Water Resources Research

RESEARCH ARTICLE
10.1029/2018WR024592

Predicting CO_2 Plume Migration in Heterogeneous Formations Using Conditional Deep Convolutional Generative Adversarial Network

Zhi Zhong[1], Alexander Y. Sun[1], and Hoonyoung Jeong[2]

[1]Bureau of Economic Geology, Jackson School of Geosciences, University of Texas at Austin, Austin, TX, USA, [2]Department of Energy Resources Engineering, College of Engineering, Seoul National University, Seoul, South Korea

Key Points:
- A machine learning-based surrogate model is proposed for predicting CO_2 plume migration
- The method is based on conditional deep convolutional generative adversarial network (cDC-GAN)
- cDC-GAN can facilitate the high-dimensional cross-domain learning and predict the CO_2 saturation with high accuracy at any time instance

Supporting Information:
- Supporting Information S1

Correspondence to:
Z. Zhong and A. Y. Sun,
zhi.zhong@beg.utexas.edu;
alex.sun@beg.utexas.edu

Citation:
Zhong, Z., Sun, A. Y., & Jeong, H. (2019). Predicting CO_2 plume migration in heterogeneous formations using conditional deep convolutional generative adversarial network. *Water Resources Research*, *55*, 5830–5851. https://doi.org/10.1029/2018WR024592

Received 14 DEC 2018
Accepted 5 JUN 2019
Accepted article online 11 JUN 2019
Published online 19 JUL 2019

Abstract Numerical simulation of flow and transport in heterogeneous formations has long been studied, especially for uncertainty quantification and risk assessment. The high computational cost associated with running large-scale numerical simulations in a Monte Carlo sense has motivated the development of surrogate models, which aim to capture the important input-output relations of physics-based models but require only a fraction of the cost of full model runs. In this work, we formulate a conditional deep convolutional generative adversarial network (cDC-GAN) surrogate model to learn the dynamic functional mappings in multiphase models. The cDC-GAN belongs to a class of semisupervised learning methods that can be used to learn the data generation processes. Like the original GAN, a main strength of the cDC-GAN is that it includes a self-training scheme for improving the quality of generative modeling in a game theoretic framework, without requiring extensive statistical knowledge and assumptions on input data distributions. In particular, our cDC-GAN model is designed to learn cross-domain mappings between high-dimensional input (e.g., permeability) and output (e.g., phase saturations) pairs, with the ability to incorporate conditioning information (e.g., prediction time). As a use case, we demonstrate the performance of cDC-GAN for predicting the migration of carbon dioxide (CO_2) plume in heterogeneous carbon storage reservoirs, which has both numerical and practical significance because of the safe storage requirements now mandated in many countries. Results show that cDC-GAN achieves high accuracy in predicting the spatial and temporal evolution patterns of the injected CO_2 plume, as compared to the original results obtained using a compositional reservoir simulator. The performance of cDC-GAN models, trained using the same number of training samples, stays relatively robust when the level of spatial heterogeneity is increased. Our cDC-GAN is pattern based and is not limited by the underlying physics. Thus, it provides a general framework for developing surrogate models, and for conducting uncertainty analyses for a wide range of physics-based models used in both groundwater and subsurface energy exploration applications.

1. Introduction

Subsurface resource exploration and management represent one of the main application areas in which understanding of the formation heterogeneity is critically important for resource production optimization and risk management (De Silva et al., 2016; Luo et al., 2013). Historically, flow and transport in heterogeneous formations have been extensively investigated in the subsurface modeling community, under topics such as stochastic hydrogeology (Dagan & Neuman, 2005; Gelhar, 1993; Rubin, 2003), data assimilation and inversion (Oliver & Chen, 2011; Schöniger et al., 2012; Sun & Sun, 2015; Zhou et al., 2011), uncertainty quantification (UQ; Tartakovsky, 2013; Zhang, 2001), and multiobjective optimization (Costa & Nannicini, 2018; Müller et al., 2013; Queipo et al., 2005). A main driving force behind many of these existing efforts is the increasing need to develop distributed high-resolution simulation models, on the one hand, while properly accounting for uncertainties in model structures and parameters, on the other (Wood et al., 2011). Risk assessment and UQ conducted in the Monte Carlo sense typically require sampling a high-dimensional input space and running a large number of forward simulations, which is computationally expensive, especially for large-scale dynamic simulation models. As a result, a large number of surrogate modeling techniques have been developed to reduce the computational burden.

The basic idea behind surrogate modeling is to find an alternative and yet computationally efficient and accurate approximation of the input-output relations simulated in a large-scale dynamic model (Agarwal

et al., 2014; Sun & Sun, 2015). In a slightly different definition, Lucia et al. (2004) defined the purpose of surrogate modeling as to "provide quantitatively accurate descriptions of the dynamics of systems at a computational cost much lower than the original numerical model and to provide a means by which system dynamics can be readily interpreted." So far, a large number of surrogate modeling techniques have been developed in the literature. Examples include (a) kriging (Gaussian process regression) and its variants that use a covariance-based method to interpolate a model's response surface (Kleijnen, 2009; Marrel et al., 2008); (b) the classical supervised machine learning (ML) methods (e.g., artificial neural networks and support vector regression) that often use a combination of nonlinear basis functions to approximate the relations between the input and output; (c) the proper orthogonal decomposition (POD) methods that represent a system's dynamics using a set of orthogonal basis functions obtained through the eigenanalysis of system snapshots (Lucia et al., 2004); and (d) stochastic polynomial chaos expansion (PCE) methods that approximate input-output relations using an expansion of orthonormal polynomials (Xiu & Karniadakis, 2002). Methods such as the kriging and classic machine learning methods generally do not scale well for large-scale dynamic models, unless model inputs and outputs are reparameterized through dimension reduction (Jeong et al., 2018; Sun & Durlofsky, 2017; Swischuk et al., 2018). By design, POD methods find a reduced-order representation of a deterministic model through subspace projection and, thus, can be applied to large-scale dynamic systems. Application of POD to uncertain inputs, however, is nontrivial. In contrast, the PCE methods are designed to approximate stochastic partial differential equations (PDE) and treat the uncertain inputs as random variables; these methods exploit regularity in the dependence of model outputs on the uncertain model inputs by solving a forward problem at a finite number of realizations of the random inputs (Huan & Marzouk, 2013; Ma & Zabaras, 2009; Sun et al., 2018). The number of polynomial terms required for accurate PCE approximation, however, increases rapidly with the number of stochastic dimensions and the order of expansion. Thus, reparameterization of model inputs using dimension reduction techniques is still necessary to constrain the number of stochastic dimensions (Li & Zhang, 2007; Ma & Zabaras, 2009; Sun et al., 2013; Zhang et al., 2015; Zeng et al., 2016). For example, Karhunen-Loève expansion (KLE) or principal component analysis (PCA) is commonly used as a parameterization technique to reduce the dimensionality of random fields (Zhang & Lu, 2004).

