuous variables for history matching with an ensemble smoother. In a later publication, Canchumuni et al. (2018) broadened this parameterization using deep belief networks (DBN). Chan and Elsheikh (2017) employed a Wasserstein GAN to generate binary channelized facies realizations.

The application of GANs in the field of hydrogeology has become a promising approach. GANs are employed to produce hydrogeologic facies models, which are subsequently incorporated into groundwater modeling systems and in previous studies, they are linked with inverse models for performing parameter estimations (Mosser et al., 2018; Laloy et al., 2018). GANs can be used to generate facies models that closely mimic the statistical properties of real-world subsurface data. One of the major advantages of GANs in hydrogeology is that they have been utilized to generate hydrogeologic facies models and have further been incorporated into groundwater models for estimation of hydraulic properties and prediction of groundwater flow. GANs can also be used to generate data for scenarios in which data are limited or unavailable. This can be useful in areas where there are difficulties in accessing monitoring wells or other data sources. Although GANs have shown successes in generating realistic hydrogeologic facies models, recent work by Bao et al. (2020) revealed limitations when reconstructing channel structures of a reference hydraulic conductivity by using a conventional GAN, conditioned to hydraulic head data and coupled with an ensemble smoother with multiple data assimilation. Their results showed that the method was able to reconstruct the channels of the reference hydraulic conductivity that was generated by the GANs but the method was not able to properly reconstruct the hydraulic conductivity field when the reference hydraulic conductivity was randomly obtained from the training image, even as the observed hydraulic head data was increased. In real-world applications, the GAN cannot generate its own reference data points to perform data assimilation. It will be given some limited measured data (e.g., facies type, hydraulic heads, etc.) of a particular area of interest that has not been learned by the GAN, to estimate the parameters of the entire area.

The exploration of utilizing deep learning for conditioning hard data, such as facies, has garnered significant attention in research. In Chan and Elsheikh (2018), they integrated an inference network with a pre-trained GAN to generate facies realizations that are conditioned on facies observations, or hard data. Similarly, Dupont et al. (2018) tackled the challenge of conditioning facies to hard data, employing a semantic inpainting approach with GAN. Liu et al. (2018) took a different route by utilizing the fast neural style transfer algorithm as a generalization of O-PCA for generating conditional facies realizations using randomized maximum likelihood. Ruffino et al. (2020) delved into the effectiveness of conditioning GANs with limited pixel values. Their framework introduced an explicit cost term to the GAN objective function, enforcing pixel-wise conditioning. Zhang et al. (2019) adopted a different strategy, utilizing a pre-trained GAN and adjusting weights, along with a "context loss" defined using distance transformation to measure the mismatch between GAN-generated samples and conditioning data. Canchumuni et al. (2019) explored the use of prior realizations from Convolutional Variational Autoencoder and ES-MDA to condition hard data, testing parameterization in synthetic history-matching problems involving channelized facies. Abdellatif et al. (2022) proposed SPADE-GAN, a conditional Generative Adversarial Network (GAN) model, designed specifically for generating missing facies proportions based on a dataset of geological images. This model was later employed by Fossum et al. (2023) for performing history matching of reservoir simulation using ensemble-based history-matching methods.

The majority of the proposed methods focus on conditioning facies models to hard data by first training GANs or VAEs on facies model realizations. The pre-trained generator is then combined with various approaches like semantic inpainting or geostatistical algorithms. However, Ruffino et al. (2020) took a different approach by directly conditioning hard data to the GAN, allowing the network to learn to generate realizations constrained to the hard data during training through adjustments in the GAN network's weights. Traditional GAN architectures, also known as conventional GANs, generate synthetic images using a single generator and discriminator pair. However, traditional GANs have been criticized for producing low-resolution images that lack detail and are unstable during training (Karras et al., 2018; Song et al., 2021). Progressive Growing GANs (PGGANs) developed by Karras et al. (2018) have recently emerged as a promising alternative to traditional GANs for image synthesis. PGGAN is a technique for training GANs in a more stable and efficient manner. The basic idea is to begin training the GAN on a low-resolution image and gradually increase the image resolution as training progresses. This enables the GAN to learn the image's basic shapes and colors first, and then gradually refine its output to capture more detail and complexity. In a PGGAN, the generator and discriminator initially train on low-resolution images, and layers are progressively added to both networks as the resolution increases. This ensures stability in the training process without disrupting existing layers. One notable advantage of PGGAN is its flexibility to add layers at each resolution, allowing for the incorporation of hard data at different resolution levels. This way, the PGGAN model can learn to generate realizations that honor the specified hard data. Song et al. (2021) effectively incorporated hard data, specifically well facies data, into the Progressive Growing of Generative Adversarial Networks (PGGAN). The generated realizations demonstrated a successful alignment with the provided hard data, indicating a well-conditioned outcome.

The ensemble Smoother (ES) by Evensen (2018) is a data assimilation method that is used in hydrology and other fields to estimate the state of a system based on observations and a numerical model. The basic idea of ensemble smoother is to use a set of model runs, called an ensemble, to generate a probability distribution of possible states of the system, and then update this distribution based on observations in order to obtain a more accurate estimate of the true state of the system. The ES algorithm works by first generating an ensemble of model runs, where each run represents a possible state of the system. The ensemble runs are used to generate a probability distribution of possible states of the system, which represents the uncertainty in the model predictions. Next, the observations are incorporated into the ensemble using a Bayesian update, which adjusts the probability distribution to better match the observed data. The ES algorithm can be used to directly estimate the state of a system at any given point in time. Like other variants of Kalman Filter, the parameters and states have to follow multi-Gaussian distributions for optimal solutions (Evensen, 2018). The accuracy of the estimates depends on the quality of the observations and the model, as well as the size of the ensemble and its degree of representation. In ES, the entire set of model realizations is combined in a single integration to produce a prediction. Subsequently, the ensemble of uncertain parameters is updated using the "Kalman Filter" equations Stordal et al. (2011), assimilating all data simultaneously. ES demonstrates superior performance in the context of linear dynamical models but shows limitations when applied to nonlinear dynamical models, especially in the context of complex aquifers. The intricate nonlinear interactions within these aquifers can contribute to deviations from gaussian behavior in the observed data.. After the introduction of ES for history matching by Skjervheim et al. (2011), two iterative variations of the smoother formulation emerged. The Iterative Ensemble Smoother Algorithm (IES) was proposed by Chen and Oliver (2012), while Emerick and Reynolds (2013) developed the Multiple-Data Assimilation ES (ES–MDA). The iterative nature of these variants addresses some challenges associated with nonlinearity, leading to improved outcomes compared to those achieved with ES alone. Evensen (2018), evaluated two iterative ensemble smoothers for solving inverse problems. The results indicated that ES–MDA performs better as the number of Multiple Data Assimilation (MDA) steps increases. However, achieving convergence with ES–MDA may necessitate a considerable number of MDA steps. On the other hand, Iterative Ensemble Smoother (IES) may require fewer iterations for convergence, but its implementation is separate and may encounter convergence challenges if poorly chosen step length values are used.

In a previous study conducted by Song et al. (2023) which employed four latent vector search approaches including IES, a hybrid model combining a conditioned PGGAN and a Convolutional Neural Network (CNN)-based surrogate was employed for reservoir inverse modeling. The aim was to estimate the properties of reservoir facies models. The CNN-based surrogate served as a forward simulator within the inverse modeling framework. The research approach involved training facies models with channel directions constrained within ±25 degrees of deviation from the north direction. This constraint was implemented to alleviate the training burden on the surrogate. Hydrogeological facies models are often characterized by high heterogeneity, with properties varying across space and in different directions. Additionally, these models frequently feature complex channels with intricate relationships, posing a computational challenge for the surrogate to effectively learn them. Previous studies by Bao et al. (2020) utilized ES-MDA coupled with GAN, resulting in improved minimization of observed and predicted data. However, only hydraulic head data are considered in their studies.

Building on the previous work from Bao et al. (2020) and Song et al. (2023), the current study integrates the conditioned PGGAN with ES-MDA and utilizes MODFLOW as the forward simulator to integrate facies data as well as hydraulic head data into groundwater modeling. To the best of our knowledge, this is the first study to integrate both hard data and soft data (e.g., head data) into hydrogeological modeling by coupling deep learning and data assimilation algorithms. In order to evaluate the influence of conditioned data (both hard and soft data) on hydrogeological inversions, a sensitivity analysis was also conducted. This involved running various scenarios with an incremental increase in conditioned data. Specifically, the study investigates the ability of the coupled conditioned PGGAN and ES-MDA to generate realistic facies data, capturing intricate geological structures, and enhancing the accuracy of hydrogeologic simulations. Furthermore, we assessed the influence of priors sourced from the training image, as opposed to using a generated image from the trained model, aiming for greater realism in the field.

2. Methodology

2.1. Forward Modeling

The single phase groundwater flow equation can be expressed as follows (Fetter, 2018):

$$\frac{\partial}{\partial x}(K_{xx}\frac{\partial h}{\partial x}) + \frac{\partial}{\partial y}(K_{yy}\frac{\partial h}{\partial y}) + \frac{\partial}{\partial z}(K_{zz}\frac{\partial h}{\partial z}) + W = S_s\frac{\partial h}{\partial t} \quad (1)$$

where K_{xx}, K_{yy}, and K_{zz} is the hydraulic conductivity tensors in 3D; h is the hydraulic head; W represents sources and sinks; S_s is the specific storage; x, y, and z represent the coordinates; and t is time. The equation is solved using the finite-difference method in which the aquifer is divided into a number of cells and the head is calculated at each node (Harbaugh et al., 2000).

2.2. Progressive Growing of GANs

The framework of how the generator of the PGGAN was trained is illustrated in Figure 1. The inputs consisted of a 128 $\times$ 1 latent vector, sampled from a uniform distribution ($\mathbf{z} \sim U(-1, 1)$), and hard data (observed data). The hard data were obtained by choosing specific points from the training images. These points represented known well facies at the specified locations. For the purpose of this study, the training images consisted of two facies types. The latent vector first went through a transposed convolution with 4 $\times$ 4 kernel size. The result is a 4 $\times$ 4 latent size. The conditioned hard data then were downsampled from the original 128 $\times$ 128 resolution to various resolutions to match the image sizes of the resolutions at each stage. For resolution 4, a downsampled 4 $\times$ 4 resolution of hard data was concatenated to the output 4 $\times$ 4 latent space produced by the transposed convolution. The result then went through two layers of convolution before it was upsampled to 8 $\times$ 8 resolution. The output went through the same steps to produce 16 $\times$ 16, 32 $\times$ 32, 64 $\times$ 64, and 128 $\times$ 128 resolutions. The final 128 $\times$ 128 resolution went through one more convolution with a 1 $\times$ 1 kernel size to produce a 128 $\times$ 128 image. At each resolution stage, the hard data were introduced in order for the generator to learn to create realistic images with the conditioned data provided.