In the last several years, the generative adversarial network (GAN) models have attracted wide attention in the artificial intelligence community (Arjovsky et al., 2017; Goodfellow et al., 2014; Isola et al., 2017; Mirza & Osindero, 2014). The original GAN (often called the vanilla GAN) introduced by Goodfellow et al. (2014) is a type of generative model that can generate samples following the distribution of input data (i.e., training samples). Specifically, the design of the vanilla GAN is set in a game theoretic framework that involves two competing players, a generator and a discriminator. The job of the generator is to transform samples from a low-dimensional latent space to samples of the variable of interest, which may potentially exist in a high-dimensional space. The discriminator is a classifier that tries to tell whether a sample is from the generator (fake sample) or from the training data (real sample). By training the generator and discriminator adversarially, GAN builds a generator that can create high-quality fake samples that cannot be distinguished from the real data by the discriminator. In essence, the GAN models include a self-training generative modeling mechanism, providing an attractive alternative for learning the input data distribution without requiring extensive statistical knowledge (e.g., needed for specifying parametric distributions) and code modifications from the end users. With the rapid advance in deep learning training techniques and computing hardware in recent years, the capacity of deep learning networks has also increased significantly, showing superior performance in learning hierarchical feature representations in various image analysis problems. Recognizing the strong linkage between image analyses and the input-output fields generated by distributed models, significant interests now exist in replicating the success of GAN achieved in learning the cross-domain image patterns to learning complex input-output dynamic mappings embedded in large-scale numerical models.

In subsurface modeling, GAN has recently been used to generate stochastic realizations of facies or permeability field from multimodal, high-dimensional distributions (Chan & Elsheikh, 2017; Dupont et al., 2018; Laloy et al., 2017), which is challenging for conventional generative modeling approaches used in geosciences. For example, the Markov chain Monte Carlo (MCMC) method has long been used as a generative model to approximate probability distributions of random variables, but the standard MCMC algorithms are mainly suitable for relatively low-dimensional parameters (Cotter et al., 2013). In Laloy et al. (2017), a spatial GAN is trained to generate 2-D and 3-D unconditional realizations of random parameter

fields and then MCMC is applied on a low-dimensional latent space for Bayesian inversion. Alternatively, using the end-to-end (or image-to-image) generative models, GANs can be trained directly to learn dynamic mappings between a pair of high-dimensional model input and output domains. In Sun (2018), a state-parameter identification GAN (SPID-GAN) is trained to learn the forward and reverse mappings between the high-dimensional model parameters and model states and is demonstrated for a single-phase, groundwater flow problem; the forward mapping learned by the GAN is essentially a surrogate model of the forward simulator. In Zhu and Zabaras (2018), a fully convolutional encoder-decoder network is designed to capture the complex forward mapping between the high-dimensional input (permeability) and output fields (pressure) through end-to-end learning. On the basis of Zhu and Zabaras (2018), a deep convolutional encoder-decoder neural network is used in Mo et al. (2018) to develop surrogate models of dynamic multiphase flow models. In general, these recent studies suggest that the new image-to-image deep learning methods are promising, yielding impressive results in terms of predictive performance and uncertainty modeling, even with limited training data. The existing works either focused on learning the bidirectional input-output mappings at a single simulation time step (Sun, 2018) or did not leverage the strengths of GAN for cross-domain learning (Laloy et al., 2017; Mo et al., 2018; Zhu & Zabaras, 2018). In this work, we formulate a conditional deep convolutional GAN (cDC-GAN) to learn the dynamic mappings between high-dimensional model inputs and outputs in a multiphase model and then apply the cDC-GAN to predicting CO_2 plume migration in heterogeneous carbon storage reservoirs.

Carbon capture and storage (CCS) is being investigated globally as a geoengineering technology for helping transition from the current fossil fuel dominant economy to a low-carbon economy. Leakage from geological carbon storage reservoirs is a priori nonzero because of the existence of natural faults and/or abandoned wells (Lewicki et al., 2007; Sun et al., 2013, 2018). Thus, high-fidelity simulation models, in conjunction with comprehensive site characterization and monitoring, are required to predict the long-term fate of the injected CO_2 and to demonstrate the secure containment of the CO_2 plume, with reasonable consideration of site uncertainty. Simulating CO_2 flow and transport behavior in porous media is difficult because of the interplay among phase change, composition, and reservoir heterogeneity (Doughty & Pruess, 2004; Jiang, 2011; Zhong & Carr, 2019). The computational costs associated with simulating these aspects of CCS can be prohibitive, necessitating the use of surrogate models. Although a large number of surrogate modeling studies have been conducted for CCS in the context of risk assessment, sensitivity analysis, UQ, and monitoring network design (Dai et al., 2018; Jeong & Srinivasan, 2016, 2017; Keating et al., 2016; Oladyshkin et al., 2011; Pawar et al., 2015; Sun et al., 2013, 2018), development of high-fidelity surrogate models remains a challenging subject in the high-dimensional decision space.

In the following, we first present the design of cDC-GAN and then demonstrate its performance in training a surrogate model for predicting the CO_2 plume migration in heterogeneous formations. Although our application area in this study focuses on CCS, the deep-learning-based approach proposed here has practical implications for many other surface and subsurface modeling problems that call for the use of high-fidelity surrogate models. This paper is organized as follows. In section 2, the general GAN framework is briefly reviewed and the formulation of cDC-GAN for dynamic surrogate modeling is described. In section 3, CO_2 injection into a hypothetical brine aquifer is considered and the cDC-GAN is used to predict the shape of CO_2 plumes at different times. Finally, conclusions are provided in the last section.

2. Material and Methods

2.1. GAN

The vanilla GAN consists of a generator and a discriminator that are in competition with each other. Let $\mathcal{G}(\mathbf{z};\boldsymbol{\theta}_g)$ denote a generator for a random variable (vector) $\mathbf{x}$, with $\mathbf{z}$ and θ_g representing its input and model parameters, respectively. Let $p_{\mathcal{G}}(\mathbf{x};\boldsymbol{\theta}_g)$ denote the distribution of the outputs produced by the generator. Similarly, let $\mathcal{D}(\mathbf{x};\boldsymbol{\theta}_d)$ denote a discriminator that takes $\mathbf{x}$ as input and is specified by a set of parameters θ_d. The goal of the generator $\mathcal{G}(\mathbf{z};\boldsymbol{\theta}_g)$ is to learn the distribution of the training data $p_{\text{data}}(\mathbf{x})$ and to generate samples that are as genuine as possible, namely, making $p_{\mathcal{G}}(\mathbf{x};\boldsymbol{\theta}_g)$ as close as possible to $p_{\text{data}}(\mathbf{x})$. The goal of the discriminator $\mathcal{D}(\mathbf{x},\boldsymbol{\theta}_d)$ is to determine whether a sample $\mathbf{x}$ is generated from $p_{\text{data}}(\mathbf{x})$ or from $p_{\mathcal{G}}(\mathbf{x};\boldsymbol{\theta}_g)$, and to assign probabilities accordingly, namely,

$$\mathcal{D}(\mathbf{x},\boldsymbol{\theta}_d) = \begin{cases} 1, & \text{if } \mathbf{x} \sim p_{\text{data}}(\mathbf{x}) \\ 0, & \text{if } \mathbf{x} \sim p_{\mathcal{G}}(\mathbf{x};\boldsymbol{\theta}_g) \end{cases}, \tag{1}$$

The training of GANs typically involves solving a minimax optimization problem in the game theoretic framework (see subsections below). At convergence, the discriminator is maximally confused, and cannot distinguish fake samples produced by the generator from the real data from $p_{\text{data}}(\mathbf{x})$, meaning $\mathcal{D}(\mathbf{x}, \boldsymbol{\theta}_d)$ predicts with a probability of 0.5 for all inputs (Goodfellow et al., 2014).

Sampling directly from $p_{\mathcal{G}}(\mathbf{x}; \boldsymbol{\theta}_g)$ can be difficult when the dimensionality of input space is high. In the vanilla GAN proposed by Goodfellow et al. (2014), the generator learns a mapping from a latent space $\mathbf{z}$ to the sample space of $\mathbf{x}$ in order to generate samples from $p_{\mathcal{G}}(\mathbf{x}; \boldsymbol{\theta}_g)$:

$$p_{\mathcal{G}}(\mathbf{x}; \boldsymbol{\theta}_g) = \int_{\mathbf{z}} p(\mathbf{z}) I_{[\mathcal{G}(\mathbf{z}; \boldsymbol{\theta}_g)=\mathbf{x}]} \mathrm{d}\mathbf{z}, \tag{2}$$

where I is an indicator function that assumes a value of 1 if $\mathcal{G}(\mathbf{z}; \boldsymbol{\theta}_g) = \mathbf{x}$ and is 0 otherwise. The approach exploits the idea that the data-generating process concentrates near a low-dimensional region (or manifold), and the representation of which can be efficiently learned by the generator. Note that the notion of transformation mapping from a latent space to a high-dimensional sample space is not new and is actually the basis of many random field generators and dimension reduction methods, such as the PCA and kernel PCA (Ma & Zabaras, 2011; Sarma et al., 2007).

Existing GANs approach the low-dimensional representation problem in two ways: (a) training a GAN to learn a latent space to data space mapping (Goodfellow et al., 2014; Laloy et al., 2017) or (b) training a GAN to learn a high-dimensional image-to-image (or end-to-end) mapping, but implicitly assuming that a low-dimensional representation exists and can be efficiently learned by the GAN (Isola et al., 2017; Sun, 2018; Zhu et al., 2017). We follow the latter approach in this work (see also section 2.2).

A critical issue in the design of GAN algorithms is how to measure the distance between the distribution $p_{\mathcal{G}}(\mathbf{x}; \boldsymbol{\theta}_g)$ and $p_{\text{data}}(\mathbf{x})$ accurately and reasonably. In terms of distance between two probability distributions, an inappropriate measure may lead to mode collapse (Arjovsky et al., 2017). In the subsections below, we briefly introduce the Kullback-Leibler (KL) divergence and Jensen-Shannon (JS) divergence measures, which are used in training the vanilla GAN, followed by a description of the optimization problem used to train $\mathcal{G}(\mathbf{z}; \boldsymbol{\theta}_g)$ and $\mathcal{D}(\mathbf{x}, \boldsymbol{\theta}_d)$. For clarity, we shall omit the dependence on GAN parameters θ_g and θ_d in the following discussion where no confusion should occur.

2.1.1. KL and JS Divergence

The KL divergence $D_{\text{KL}}(p\|q)$ is a measure of how a probability distribution $p(\mathbf{x})$ differs from a reference distribution $q(\mathbf{x})$. One of the most important properties of KL divergence is that $D_{\text{KL}}(p\|q) \geq 0$, where the equal sign holds if and only if the two distributions in question are identical. If $\mathbf{x} \in \mathbf{X}$ is a discrete random variable, $D_{\text{KL}}(p\|q)$ is defined by

$$D_{\text{KL}}(p(\mathbf{x})\|q(\mathbf{x})) = \sum_{\mathbf{x}\in\mathbf{X}} p(\mathbf{x}) \log\left(\frac{p(\mathbf{x})}{q(\mathbf{x})}\right), \tag{3}$$

When $\mathbf{x}$ is a continuous random variable, $D_{\text{KL}}(p\|q)$ is defined by the following integral:

$$D_{\text{KL}}(p(\mathbf{x})\|q(\mathbf{x})) = \int_{\mathbf{x}} p(\mathbf{x}) \log \frac{p(\mathbf{x})}{q(\mathbf{x})} \mathrm{d}\mathbf{x}, \tag{4}$$

By definition, the KL divergence is asymmetric. An alternative to the KL divergence is the JS divergence, which may be considered a symmetric and smoothed version of the KL divergence and is defined by

$$JS(p(\mathbf{x})\|q(\mathbf{x})) = \frac{1}{2} D_{KL}(p(\mathbf{x})\|m(\mathbf{x})) + \frac{1}{2} D_{\text{KL}}(q(\mathbf{x})\|m(\mathbf{x})), \tag{5}$$

where $m(\mathbf{x}) = \frac{1}{2}(p(\mathbf{x}) + q(\mathbf{x}))$. JS divergence varies in $[0, 1]$, with a value of 0 indicating that the two distributions in question are identical.