The discriminator took in both the images generated by the generator as well as a batch of real images to determine, with probability, categories that were real or fake images. Figure 1 shows the architecture of the discriminator and how it worked inversely to how the generator worked. For the 4 $\times$ 4 resolution, the images first went through a convolutional layer with a kernel size of 3 and padding of 1, followed by a second convolution with a kernel size of 1. For this study, this process was termed the final block of convolutions in the discriminator. The discriminator also had a general block that contained two convolutional layers with kernel size of 3 and a downsample layer that downsampled the image by a scale factor of 2. This general block was implemented on the subsequent image resolutions during training before we applied the final block. For example, the 8 $\times$ 8 image resolution passed through the general block, and afterward, the output passed through the final block. The 16 $\times$ 16 image resolution went through the general block two times. The 32 $\times$ 32 image resolution passed through the general block three times. In turn, the 64 $\times$ 64 image resolution passed through the general block four times, and the 128 $\times$ 128 passed through the general block five times. An additional input channel was added to the first convolutional layer of the final block. The purpose of this addition is to calculate the standard deviation of the batch of images that went through the discriminator. This enforced the generator to create output of varied images.

2.3. Training of PGGAN conditioned to hard data

Thirty thousand images with size 128 × 128 were randomly cropped from the original training image (TI) (Zahner et al., 2015) for training (Figure 2). The generator was trained with a batch size of 32 and each resolution was trained for 42 epochs. Only one GPU was used and the total time for completing training was 66 hours. The Adam optimizer was used to train both the generator and discriminator. The Adam optimizer was used with $\beta_1 = 0$ and $\beta_2 = 0.99$. The generator and the discriminator both used a learning rate of 0.03. The hard data conditioned to the generator were formatted as a two-dimensional array containing the locations of the wells and the type of facies known at those locations. For this study, the array elements could have one of three possible values: "1" denoted high conductivity facies, "-1" denoted low conductivity facies, and "0" denoted areas where the facies type was unknown. The generator and discriminator were trained to compete with each other with the aim of improving the performance of the generator to generate images that would convince the discriminator to think they were real images. The discriminator also was been trained to give a low score to the images coming from the generator. This was done by adjusting the parameters of both the generator and discriminator at each iteration. In the case of conditioning hard data, the aim of training the PGGAN conditioned to hard data was to train the generator to create realistic images that honored the conditioned hard data. Therefore, during optimization, an additional loss function was introduced called "context loss", which was added to the generated loss to account for losses caused when the generator created images conditioned to hard data. The context loss used for this work was the L2 norm distance between the real image and the fake image measured at well facies locations similar to the implementation from Song et al. (2021). Figure 3 shows the architecture of how the hard data was conditioned with PGGAN. Below is the loss function used by the generator:

$$L(G) = -\mathbb{E}z \sim p_z\left[D(G(z))\right] + \ln\left(\left|\left(US(G(z)) - x\text{ref}\right) \odot I\right|_2\right). \tag{2}$$

Also, the loss function of the discriminator was similar to the loss function used by (Gulrajani et al., 2017) for WGAN-GP, as given below:

$$L(D) = \mathbb{E}z \sim p_z\left[D(G(z))\right] - \mathbb{E}x \sim p_x\left[D(x)\right] + \lambda\mathbb{E}\hat{x} \sim p\hat{x}\left[\left(\left|\nabla_{\hat{x}} D(\hat{x})\right|_2 - 1\right)^2\right]. \tag{3}$$

The original loss term was:

$$-\mathbb{E}_{z\sim P_z}\left[D(G(z))\right] - \mathbb{E}x \sim p_x\left[D(x)\right] \tag{4}$$

The gradient penalty term was:

$$\lambda \mathbb{E}\hat{x} \sim p\hat{x} \left[\left(|\nabla_{\hat{x}} D(\hat{x})|_2 - 1 \right)^2 \right]. \tag{5}$$

The discriminator's original loss term was in charge of adjusting the discriminator so that it calculated a small score for images generated by the generator and a large score for images taken from the training set. The gradient penalty term of the discriminator was in charge of ensuring that the discriminator did not change too much, causing instability in the PGGAN training process. The term, λ from the gradient penalty expression determined the weight of the penalty. In this work, it was set to $\lambda = 10$ which was proposed by Gulrajani et al. (2017) to work across a variety of architectures used.

Once the training was completed, the generator was able to generate a variety of images within seconds. In the case of conditioning with hard data, the generator was able to create images that conditioned the hard data at their specified locations.

2.4. Ensemble Smoother with Multiple Data Assimilation

The methodology for the ensemble smoother with multiple data assimilation consisted of the following steps:

Step 1: Ensemble Generation

$$\mathbf{K} = \begin{bmatrix} \mathbf{k}_{11} & \mathbf{k}_{12} & \cdots & \mathbf{k}_{1n} \\ \mathbf{k}_{21} & \mathbf{k}_{22} & \cdots & \mathbf{k}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{k}_{m1} & \mathbf{k}_{m2} & \cdots & \mathbf{k}_{mn} \end{bmatrix} \tag{6}$$

The matrix ($\mathbf{K}$) represents the hydraulic conductivity parameters from the ensemble of prior hydraulic conductivity parameters that was integrated into the forward model. The subscript $\mathbf{m}$ is the $\mathbf{m_{th}}$ hydraulic conductivity parameter in the vertical direction and n is the $\mathbf{n_{th}}$ hydraulic conductivity parameter in the horizontal direction. The forward model can be represented as the following:

$$\mathbf{d} = F(\mathbf{K}) \tag{7}$$

where $\mathbf{d}$ is a vector containing the number of simulated data points, such as the hydraulic head in this study; $\mathbf{F}(.)$ is the forward operator, in this case, MODFLOW-2000.

Step 2: Data Assimilation

Multiple data assimilation techniques were applied to the ensemble to incorporate observations into the forecast. We used the ES-MDA method proposed by Emerick and Reynolds (2013) to improve the accuracy of the forecast. The ES-MDA method is a data assimilation technique that updates the ensemble members using a smoother, and which the same measured data can be assimilated multiple times to reach a better result as compared to ES which only assimilates data ones. The following mathematical implementation of ES-MDA is based on (Emerick and Reynolds, 2013).

Define forecast ensemble and observation operator:

$$\mathbf{d} = \begin{bmatrix} \mathbf{h}_{11} & \mathbf{h}_{12} & \cdots & \mathbf{h}_{1N_r} \\ \mathbf{h}_{21} & \mathbf{h}_{22} & \cdots & \mathbf{h}_{2N_r} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{h}_{m1} & \mathbf{h}_{m2} & \cdots & \mathbf{h}_{mN_r} \end{bmatrix} \tag{8}$$

where $\mathbf{H}$ is the ensemble of forecasted hydraulic head data at specified well locations simulated with the forward model, $\mathbf{m}$ denotes the $\mathbf{m_{th}}$ forecasted hydraulic head data for each prior model parameters in the ensemble, and $\mathbf{N_r}$ denotes the number of ensembles.

$$\mathbf{H}_{obs} = \begin{bmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \\ \vdots \\ \mathbf{h}_m \end{bmatrix} \tag{9}$$

where $\mathbf{H}_{obs}$ is the observation or the measured head data.

Compute the forecast ensemble mean and covariance matrix:

$$\bar{\mathbf{d}} = \frac{1}{N_r - 1} \sum_{j=1}^{N_r} \mathbf{d}_j \tag{10}$$

$$\mathbf{C}DD = \frac{1}{N_r - 1} \sum_{j=1}^{N_r} (\mathbf{d}_j - \bar{\mathbf{d}})(\mathbf{d}_j - \bar{\mathbf{d}})^{\mathrm{T}} \tag{11}$$

where $\bar{\mathbf{d}}$ is the forecast ensemble mean, $\mathbf{C}_{DD}$ is the forecast ensemble covariance matrix, N_r is the number of ensemble members, and $\mathbf{d}_j$ is the j_{th} member (vector) of the forecast ensemble. data

Perturb the observation data:

Repeat the observation data, N_r times to form a matrix:

$$\mathbf{H}_{obs} = \begin{bmatrix} \mathbf{h}_{obs11} & \mathbf{h}_{obs21} & \cdots & \mathbf{h}_{obsN_r1} \\ \mathbf{h}_{obs12} & \mathbf{h}_{obs22} & \cdots & \mathbf{h}_{obsN_r2} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{h}_{obs1m} & \mathbf{h}_{obs2m} & \cdots & \mathbf{h}_{obsN_rm} \end{bmatrix} \tag{12}$$

$$\mathbf{d}_{uc} = \mathbf{H}_{obs} + \sqrt{\alpha_i}\mathbf{U}^{1/2} \tag{13}$$

where α_i is the inflation coefficient of the ith iteration ($\sum_{i=1}^{N_a} \frac{1}{\alpha_i} = 1$), $\mathbf{U}$ is a dot product of the observation error covariance $\mathbf{C}_D$ and random noise, $\mathbf{C}_D$ is rescaled with Cholesky decomposition,N_a is the number of iterations, d_{uc} is the perturbed observation data

Compute the forecast anomalies:

$$\mathbf{A} = \mathbf{d}_{uc} - \mathbf{d} \tag{14}$$

where $\mathbf{A}$ is the forecast ensemble anomaly.

Compute the cross-covariance between the forecast ensemble anomalies and the ensemble of latent vector anomalies:

$$\mathbf{C}_{ZD} = \frac{1}{N_r - 1} \sum_{j=1}^{N_r} (\mathbf{z}_j - \bar{\mathbf{z}})(\mathbf{d}_j - \bar{\mathbf{d}})^{\mathrm{T}} \tag{15}$$

where $\mathbf{C}_{ZD}$ is the cross-covariance matrix between the variance of the ensemble of latent vectors and the variance of the ensemble of forecast data, z_j is the j_{th} latent vector in the ensemble, $\bar{z}$ is the mean latent vector.