Assume we have a set of M training samples $\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^M$ that follow the data distribution $p_{\text{data}}(\mathbf{x})$, and let $P_{\mathcal{G}}(\mathbf{x}^i)$ denote the probability of a sample $\mathbf{x}^i$ actually coming from the generator output distribution $p_{\mathcal{G}}(\mathbf{x})$. The likelihood function of all training samples is given by

$$L = \prod_{i}^{M} P_{\mathcal{G}(\mathbf{x}^i)}, \tag{6}$$

which gives the probability of all M samples being from the generator distribution $p_{\mathcal{G}}(\mathbf{x})$. The greater the value of L is, the higher the chance that $p_{\mathcal{G}}(\mathbf{x})$ approaches $p_{\text{data}}(\mathbf{x})$. To maximize the likelihood, the following optimization problem is solved:

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}_g} \prod_i^M P_{\mathcal{G}}(\mathbf{x}^i). \tag{7}$$

It can be shown that the maximum likelihood estimation problem in equation (7) is equivalent to the following minimization problem on the KL divergence (Goodfellow et al., 2014):

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}_g} D_{\text{KL}}(p_{\mathcal{G}}(\mathbf{x}) \| p_{\text{data}}(\mathbf{x})), \tag{8}$$

which would recover $p_{\text{data}}(\mathbf{x})$ exactly if it lies within the family of distributions covered by the generator distribution, $p_{\mathcal{G}}(\mathbf{x})$ (Goodfellow et al., 2014).

2.1.2. Loss Function of GAN

The vanilla GAN solves a minimax problem shown below (Goodfellow et al., 2014)

$$\arg\min_{\mathcal{G}} \max_{\mathcal{D}} V(\mathcal{G}, \mathcal{D}), \tag{9}$$

$$\begin{aligned} V(\mathcal{G}, \mathcal{D}) &= \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})} \log(\mathcal{D}(\mathbf{x})) + \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})} \log(1 - \mathcal{D}(\mathcal{G}(\mathbf{z}))), \\ &= \int_{\mathbf{x}} p_{\text{data}}(\mathbf{x}) \log(\mathcal{D}(\mathbf{x})) \mathrm{d}\mathbf{x} - \int_{\mathbf{z}} p(\mathbf{z}) \log(1 - \mathcal{D}(\mathcal{G}(\mathbf{z}))) \mathrm{d}\mathbf{z}, \end{aligned} \tag{10}$$

where $V(\mathcal{G}, \mathcal{D})$ is the loss function and $\mathbb{E}$ denotes the expectation operator. Solution of equation (9) involves maximization of discriminator parameters θ_d in the inner loop and minimization of generator parameters θ_g in the outer loop. In a game theoretic framework, it can be shown that learning in the sense of equation (9) resembles minimizing the JS divergence between the data and the model distribution, and the minimum of the loss function in equation (9) is achieved if and only if $p_{\mathcal{G}}(\mathbf{x}) = p_{\text{data}}(\mathbf{x})$, at which point the theoretical value of the optimum is $-\log 4$ (Goodfellow et al., 2014). In actual implementation, a two-step process is often taken to train the generator and discriminator networks iteratively, in which the parameters of the generator are fixed when parameters of the discriminator are being optimized, and vice versa (Goodfellow et al., 2016).

2.2. The cDC-GAN

The vanilla GAN has demonstrated exceptional performance in solving certain unsupervised and semisupervised learning problems (see a recent review by Goodfellow, 2016). However, training of the vanilla GAN is challenging due to the lack of constraints, mode collapse (i.e., the generator produces very similar samples for different inputs), and the discriminator converging too quickly to zero (Goodfellow, 2016; Isola et al., 2017; Mirza & Osindero, 2014). Those drawbacks motivate the development of a number of variants of the vanilla GAN, which are proposed to overcome the shortcomings of the vanilla GAN for different application domains, such as image-to-image translation (Isola et al., 2017), image segmentation (Long et al., 2015), video generation (Baddar et al., 2017; Vondrick et al., 2016), and cross-domain learning (Zhu et al., 2017).

Instead of just using a vector $\mathbf{z}$ (which may denote either a latent space vector or a high-dimensional image in the input domain), the conditional generative adversarial network (cGAN) proposed by Mirza and Osindero (2014) also facilitates the inclusion of additional conditions as inputs to train the generator, which is shown by the authors to improve the convergence of GAN significantly. Most of the state-of-the-art GAN models use convolutional layers as building blocks, which represent the inputs as a hierarchy of feature maps. For example, in Radford et al. (2015) a deep convolutional GAN (DC-GAN) is proposed to generate images from latent vectors by applying autoencoding and autodecoding techniques. The encoder-decoder design, together with other deep learning constructs (e.g., batch normalization layers), helps the generator and discriminator to learn downsampling and upsampling operations and improves the training stability of GAN models (Radford et al., 2015).

In this work, we adopt a conditional deep convolutional GAN (cDC-GAN) architecture that combines the strengths of cGAN and DC-GAN for image-to-image mapping. The loss function of the vanilla GAN in

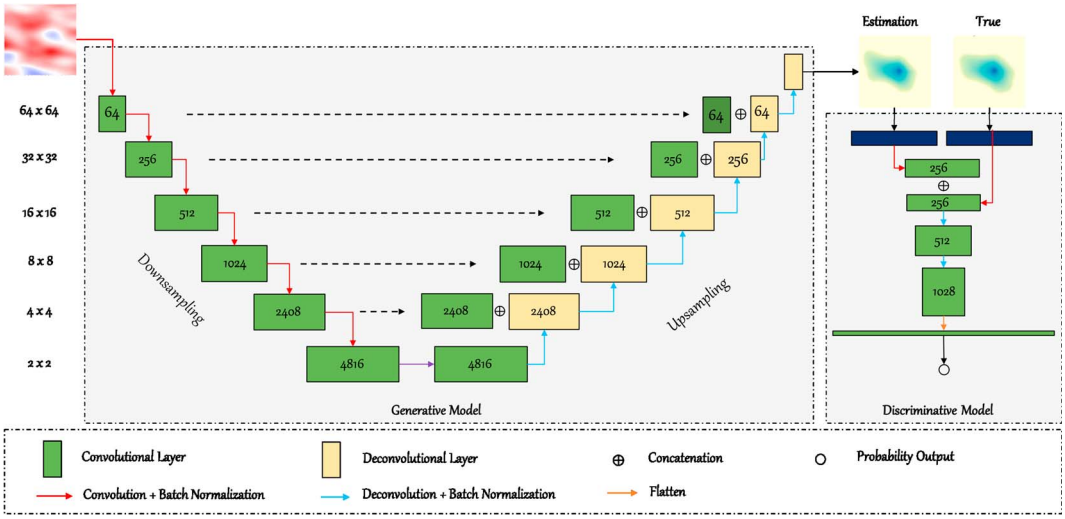

Figure 1. Design of cDC-GAN. The number of filters of each convolutional layer is shown in the green boxes, while the number of filters of the deconvolutional layer is twice the corresponding convolutional layer connected by the dashed line. The size of feature map corresponding to each level is shown on the left side of this figure. Input data is permeability map, output is simulated CO_2 saturation map, and the time associated with each output map is provided as conditioning data. The generator performs image-to-image translation, while the discriminator is a classifier that gives the probability that the generator output is from the data distribution.