Update the ensemble members for the i_{th} iteration:

$$\mathbf{z}^{i+1} = \mathbf{z}^i + \mathbf{C}^i_{ZD}(\mathbf{C}^i_{DD} + \alpha_i \mathbf{C}_D)^{-1}(\mathbf{A}^i) \tag{16}$$

In the equation where we need to invert the matrix $C = (\mathbf{C}^i_{DD} + \alpha_i \mathbf{C}_D)$, we face potential issues like singularity and instability due to small singular values. To tackle this, the observation error covariance C_D, was rescaled with the Cholesky decomposition resulting in $C_D = C_D^{1/2}(C_D^{1/2})^T$ as implemented in Emerick (2012), and Truncated Singular Value Decomposition (TSVD) was applied to find the pseudo-inverse, resulting in a matrix $\widehat{C} = U_n \wedge_n V_n^T$. $\wedge_n$ is a diagonal matrix with the N_n largest singular values. Determining N_n involves considering the ratio of the sum of these largest singular values to the total sum (N_T). This ratio must be less than or equal to a given energy threshold E which is 0.99 in this work. In simpler terms, we want to retain enough significant singular values to capture at least E proportion of the total energy. The pseudoinverse C^+ is obtained through $V_n \wedge_n^+ U_n^T$, where $\wedge_n^+$ is the pseudo-inverse of $\wedge_n$ This whole process ensures stability and accuracy in handling the inversion of C.

3. Synthetic Examples

To evaluate our proposed approach, we initially acquired the reference facies model randomly, as depicted in Figures 4 and 5. The hard data were then generated based on this reference model, illustrated in four scenarios in Figure 4. Combining the hard data with an ensemble of 100 latent vectors, each of size (128×1), we fed them into the trained generator, resulting in facies models of size $(128 \times 128 \times 1)$. These generated facies models were further translated into hydrogeological properties, specifically hydraulic conductivity in this study. The hydraulic conductivity fields, derived from the generated facies models, acted as prior model parameters. These parameters were integrated into the forward model to simulate dynamic data, namely hydraulic head. Simultaneously, the reference facies models were also transformed into hydraulic conductivity fields and integrated into the forward simulator to simulate hydraulic heads, serving as our observed hydraulic head data. Hydraulic heads from specified well locations, along with latent vectors and observed hydraulic head data, were assimilated into the ES-MDA algorithm. Through multiple data assimilation steps, the ES-MDA algorithm updated the latent vectors. When these updated latent vectors, along with the hard data, were reintegrated into the trained PGGAN, they produced facies models that closely matched the initial reference facies model.

The groundwater flow model produced from MODFLOW was used as a forward model. An image randomly extracted from the training image as well as an image generated by PGGAN were used as the reference hydraulic conductivity. All areas marked as “1” represent high hydraulic conductivities and areas marked as “-1” are areas representing low hydraulic conductivities. The confined aquifer had a size of 128 m $\times$ 128 m $\times$ 1 m, which was discretized into 128 rows by 128 columns by 1 layer Figure 6. The northern and southern boundaries were taken to be no-flow boundaries. The western side was a constant head boundary

($h = 0m$). The eastern side was given a constant flux of -12.9 m^3/d. The initial hydraulic head was equal to the depth of the aquifer (10 m) over the simulated domain. The specific storage of the aquifer was assumed to be a constant value of 0.003 m^{-1}. The porosity was set to 0.3 everywhere. MODFLOW-2000 (Harbaugh et al., 2000) was used to model groundwater flow for 100 days

3.1. Cases

3.1.1. Case 1: Conditioning PGGAN on hard data only

In this case, we compared the output of the PGGAN to the reference data when conditioned on only hard data. Figure 4 shows the scenarios considered. This comprised four scenarios in which we increased progressively the number of well facies data to examine the incremental impact on the output when compared to the reference image. Scenario 1 included 9 wells of facies data; Scenario 2 had 25 wells of facies data; Scenario 3 had 49 wells of facies data; and Scenario 4 had 81 wells of facies data. It was expected that the output realization after each increase should be closer to the reference image than in the previous scenario.

3.1.2. Case 2: Coupling PGGAN and ES-MDA conditioned to only hydraulic head data

Algorithm 1 described the steps used for this case. The image generated by the PGGAN Figure 5 was used as the reference image in this case for the forward modeling in MODFLOW. The reference image was converted to hydraulic conductivity data, which passed through the forward model to produce hydraulic head data at specified well locations, which then were used as reference head data. The reference head data aided in optimizing the ensemble of latent vectors with the help of the ensemble smoother algorithm, in order to generate images that were close to the reference image. Figure 5 shows well locations, where we want to generate hydraulic heads. Two scenarios were considered, 9 well locations (Scenario 6) and 25 well locations (Scenario 7) equally distributed in the domain. We first generated the reference hydraulic head at these well locations. We generated our random ensemble of the initial latent vector from a uniform distribution. The latent vector, with the help of PGGAN was used to generate the ensemble of hydraulic conductivity fields. These hydraulic conductivity fields were used to simulate the hydraulic heads with the help of the forward model generated by MODFLOW. The ES-MDA was used to optimize the latent vector with the help of the simulated hydraulic head and the reference hydraulic head. The process went through six iterations until the latent vector has been optimized and is able to generate images close to the reference image.

3.1.3. Case 3: Coupling PGGAN and ES-MDA conditioned to hard data and head data

Algorithm 1 describes the steps used in this case. We obtained our hard data from the reference image shown in Figure 7. In case 3, we examine how the number of hard data and head data points affect the

Table 1: Scenarios studies.

Case	Scenario	Ensemble size	Obs Wells	Well facies
	1	100	0	9
1	2	100	0	25
	3	100	0	49
	4	100	0	81
	5	100	0	0
2	6	100	9	0
	7	100	25	0
	8	100	9	9
3	9	100	9	25
	10	100	25	25
4	11	100	25	25
	12	100	49	49

parameter estimation. This was characterized by different scenarios. Scenario 8 looked at the case of 9 observation wells and 9 well facies locations, Scenario 9 included 9 observation wells and 25 well facies locations. Scenario 10 looked at 25 observation wells and 25 well facies locations. We first generated the reference hydraulic head at these well locations. The random ensemble of the initial latent vector was generated from a uniform distribution. In this case, the generator of the PGGAN will take both the latent vector and the hard data as inputs to generate the ensemble of hydraulic conductivity fields that were conditioned on hard data from the reference image. These hydraulic conductivity fields were used to simulate the hydraulic heads with help of the forward model generated by MODFLOW. During optimization, the hard data remained fixed and only the latent vector was optimized with the help of the simulated hydraulic head and the reference hydraulic head. The aim was to show that in real-world applications when provided with some sort of data of a particular hydrogeologic or reservoir domain (for example, hydraulic head and facies at well locations), this method will be able to estimate with a high degree of certainty the hydraulic head of the entire geological area under study.

3.1.4. Methods of Evaluation

Two metrics were used to evaluate the results. The first evaluation was by visually analyzing the individually generated images from cases 1 to 3 to see how closely they match their various reference images. The mean and variance of each scenario were computed to inspect the closeness of the reference images and the images generated after parameter estimation and to determine how the ensemble of hydraulic conductivity parameters varied from each other. The second evaluation was by plotting the root mean square error (RMSE) and spread of the generated hydraulic conductivities of the scenarios. These metrics have been used in other studies (Chen and Zhang, 2006; Bao et al., 2020). RMSE measured the bias between the reference hydraulic conductivity and the estimated hydraulic conductivity of each scenario which helped determine the accuracy of the estimation. The spread measured the uncertainty of ensemble realizations. Below are the equations for the RMSE and the spread:

$$\text{RMSE} = \sqrt{\frac{1}{N_k}\sum_{j=1}^{N_k}(K_j - K_{\text{ref}})^2} \tag{17}$$

$$\text{Spread} = \sqrt{\frac{1}{N_k}\sum_{j=1}^{N_k}\text{Var}(K_j)} \tag{18}$$

where N_k is the number of parameters, K_j is the mean of the estimated parameters, K_{ref} is the reference parameters, and $Var(K_j)$ is the variance at each point.

Algorithm 1: Coupling PGGAN with ES-MDA

Set: N_w = Number of iterations
Set: $\mathbf{d}_{obs}$ = Observation data (hydraulic head)
Set: N_r = The number of ensemble
begin
  Generate the conditioning hard data from the reference hydraulic conductivity, **cond_array**
  Sample initial $\mathbf{z}$ from the uniform distribution $\mathbf{z} \sim U(-1, 1)$
  for $i = 1, 2, \cdots, N_{iter}$ **do**
    $\alpha_i = 2^{N_{iter}-i}$
    for $j = 1, 2, \cdots, N_r$ **do**
      Generate ensemble of hydraulic conductivity field with the trained generator:
      if ***cond_array*** **then**
        $\mathbf{K}_j^i = G(\mathbf{z}_j^i, \mathbf{cond\_array})$
      else
        $\mathbf{K}_j^i = G(\mathbf{z}_j^i)$
      Run forward modeling using the generated $\mathbf{K}_j^i$ to obtain the hydraulic head $\mathbf{d}_j^i$
    Perturb the observation data: $\mathbf{d}_{uc,j}^i = \mathbf{d}_{obs} + \sqrt{\alpha_i}\mathbf{U}^{1/2}$
    Compute the mean of the ensembled hydraulic conductivity $\mathbf{z}^i$
    Compute the mean of the ensembled hydraulic head $\mathbf{d}^i$
    Calculate: $\mathbf{C}_{ZD}^i = \frac{1}{N_r-1}\sum_{j=1}^{N_r}(\mathbf{z}_j - \bar{\mathbf{z}})(\mathbf{d}_j - \bar{\mathbf{d}})^{\mathrm{T}}$, the cross-covariance matrix
    Calculate: $\mathbf{C}_{DD}^i = \frac{1}{N_r-1}\sum_{j=1}^{N_r}(\mathbf{d}_j - \bar{\mathbf{d}})(\mathbf{d}_j - \bar{\mathbf{d}})^{\mathrm{T}}$,
     the auto covariance of the predicted head data
    Update: $\mathbf{z}^{i+1} = \mathbf{z}^i + \mathbf{C}_{ZD}^i(\mathbf{C}_{DD}^i + \alpha_i\mathbf{C}_D)^{-1}(\mathbf{A}_j^i)$
end

3.2. Results

3.2.1. Analysis of PGGAN conditioned to hard data

Figure 8 shows individual realizations for four scenarios obtained from the trained generator that was conditioned to hard data only. The channels from the individual realizations clearly show similarities to the channels in the reference image. From the hard data conditioned to the PGGAN, it can be seen that the generator considered the positions of the hard data and therefore generated realizations that honored the conditioned hard data. All conditioning points in all of Scenarios 1 to 4 show that their resulting realizations honor the hard data. It appears that as the hard data points increased from 9 to 81 the realizations more closely matched the reference image. Pictorially, it also appears that as the hard data increased, the variance was reduced, Figure 9. The variance of Scenario 4 (81 hard data points) was the lowest among the four scenarios. The variance was greatest for Scenario 1 (9 hard data). The diversity of the realizations for scenarios 1 to 4 also clearly illustrates that implementation of the mini-batch standard deviation worked in aiding the generator to produce diverse image realizations. It also appears from Figure 8 that the hard data were well conditioned on the generated images.