equation (10) may be converted to a conditional form to accommodate any additional information that is helpful for training a GAN (Mirza & Osindero, 2014),

$$V(\mathcal{G},\mathcal{D}) = \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})} \log \mathcal{D}(\mathbf{x}|\mathbf{y}) + \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})} \log(1 - \mathcal{D}(\mathcal{G}(\mathbf{z}|\mathbf{y})), \tag{11}$$

where the conditioning data $\mathbf{y}$ may represent auxiliary information in either scalar or vector forms.

In the cDC-GAN design, the generator $\mathcal{G}(\mathbf{z}|\mathbf{y})$ and discriminator $\mathcal{D}(\mathbf{x}|\mathbf{y})$ share a similar structure as used in the pix2pix (i.e., image-to-image mapping) work (Isola et al., 2017), which includes a series of convolutional and deconvolutional layers to help discover high-level features at multiple scales (see Figure 1). For our demonstration case study, the sizes of input and output images are both set to 128×128. We assume that the permeability field represents the main source of uncertain model input. The input data ($\mathbf{z}$) to cDC-GAN is thus a permeability map (illustrated as the filled contour map under Input in the upper-left corner of Figure 1), and the target data ($\mathbf{x}$) includes simulated CO_2 saturation maps at different times (note the terms plume and saturation map are used interchangeably in the following discussion). To enable the GAN-based surrogate model to predict model outputs at different steps, we use the time step as conditioning data ($\mathbf{y}$) during training, which is represented as a constant valued image, as shown by the solid color map under Input in Figure 1. In this case, the time step provides additional information for cDC-GAN to learn the input-output dynamic mappings.

The design of the generator follows the U-net design originally proposed by Ronneberger et al. (2015). The kernels for both convolutional layers (used in the downsampling path) and deconvolutional layers (used in the upsampling path) are 4×4 with a stride size of 2. The number of filters for different layers increases from 256 to 2,048 in the downsampling path, allowing the encoder to learn low-level, fine-grained feature maps. The flow is then reversed in the upsampling path, where the coarse-grained feature maps from the decoder are combined with fine-grained feature maps from the encoder through skip connections (indicated by the dashed line in Figure 1). The leaky rectified linear unit function (leakyReLU) with a leaky value of 0.02 is used as the activation function for all hidden layers, namely,

$$f(x) = \begin{cases} x, & \text{if } x > 0 \\ 0.02x, & \text{otherwise} \end{cases} \tag{12}$$

The output layer uses the hyperbolic tangent function (tanh) as the activation function to recover continuous values, and generates an output image having the same size as the input image.

During training, the generative model and discriminative model are optimized iteratively in alternating steps. Batch normalization is used on both generative and discriminative models to stabilize training (Salimans & Kingma, 2016). In this work, we apply alternating optimization of the discriminator network $\mathcal{D}$ and generator network $\mathcal{G}$ by five steps and one step, respectively. In other words, in each epoch, five iterations are used for optimization of the generator parameters, while a single iteration is used for optimization of the discriminator parameters. The stochastic gradient descent (SGD) solver is adopted with the reduced learning rate option to avoid numerical oscillation problems during training. We implemented our models using the open-source deep learning package, PyTorch (https://pytorch.org/).

2.3. Performance Metrics

Two metrics are used to quantify the performance of the cDC-GAN. The structural similarity index (SSIM) commonly used in image analysis (Wang et al., 2004) provides a measure of "perceptual difference" between two images. For two sliding windows $\mathbf{u}$ and $\mathbf{v}$, the SSIM is defined by

$$SSIM(\mathbf{u},\mathbf{v}) = \frac{2\mu_{\mathbf{u}}\mu_{\mathbf{v}} + C_1}{\mu_{\mathbf{u}}^2 + \mu_{\mathbf{v}}^2 + C_1} \cdot \frac{2\sigma_{\mathbf{uv}} + C_2}{\sigma_{\mathbf{u}}^2 + \sigma_{\mathbf{v}}^2 + C_2}, \tag{13}$$

where $\mu_{\mathbf{u}}$ and $\mu_{\mathbf{v}}$ are the mean values of windows $\mathbf{u}$ and $\mathbf{v}$, $\sigma_{\mathbf{u}}$ and $\sigma_{\mathbf{v}}$ are the standard deviation of windows $\mathbf{u}$ and $\mathbf{v}$, and $\sigma_{\mathbf{uv}}$ is the covariance of windows $\mathbf{u}$ and $\mathbf{v}$. $C_1 = 0.01$ and $C_2 = 0.03$ are constants. In this study, the sizes of sliding windows $\mathbf{u}$ and $\mathbf{v}$ are both set to 11×11 pixels (grid cells).

Root-square-mean error (RMSE) is another commonly used metric and is defined by

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N} \|\mathbf{x}^i - \hat{\mathbf{x}}^i\|_2^2}, \tag{14}$$

where N is the number of samples and $\mathbf{x}^i$ and $\hat{\mathbf{x}}$ are the true and GAN-generated images, respectively.