3.2.2. Influence of head conditioning data

We performed a sensitivity analysis of how an increase in hydraulic head data will influence the estimation of the ensembled hydraulic conductivities with respect to the reference hydraulic conductivity. For this analysis, the reference hydraulic conductivity was created by the generator of the PGGAN after it was trained. It was not randomly cut from the training image. Bao et al. (2020) have indicated that obtaining reference hydraulic conductivity data from the generator of the PGGAN to be used for parameter estimation will result in realizations close to the reference hydraulic conductivity after parameter estimation. Figures 10 and 11 show realizations, mean, and variance of hydraulic conductivities over 100 ensemble members for 9 and 25 observation wells, respectively, as well as a scenario with no observation data. The realizations show that, when the number of wells increased from 9 in Scenario 6 to 25 in Scenario 7 in case 2, the estimated hydraulic conductivities were improved and were close to the reference hydraulic conductivity. Statistically, when there were more measured data, the estimated parameters tended to approach the true parameter. The mean map of Scenario 7 with 25 wells was closer to the reference map than Scenario 6 with 9 wells. The variance map of Scenario 6 shows greater variance than in Scenario 7. The variance of Scenario 7, which is close to zero, indicates a near convergence of the ensembled hydraulic conductivities after optimization.

The simulated heads from Figure 12 were obtained from the forward modeling for each scenario. Three heads from each scenario were selected at the same positions to compare how close the simulated heads from the ensembled hydraulic conductivities were to the reference heads. The simulated hydraulic heads of Scenario 7 were closer to the reference hydraulic heads, which also indicates that conditioning with more head data aids in reaching convergence of the calibrated parameters and the reference parameters. The results of the RMSE and the Spread are shown in figure 13. Both the RMSE and the Spread show a downward trend, which means that the error and uncertainty significantly reduce when the head data is increased.

3.2.3. Random image as reference field

We looked at the case of extracting a random image from the training image that was not used for the training of the PGGAN, which is reasonable for real-world applications, in which the generator of the PGGAN will have little (scanty measured data) or no clue of the area under study. Therefore, testing the generator based on a new reference hydraulic conductivity field should be a useful way to test the PGGAN. However, (Bao et al., 2020) experimented on a conventional GAN conditioned to only hydraulic head data to estimate the hydraulic conductivity field, given the hydraulic conductivity field randomly cut from the reference image. Their results showed that the estimated hydraulic conductivity fields get closer to the reference hydraulic conductivity as the data points for hydraulic head increase but they still do not fully mimic the channels in the reference image.

In certain cases, some sort of measured data (hard data) can be acquired that will help in parameter estimations and reduce uncertainties. Figures 14 and 15 show realizations, mean, and variance of three scenarios analyzed when both hard data and hydraulic head data were conditioned, to estimate the hydraulic conductivity with respect to the reference hydraulic conductivity field randomly cut from the training image. Based on the three scenarios, it is clear that as both the hydraulic head data and the hard data increased, the estimated hydraulic conductivity fields were closer to the reference hydraulic conductivity. Scenario 10 (25 head data and 25 hard data) was the closest to the reference image, followed by Scenario 9 (9 head data and 25 hard data), and then Scenario 8 (9 head data and 9 hard data). This demonstrates that the estimation error and uncertainty are reduced as the number of conditioned data is increased. In all scenarios, the hard data were well conditioned, which affirms that they play a major role in estimating hydraulic conductivity fields that are close to the reference hydraulic conductivity. Another observation is that realizations from scenarios 9 and 10 were close which is evident from the head data plot in Figure 16. This is because the number of hard data points conditioned to the PGGAN was the same for both scenarios; therefore, when more hard data were conditioned to the PGGAN it was able to generate data that were closer to the true data of the particular model domain.

Figure 17 shows the RMSE and Spread. The uncertainty and the errors were reduced when the conditioned data were increased. It also indicates that the RMSE and the spread of scenarios 9 and 10 were closer, which is a result of the increased hard data.

4. Reference field outside the training data

The effectiveness of the proposed technique was further evaluated using different reference hydraulic conductivity fields from Strebelle (2002), which is not part of the training data used for the PGGAN. In contrast to the reference fields in cases 1, 2, and 3, which are oriented in a north-east to south-west direction, this reference field used in case 4 exhibit a north-south direction for the channels. The parameter estimation process involved 25 hard data points and hydraulic heads from 25 observation wells for scenario 11, and 49 hard data points and hydraulic heads from 49 observation wells for scenario 12. The results from generated realizations (Figure 18) illustrate that the approach produced hydraulic conductivity fields that closely resemble the reference data field. As the number of conditioned data increased, the realizations more closely matched the reference field. The results obtained from parameter estimation suggest that the method is capable of handling a variety of hydraulic conductivity fields with different statistical properties, extending beyond the specific hydraulic conductivity fields introduced during the training of the generator.

5. Conclusion

In this study, an approach to directly conditioning on hard data and head data when performing parameter estimations in hydrogeologic modeling by leveraging the deep learning method, PGGAN and data assimilation was proposed. First, the PGGAN was trained by introducing hard data at each resolution stage, with the aim that the generator of the PGGAN would generate facies model realizations that honored the hard data. After training, the generator was coupled with ES-MDA to perform parameter estimation, given a reference image either generated by the PGGAN or randomly cut from the training image. An ensemble of random uniform distribution was generated, and together with the hard data obtained at point locations from the reference image, the generator generated an ensemble of realizations, which in this work were hydraulic conductivity parameters. The parameters went through a forward model to produce hydraulic heads at specified observation wells. The ES-MDA method was applied to optimize the parameters of the latent space with the help of both the observed and modeled hydraulic head data. The results demonstrate that using PGGAN conditioned to hard data and coupled to ES-MDA can reconstruct a channelized aquifer of an unknown model domain when some measured data from the unknown aquifer domain are obtained. This also can reduce the uncertainty of hydraulic head predictions. Synthetic examples were used to test the method. A PGGAN that was not conditioned to hard data was trained in order to test its efficiency in generating images that mimic the channels of a reference image generated by the trained PGGAN when coupled with ES-MDA and conditioned to only head data. The results show that the PGGAN was efficient in reconstructing the channels of the reference image when conditioned with head data. Results also show that, as the head data are increased, the image more closely resembles the reference image.

The results from PGGAN conditioned to hard data and coupled with ES-MDA show that when the well facies data were increased, the generated image realizations were able to mimic the channel structures in the reference image and reduce the uncertainty of the estimation. This is evident in the RMSE and spread, calculated which displayed a significant reduction when the conditioned data were increased. Results also show that having some sort of hard data during parameter estimation helps to reduce uncertainty.

Acknowledgements This work has been supported through a grant from the National Science Foundation (OIA-1833069). The authors wish to thank three anonymous reviewers for their comments, which substantially helped to improve the final version of the manuscript.

References

Abdellatif, A., Elsheikh, A.H., Graham, G., Busby, D., Berthet, P., 2022. Generating unrepresented proportions of geological facies using generative adversarial networks. Computers & Geosciences 162, 105085. URL: https://www.sciencedirect.com/science/article/pii/S0098300422000474, doi:https://doi.org/10.1016/j.cageo.2022.105085.

Bao, J., Li, L., Davis, A., 2022. Variational autoencoder or generative adversarial networks? a comparison of two deep learning methods for flow and transport data assimilation. Mathematical Geosciences 54, 1017–1042.

Bao, J., Li, L., Redoloza, F., 2020. Coupling ensemble smoother and deep learning with generative adversarial networks to deal with non-gaissianity in flow and transport data assimilation. Journal of Hydrology 590, 125443.

Caers, J., Zhang, T., 2004. Multiple-point geostatistics: a quantitative vehicle for integrating geologic analogs into multiple reservoir models. AAPG Mem 80, 383–394.

Canchumuni, S.A., Emerick, A.A., Pacheco, M.A., 2017. Integration of ensemble data assimilation and deep learning for history matching facies models. In Proceedings of the Offshore Technology Conference, Rio de Janeiro, Brazil, 24–26 October, number OTC28015-MS .

Canchumuni, S.A., Emerick, A.A., Pacheco, M.A., 2018. History matching channelized facies models using ensemble smoother with a deep learning parameterization. In Proceedings of the 16th European Conference on the Mathematics of Oil Recovery (ECMOR XVI), Barcelona, Spain, 3–6 September .

Canchumuni, S.W., 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. URL: http://dx.doi.org/10.1016/j.cageo.2019.04.006, doi:10.1016/j.cageo.2019.04.006.

Chan, S., Elsheikh, A.H., 2017. Parametrization and generation of geological models with generative adversarial networks. URL: https://arxiv.org/abs/1708.01810.

Chan, S., Elsheikh, A.H., 2018. Parametric generation of conditional geological realizations using generative neural networks. URL: https://arxiv.org/abs/1807.05207.

Chen, Y., Oliver, D.S., 2012. Ensemble randomized maximum likelihood method as an iterative ensemble smoother. Mathematical Geosciences 44, 1–26.

Chen, Y., Zhang, D., 2006. Data assimilation for transient flow in geologic formations via ensemble kalman filter. Advances in Water Resources 29, 1107–1122.

Cressie, N., 1990. The origins of kriging. Math Geol 22, 239–252.

Dubrule, L.M.O., Blunt, M.J., 2017. Reconstruction of three-dimensional porous media using generative adversarial neural networks. doi:10.1103/PhysRevE.96.043309 96(043309), 1 – 6.

Dupont, E., Zhang, T., Tilke, P.and Liang, L., Bailey, W., 2018. Generating realistic geology conditioned on physical measurements with generative adversarial networks. URL: https://arxiv.org/abs/1802.03065.

Emerick, A., 2012. History Matching and Uncertainty Characterization: Using Ensemble-based Methods. LAP LAMBERT Academic Publishing.