2.4. Multiphase Flow Governing Equations

In this work, cDC-GAN is used to develop surrogate models for CO_2 flow and transport in heterogeneous formations. For the geological carbon sequestration setting considered in this study, we assume that CO_2 is injected into a brine aquifer. We also assume, without loss of generality, that the fluid flow system consists of two components (water and CO_2 gas) and two phases (liquid w and gas g). Our starting point is Darcy's law for multiphase flow, which prescribes the relationship among fluid flux, reservoir properties, fluid properties, and phase pressure in a multiphase system:

$$\mathbf{q}_\alpha = -\frac{k_{r,\alpha}\mathbf{k}}{\mu_\alpha}\left(\nabla P_\alpha - \rho_\alpha g \nabla z\right), \quad \alpha = w, g \tag{15}$$

where the subscript α denotes phase, $\mathbf{q}_\alpha$ is the phase flux, $\mathbf{k}$ is absolute permeability, $k_{r,\alpha}$ is relative permeability, ρ_α and μ_α are the fluid density and viscosity, P_α is fluid pressure in phase α, g is the gravitational constant, and z is the reservoir depth. The corresponding flow equations for the two-phase system are given by the following PDEs:

$$\frac{\partial(\phi S_\alpha)}{\partial t} = \nabla \cdot \left(\frac{\mathbf{k}k_{r\alpha}}{\mu_\alpha}(\nabla P_\alpha - \rho_\alpha g z)\right) + q_{f,\alpha}, \quad \alpha = w, g \tag{16}$$

where ϕ is porosity, S_α is saturation, $S_g + S_w = 1$, $q_{f,\alpha}$ denotes the source/sink terms in each phase, and the fluid pressures for the two-phase system are related through the capillary pressure $P_{c,w}$

$$P_g = P_w + P_{c,w}, \tag{17}$$

We assume that CO_2 from the gas phase can dissolve in the water phase, but dissolution of water in the gas phase is neglected (i.e., the gas phase contains only one component). Mass transport for component κ is governed by the following advection-dispersion equation:

$$\frac{\partial}{\partial t}\left[\phi \sum_\alpha \left(S_\alpha \rho_\alpha X_\alpha^\kappa\right)\right] + \sum_\alpha \nabla \cdot \left(\rho_\alpha X_\alpha^\kappa \mathbf{q}_\alpha\right) - \sum_\alpha \nabla \cdot \left(\phi S_\alpha \tau_\alpha D_\alpha \nabla(\rho_\alpha X_\alpha^\kappa)\right) = f^\kappa, \tag{18}$$

where X_α^κ is the mass fraction of component κ in phase α, D_α is the diffusion coefficient, τ_α is tortuosity, and f^κ denotes the sink/source term for component κ. We used the commercial compositional reservoir simulator CMG-GEM (https://www.cmgl.ca/gem) to solve the miscible flow problem described herein.

Table 1
Parameters Used in the 2-D Reservoir Model

Parameter	Value	Parameter	Value
$X \times Y \times Z$	1,280 m × 1,280 m × 20 m	Reference pressure	11 MPa
$\delta x \times \delta y \times \delta z$	10 m × 10 m × 20 m	Reservoir temperature	45 °C
$N_x \times N_y \times N_z$	128 × 128 × 1	CO_2 injection rate	5×10^5 m^3/day

3. Results and Discussion

3.1. Experiment Setup

We consider a 2-D hypothetical carbon storage aquifer with spatially heterogeneous reservoir properties. The model dimensions are 1,280 m × 1,280 m, with a uniform lateral grid block size of 10 m × 10 m and a layer thickness of 20 m. The aquifer is confined by overlying and underlying seals (i.e., no-flow boundary in the direction perpendicular to the aquifer). Infinite acting boundary conditions (Dirichlet) are imposed on all lateral sides of the aquifer. The initial reservoir pressure is 11 MPa, and the reservoir is at a constant temperature of 45 °C. A CO_2 injection well is located in the center of the aquifer at grid block location (64, 64), with a constant injection rate of 5×10^5 m^3/day (at standard surface condition) and is constrained by the maximum bottom-hole pressure of 3×10^4 kPa. The total simulation time is 380 days. To generate CO_2 saturation maps for surrogate model training and testing, outputs from 22 time steps are saved, first from the 15- to 180-day period in 15-day intervals and then from the 180- to 380-day period in 20-day intervals. Detailed geological parameters are also listed in Table 1.

For the base case, the permeability field is assumed to follow a log-normal distribution, with a mean value of ln(100) (i.e., a geometric mean of 100 mD) and standard deviation of 1.0. The porosity map is related to permeability k via the following relationship

$$\phi = 0.05\log_{10}(k) + 0.15. \tag{19}$$

A total of 1,000 realizations of the permeability field is generated using the sequential Gaussian simulator (sgsim) from the open-source package SGeMS (Remy et al., 2009). For the base case, a Gaussian variogram model is used, where the azimuth angle is zero (i.e., the major axis of anisotropy is parallel to the positive y direction) and the correlation lengths are 50 and 25 grid blocks in the major and minor variogram directions, respectively.

To test the learning capacity of cDC-GAN under different amount of training data, we train separate cDC-GAN models using an ensemble of 200, 400, 600, and 800 permeability realizations, respectively, as input data. For each permeability realization, simulated CO_2 saturation maps from 22 time steps are used as

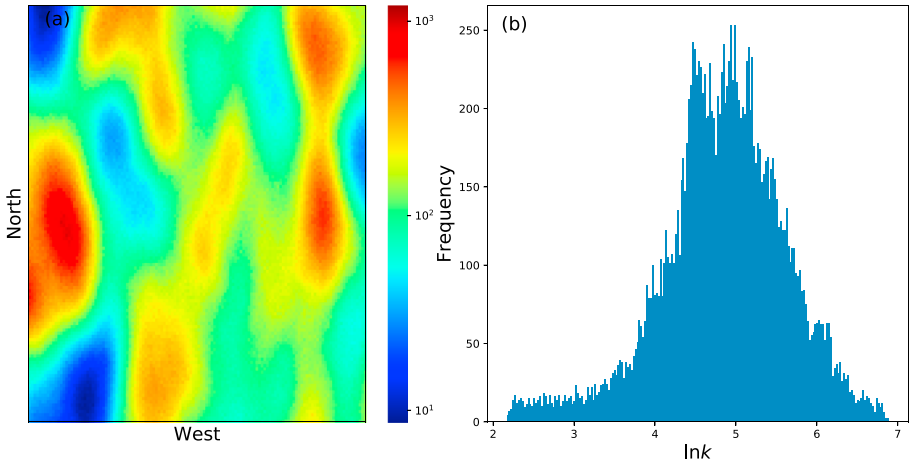

Figure 2. (a) An example permeability field realization (before log-transformation) used for training the conditional deep convolutional generative adversarial network (cDC-GAN) and (b) the corresponding histogram of the log-transformed permeability field.

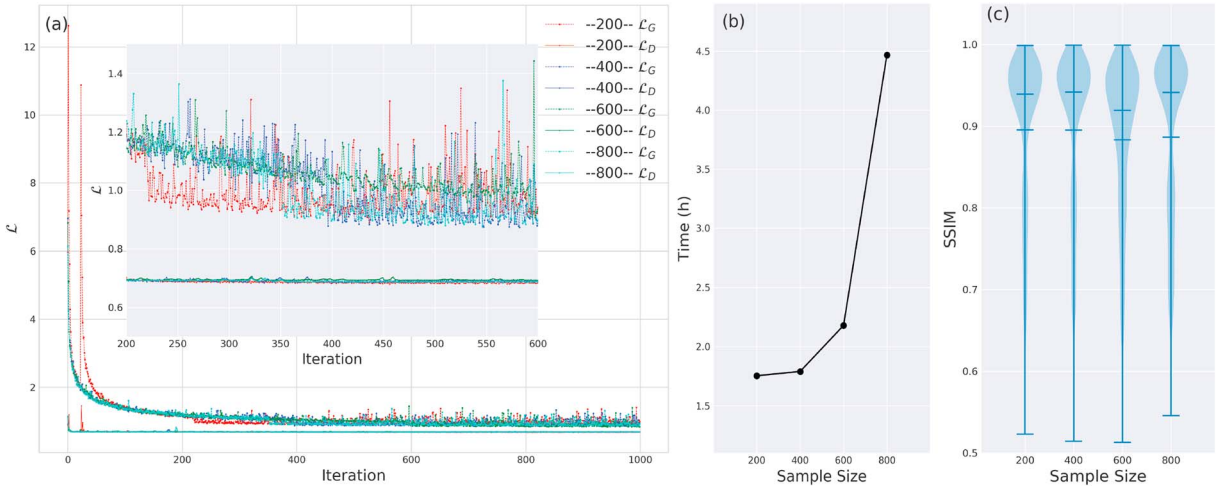

Figure 3. (a) Training loss (RMSE) decreases with the number of epoch but generally becomes stable after 400 epochs. (b) Comparison of training time taken to train cDC-GAN with a training ensemble size of 200, 400, 600, and 800 realizations, respectively. (c) Violin plot of SSIM for the testing performance of cDC-GANs trained using different training sample sizes; the short bars (from top to bottom) represent the maximum, mean, median, and minimum SSIM, respectively. RMSE = root-mean-square error; cDC-GAN = conditional deep convolutional generative adversarial network; SSIM = structural similarity index.

training targets, and the corresponding time step information is used as conditioning data (see also discussions under section 2.2). The log-transformed, input permeability field ($\ln k$) is normalized to the interval [0, 1] before training. Figure 2 shows a single realization of the permeability field and the corresponding $\ln k$ histogram. Note the histogram is not exactly Gaussian in this case because of the large correlation lengths (relative to the domain sizes) used. All trained models are tested by using a separate set of 200 permeability realizations not included in training to evaluate their performance.

3.2. Performance Evaluation

3.2.1. Training and Testing Performance

Figure 3 shows the value of cDC-GAN training loss as a function of epochs and for training ensemble sizes of 200, 400, 600, and 800, realizations. All of the cDC-GAN models are trained on a cluster node equipped with NVIDIA GeForce GTX 1080 Ti GPU for a total of 4,000 epochs. Here an epoch is defined as a single pass of the entire training set to the solver. As Figure 3a shows, the RMSE values of the generative models and discriminative models start to stabilize after 400 epochs in all cases. The value of the generative model loss converges to 0.9, and the value of the discriminative model loss converges to 0.7. Note also the convergence of the discriminative model is much faster than that of the generative model because of its simpler model structure. The inset of Figure 3a shows an amplified view of the loss function from epochs 200 to 600.

Figure 3b shows training time for different cDC-GAN models. In general, a nonlinear relationship exists between the size of training data and training time (computing cost). The training time increases from 1.7 to 4.5 hr, as the size of training set increases from 200 to 800 realizations. The cDC-GAN model trained on 200 $\ln k$ realizations achieved a mean SSIM value of 0.957 on the testing set (Figure 3c). For the same testing set, the mean SSIM value increases slightly when the training sample size is increased from 200 to 800 realizations. The highest mean SSIM value of 0.988 is achieved by the cDC-GAN model trained on 800 realizations (Figure 3c). With the increase of the training samples size, the prediction accuracy also increases, but at the cost of an almost exponential increase in training time. For the rest of this section, we choose the cDC-GAN model trained on 600 realizations as the base model, which strikes a reasonable balance between the computational time and prediction accuracy for our study. Unless otherwise specified, results reported below pertain to this base model.

We now use one realization from the testing set to exemplify how cDC-GAN works. Figure 4 illustrates the temporal evolution of CO_2 saturation maps simulated by CMG-GEM at all 22 output times for the reference realization being used as the example. In Figure 4, the second to last subplot shows the normalized reference log-permeability field, and the last subplot on the last row shows the histogram of the normalized $\ln k$ field. Results show that the CO_2 plume grows with time, but the plume shape is not symmetric because of the heterogeneity of the reservoir. The highest CO_2 saturation is found at the center of the field because of the