Emerick, A.A., Reynolds, A.C., 2013. Ensemble smoother with multiple data assimilation. Computers & Geosciences 55, 3–15.

Evensen, G., 2018. Analysis of iterative ensemble smoothers for solving inverse problems. Computational Geosciences 22, 885–908.

Fetter, C.W., 2018. Applied hydrogeology. Waveland Press.

Fossum, K., Alyaev, S., Elsheikh, A., 2023. Ensemble history-matching workflow using interpretable spade-gan geomodel 2023, 1–5. URL: https://www.earthdoc.org/content/papers/10.3997/2214-4609.202335020, doi:https://doi.org/10.3997/2214-4609.202335020.

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

Goodfellow, Ian, Y.B., Courville, A., 2016. Deep learning http://www.deeplearningbook.org.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C., 2017. Improve training of wasserstein gans. arXiv:1704.00028 .

Harbaugh, A.W., Banta, E.R., Hill, M.C., McDonald, M.G., 2000. Modflow-2000, the u. s. geological survey modular ground-water model-user guide to modularization concepts and the ground-water flow process. Open-file Report. U. S. Geological Survey , 134.

Karras, T., Aila, T., Laine, S., Lehtinen, J., 2018. Progressive growing of gans for improved quality, stability, and variation. arXiv:1710.10196 .

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, 381–406.

Laloy, E., H´erault, R.and 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.

Liu, Y., Sun, W., Durlofsky, L.J., 2018. A deep-learning-based geological parameterization for history matching complex models. arXiv:1807.02716.

Mahmud, K., G.M.J.C.P., Baker, A., 2014. Simulation of earth textures by conditional image quilting. Water Resources Research 50, 3088–3107. Doi:10.1002/2013WR015069.

Mosser, L., Dubrule, O., Blunt, M.J., 2018. Conditioning of three-dimensional generative adversarial networks for pore and reservoir-scale models. arXiv:1802.05622 .

Ruffino, C., Hérault, R., Laloy, E., Gasso, G., 2020. Pixel-wise conditioned generative adversarial networks for image synthesis and completion. Neurocomputing 416, 218–230. URL: https://www.sciencedirect.com/science/article/pii/S0925231220305154, doi:https://doi.org/10.1016/j.neucom.2019.11.116.

Skjervheim, J.A., Evensen, G., et al., 2011. An ensemble smoother for assisted history matching, in: SPE Reservoir Simulation Symposium, Society of Petroleum Engineers.

Song, S., Mukerji, T., Hou, J., 2021. Gansim: Conditional facies simulation using an improved progressive growing of generative adversarial networks (gans). Mathematical Geosciences 53, 1413–1444.

Song, S., Zhang, D., Mukerji, T., Wang, N., 2023. Gansim-surrogate: An integrated framework for stochastic conditional geomodelling. Journal of Hydrology 620, 129493. URL: https://www.sciencedirect.com/science/article/pii/S0022169423004353, doi:https://doi.org/10.1016/j.jhydrol.2023.129493.

Stordal, A.S., Karlsen, H.A., Nævdal, G., Skaug, H.J., Vallès, B., 2011. Bridging the ensemble kalman filter and particle filters: the adaptive gaussian mixture filter. Computational Geosciences 15, 1573–1499. URL: https://doi.org/10.1007/s10596-010-9207-1.

Strebelle, S., 2002. Conditional simulation of complex geological structures using multiple-point statistics. Mathematical Geology 34, 21. URL: https://doi.org/10.1023/A:1014009426274, doi:https://doi.org/10.1023/A:1014009426274.

Tahmasebi, P., 2018. Multiple Point Statistics: A Review. Springer International Publishing, Cham. pp. 613–643. URL: https://doi.org/10.1007/978-3-319-78999-6_30, doi:10.1007/978-3-319-78999-6_30.

Zahner, T., Lochbühler, T., Mariethoz, G., Linde, N., 2015. Image synthesis with graph cuts: a fast model proposal mechanism in probabilistic inversion. Geophysical Journal International 204, 1179–1190. URL: https://doi.org/10.1093/gji/ggv517, doi:10.1093/gji/ggv517, arXiv:https://academic.oup.com/gji/article-pdf/204/2/1179/1621207/ggv517.pdf.

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, 541–549. URL: https://doi.org/10.1007/s12182-019-0328-4.

Zuo, C., Pan, Z., Yin, Z., Guo, C., 2022. A nearest neighbor multiple-point statistics method for fast geological modeling. Computers & Geosciences 167, 105208. URL: https://www.sciencedirect.com/science/article/pii/S0098300422001595, doi:https://doi.org/10.1016/j.cageo.2022.105208.

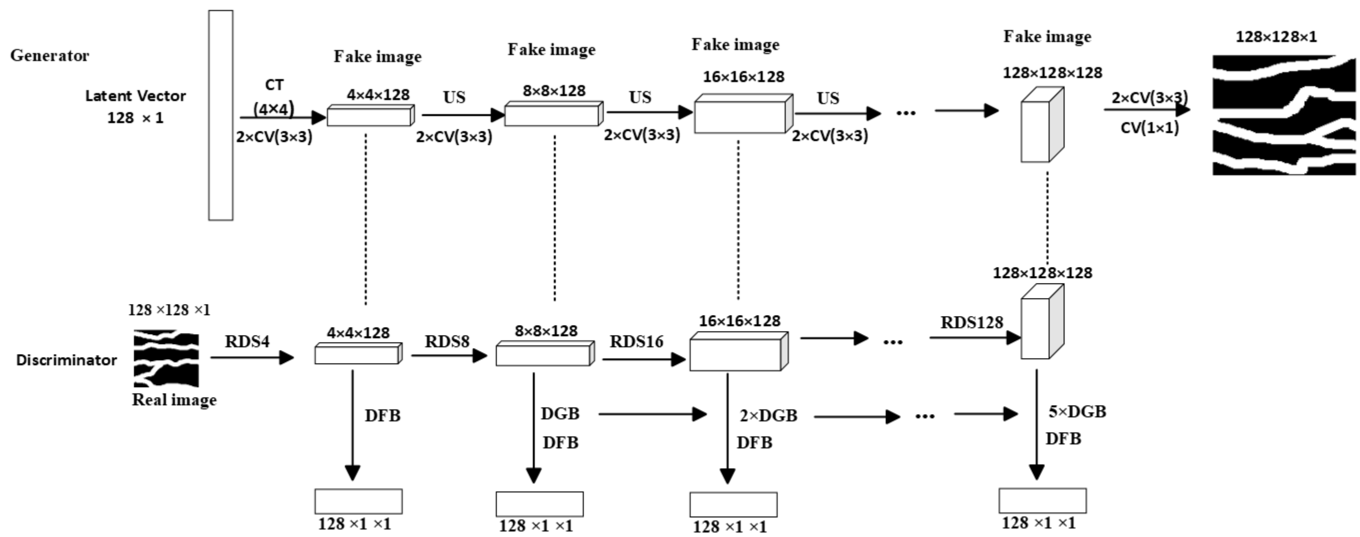

Figure 1: The architecture of the unconditional progressive growing of GAN. CT(4×4) represents transposed convolution with a kernel size of 4. US represents the upsampling layer used to upscale the images from low resolution to high resolution. RDS4, RDS8, RDS16, ..., RDS128 represents real images downsampled to (4×4),(8×8), (16×16), ..., (128×128) image sizes. DGB represents the general block of the discriminator which has two convolutional layers with (3×3) kernel size and a downsampling layer with a scale factor of 2. DFB is a final block of the discriminator (3×3), (4×4), (1×1) convolutional layers

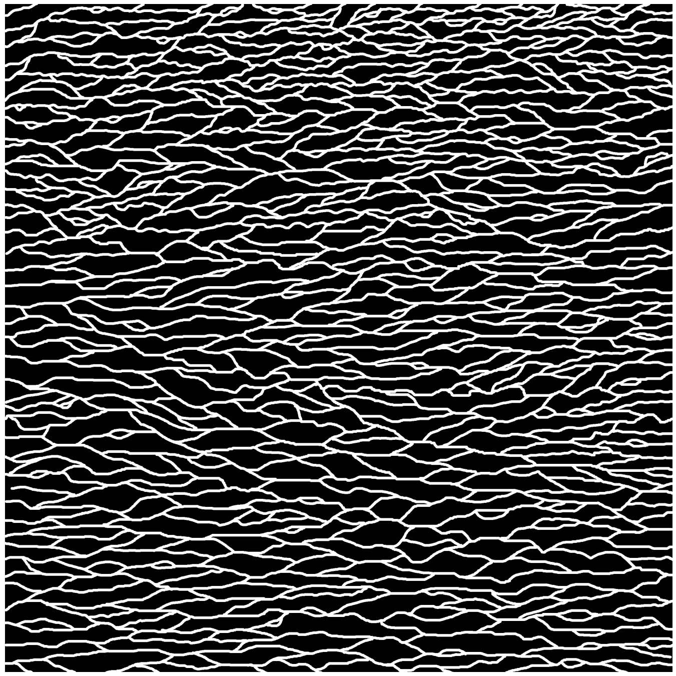

Figure 2: Training image (2500 × 2500).

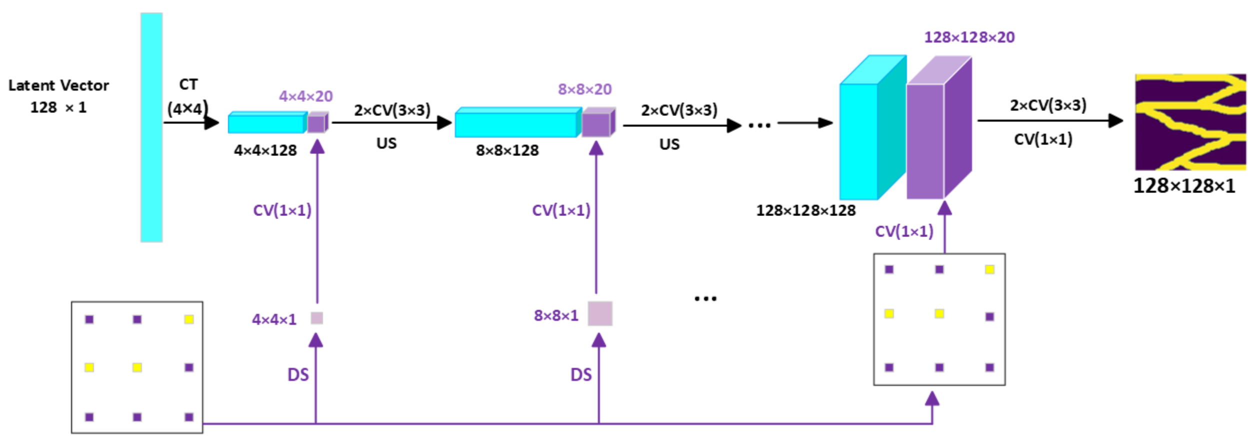

Figure 3: The architecture of the generator conditioned to hard data. CT(4×4) is a transposed convolution with (4×4) kernel sizes. CV(3×3) is a convolutional layer with (3×3) kernel size. CV(1×1) is a convolutional layer with (1×1) kernel size. US is an upsampling layer and DS is a downsampling layer

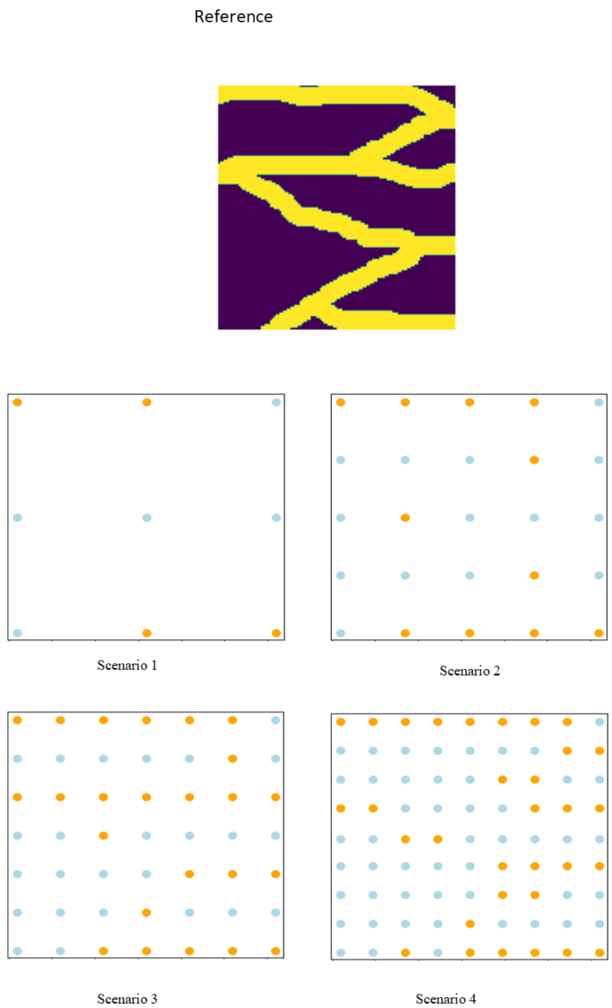

Figure 4: Well facies locations for case 1.

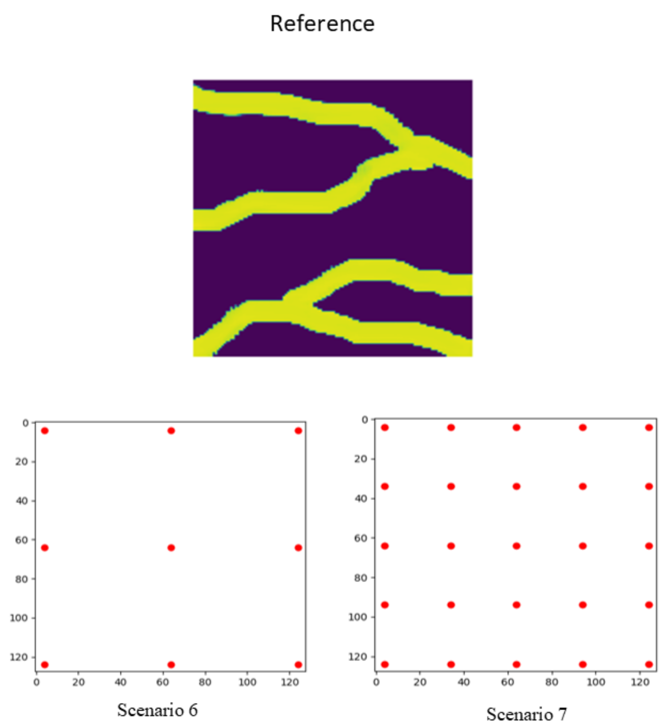

Figure 5: Observation well locations for case 2.

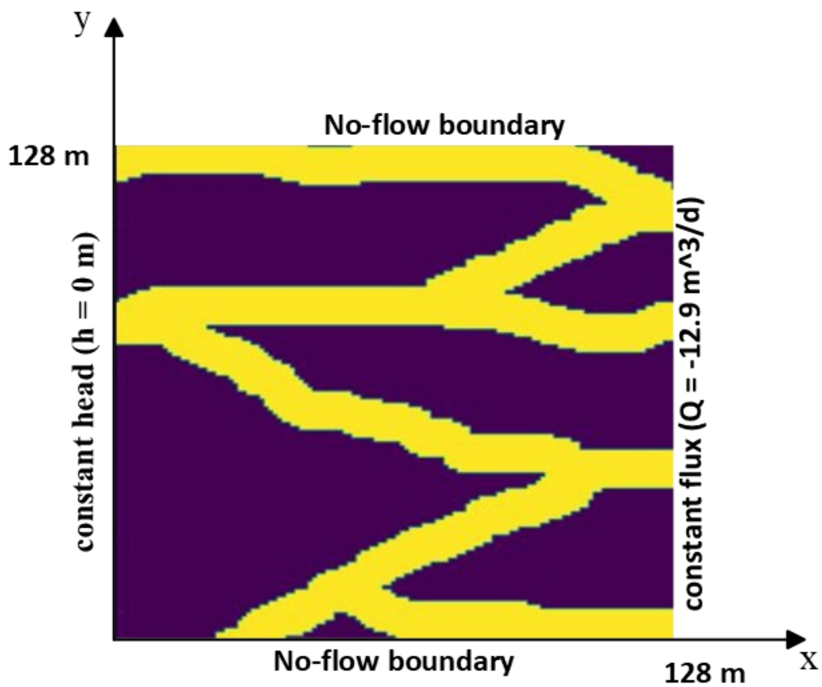

Figure 6: The reference hydraulic conductivities with boundary conditions. No-flow boundary on north and south, constant head on the western boundary, and constant flux on the eastern boundary. The yellow areas represent the channels with high hydraulic conductivity.

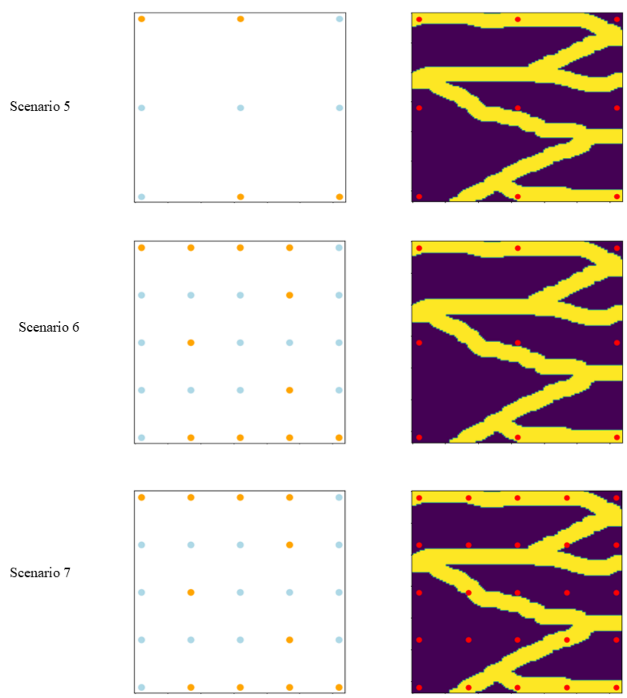

Figure 7: Well facies locations and observation well locations for case 3.

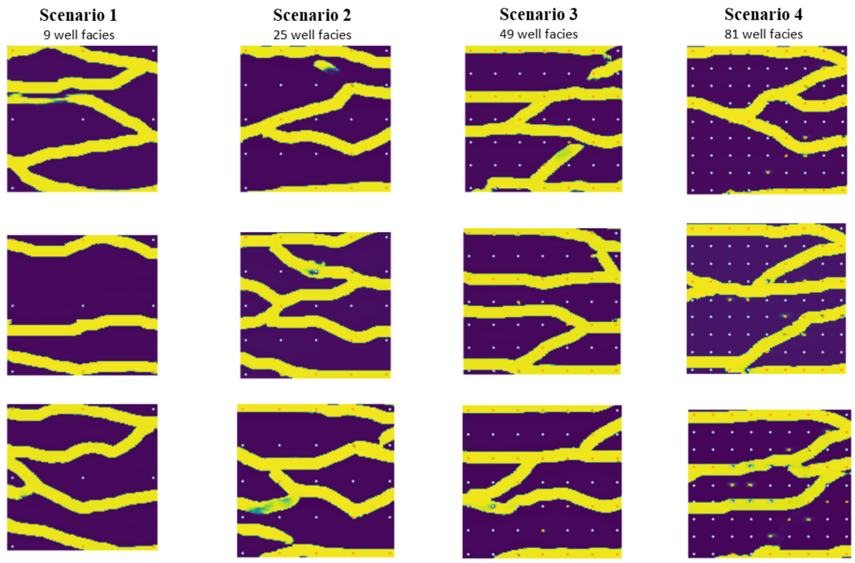

Figure 8: Realizations for PGGAN conditioned on hard data (well facies data).

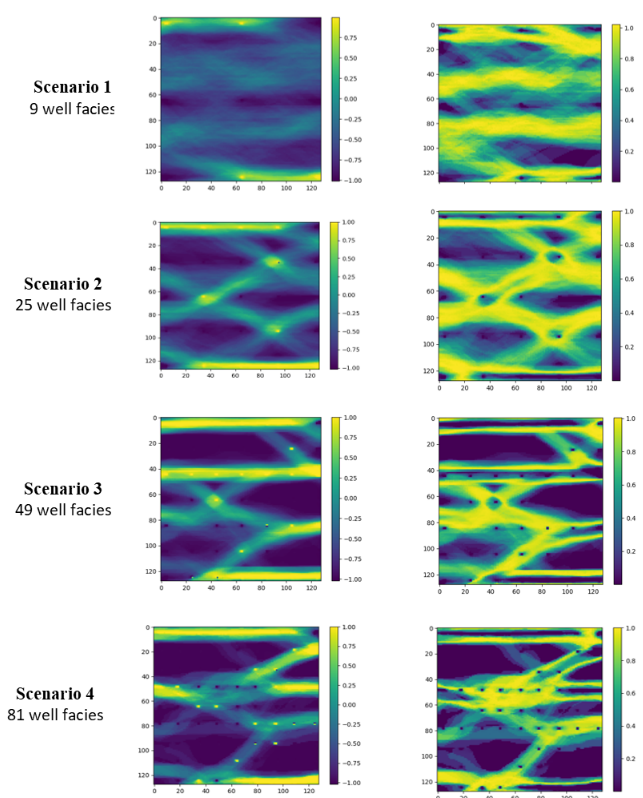

Figure 9: The mean and variance of hydraulic conductivity for case 1.

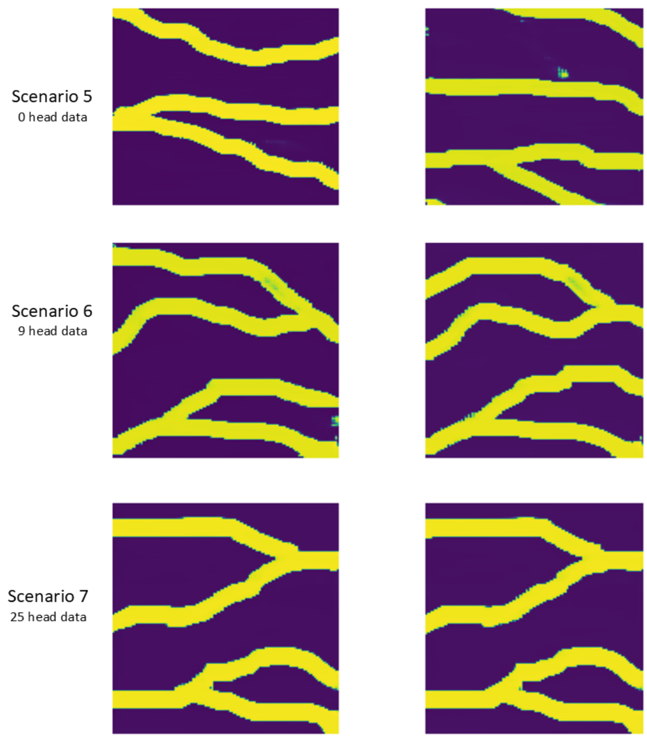

Figure 10: Realizations for PGGAN conditioned to head data.

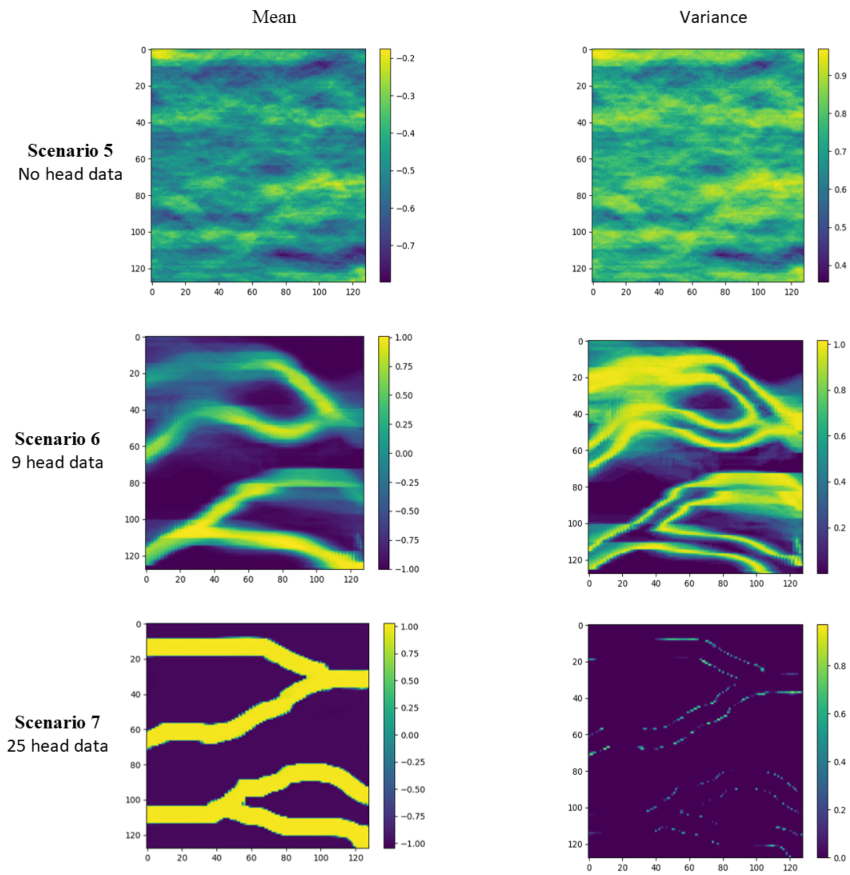

Figure 11: The mean and variance of hydraulic conductivity for case 2.

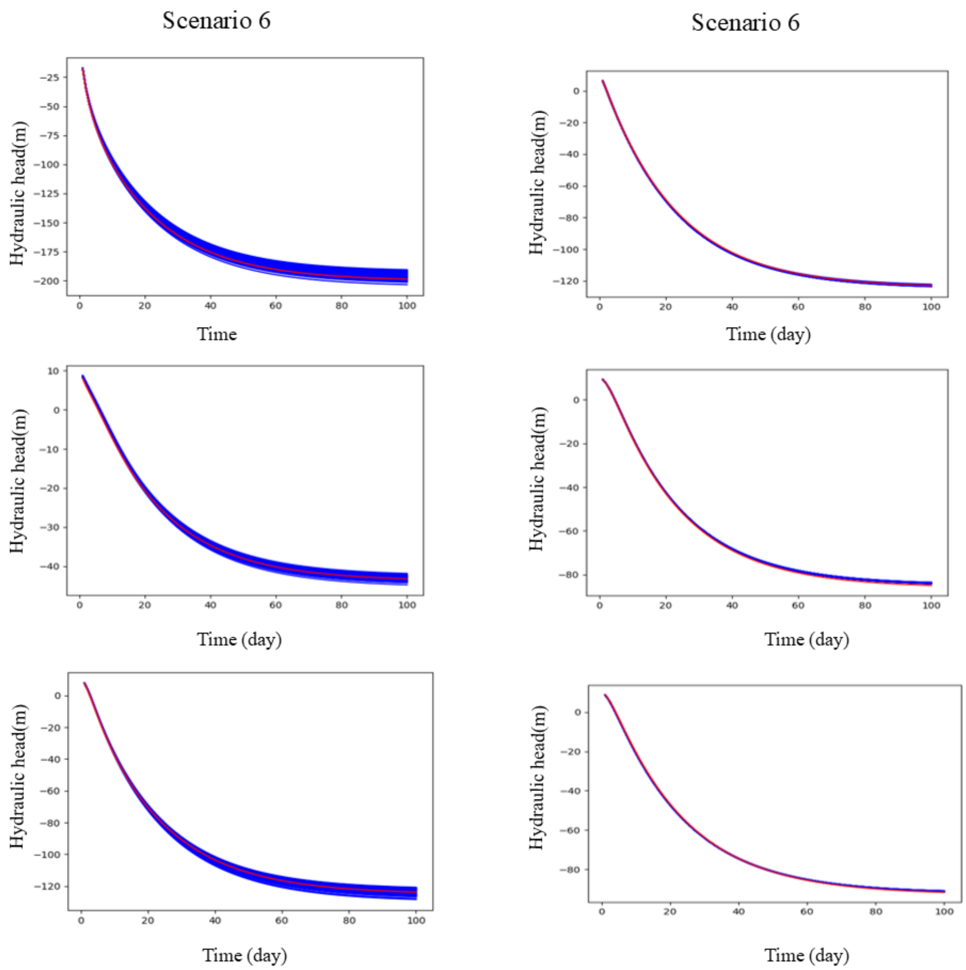

Figure 12: Hydraulic head for three observation well locations for case 2.

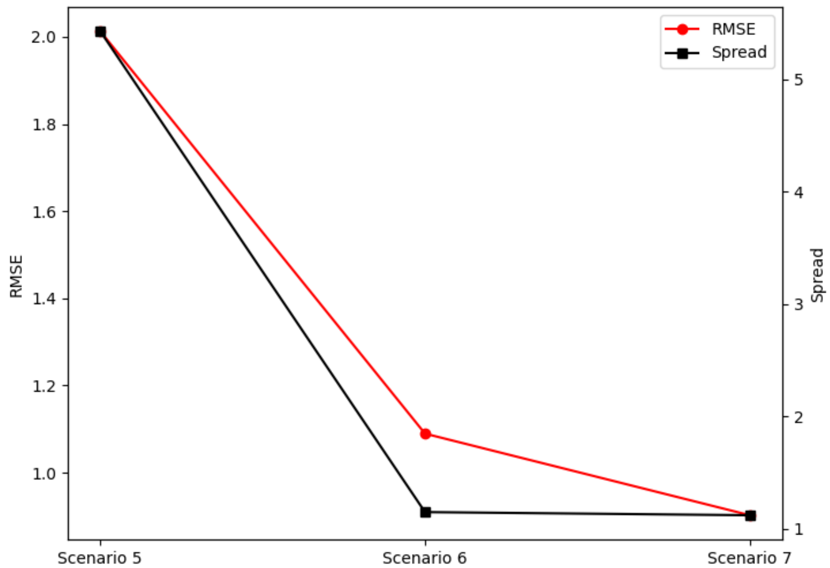

Figure 13: The RMSE and spread for case 2. The red line represents the RMSE between the reference and predicted hydraulic conductivity fields and the black line represents the spread of the predicted hydraulic conductivity fields.

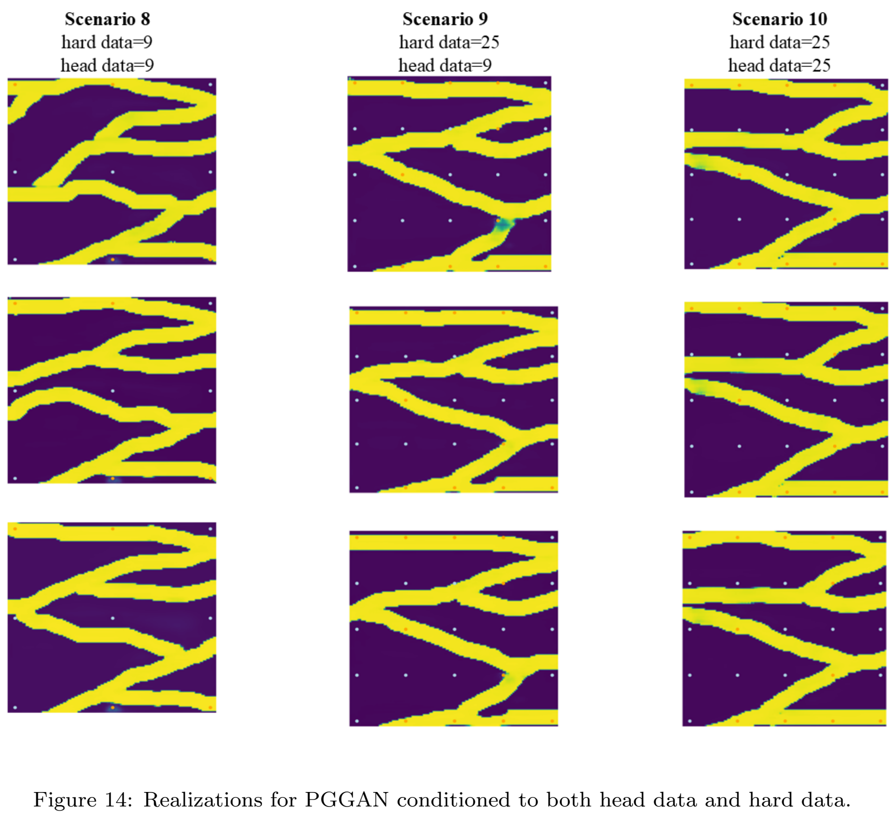

Figure 14: Realizations for PGGAN conditioned to both head data and hard data.

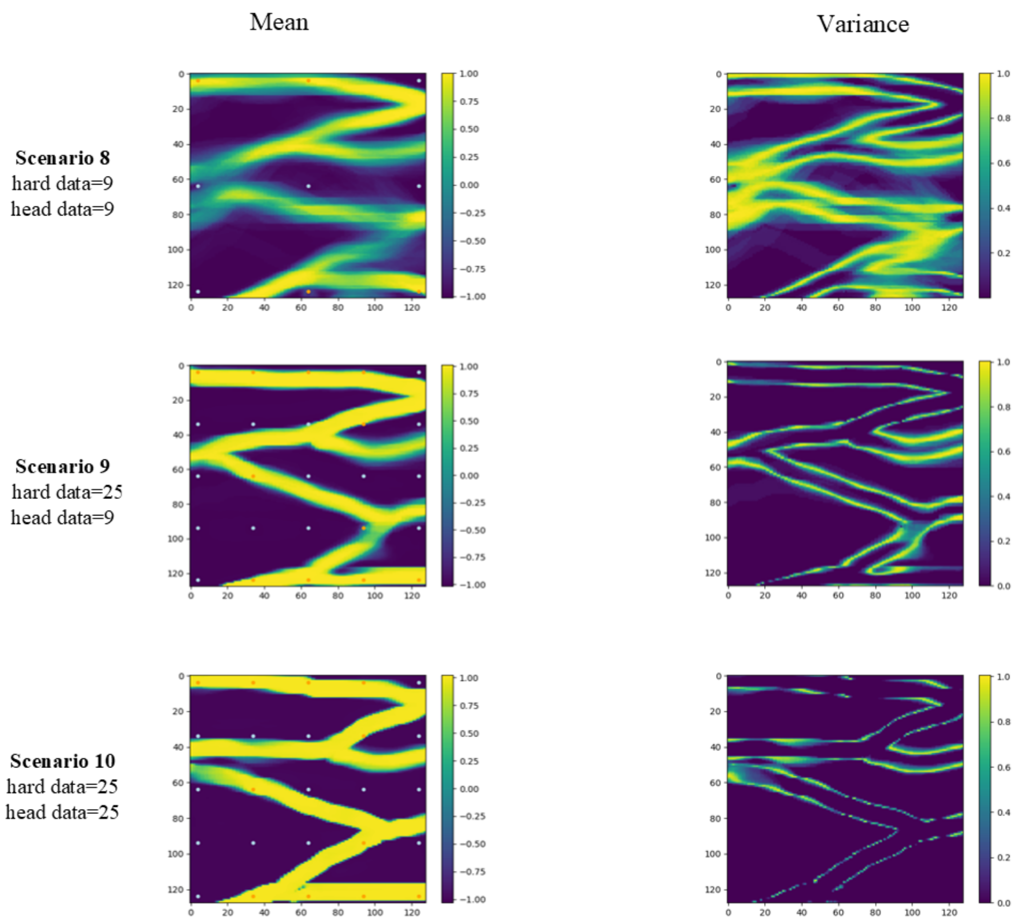

Figure 15: The mean and variance of hydraulic conductivity fields for case 3.

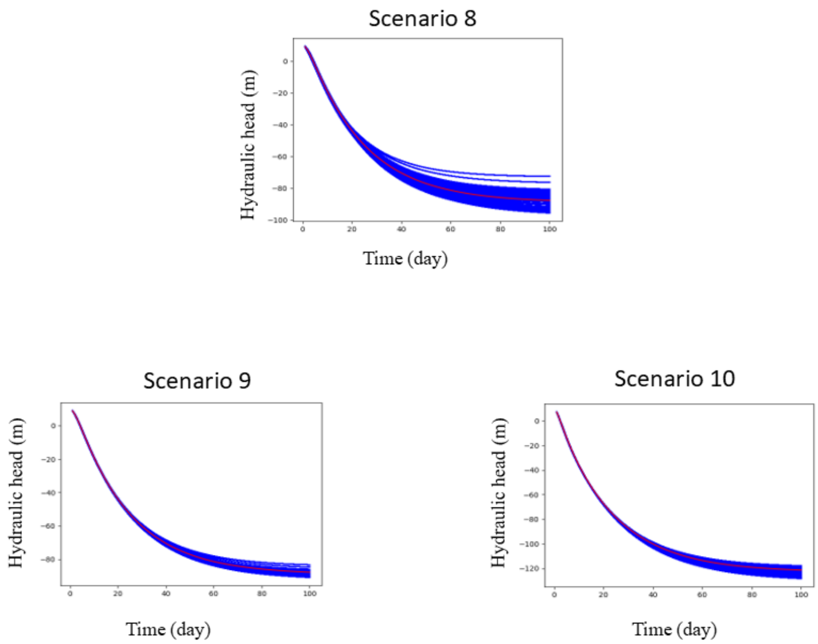

Figure 16: Hydraulic head of the middle observation wells for case 3.

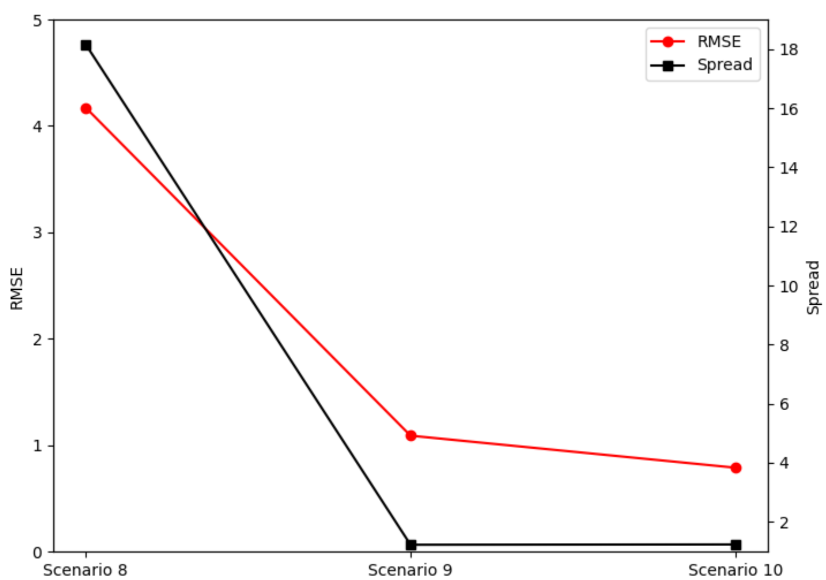

Figure 17: The RMSE and spread for case 3. The red line represents the RMSE between the reference and predicted hydraulic conductivity fields and the black line represents the spread of the predicted hydraulic conductivity fields.

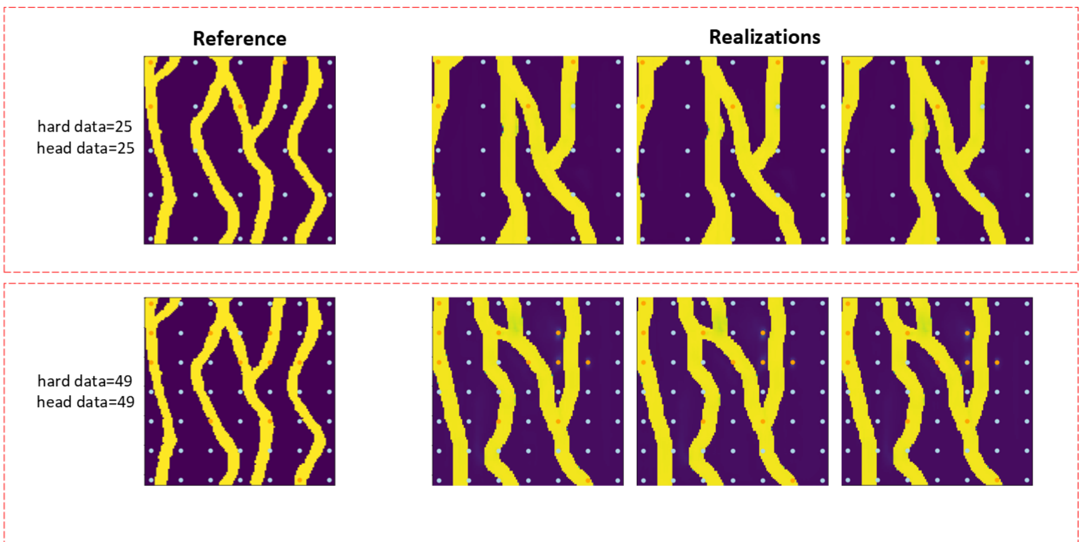

Figure 18: Realizations of hydraulic conductivity fields for case 4