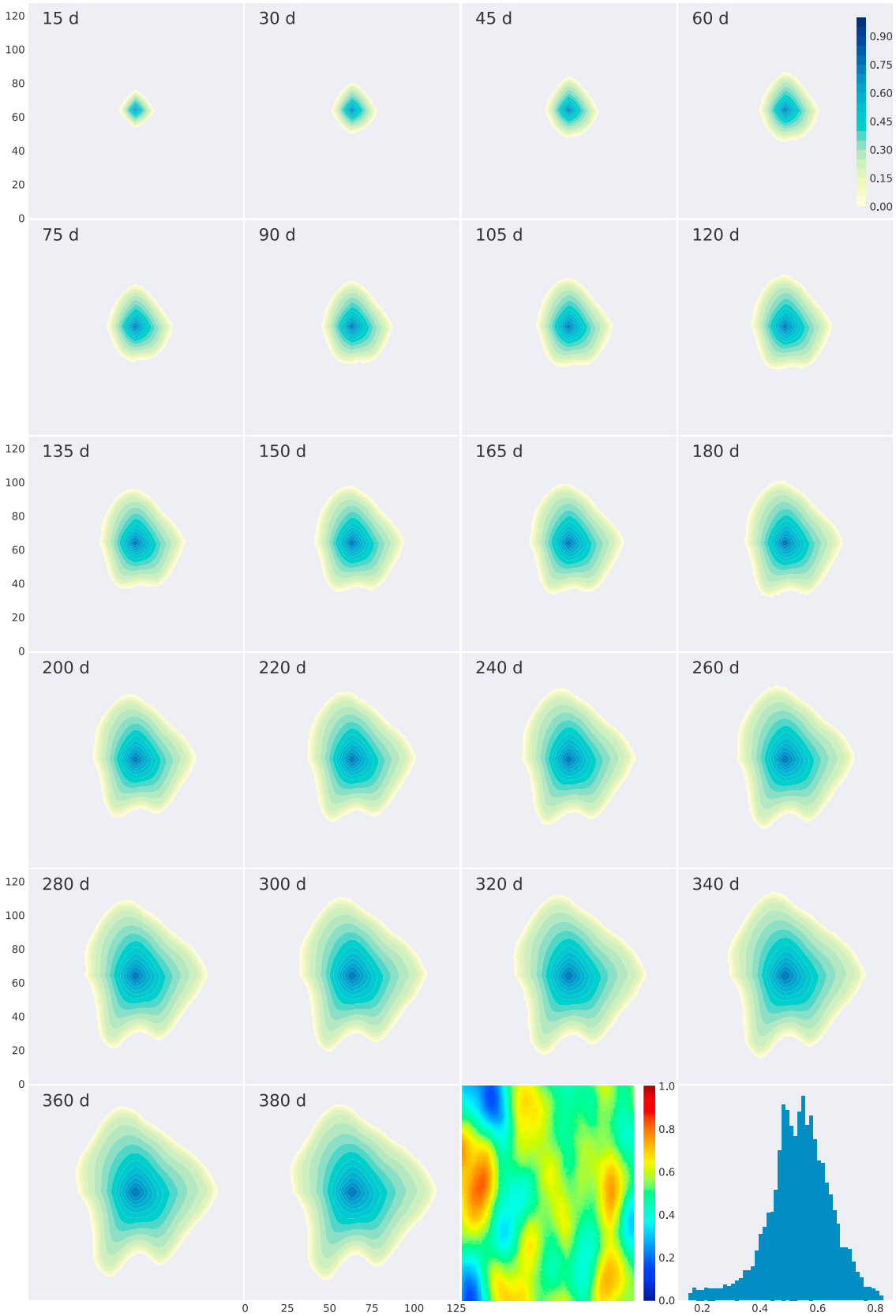

Figure 4. CO_2 saturation maps at 22 different time steps as simulated by CMG-GEM for a reference permeability field (normalized $\ln k$), which is shown in the second to last subplot in the last row. The last subplot shows the corresponding histogram of the reference permeability field.

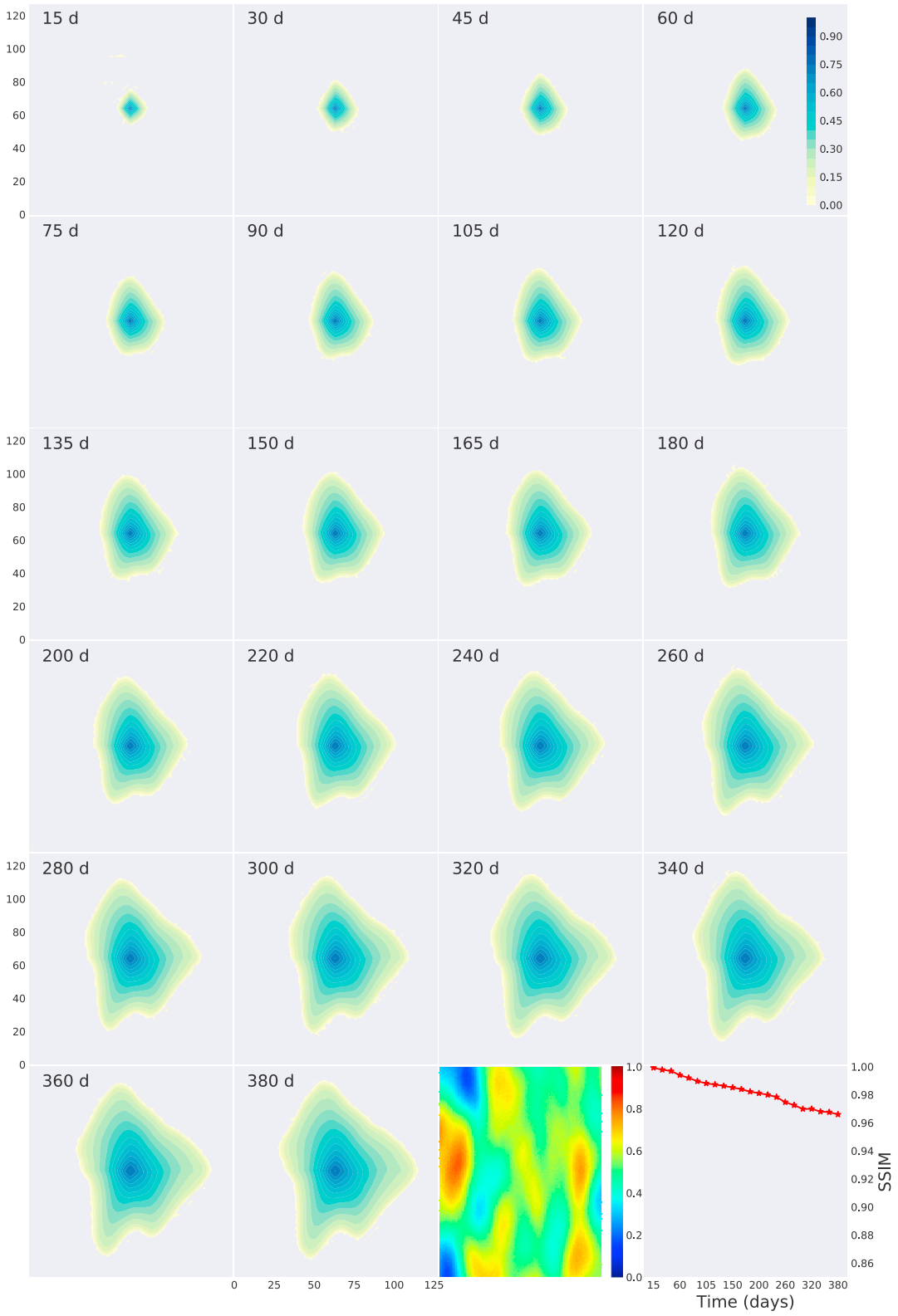

Figure 5. Snapshots of CO_2 saturation maps at different time steps obtained by using the conditional deep convolutional generative adversarial network (cDC-GAN) surrogate model trained using 600 training realizations. The second to the last subplot on the last row shows the permeability field (normalized ln k) used to train the cDC-GAN, and the last subplot on the last row shows the structural similarity index (SSIM) between the CO_2 saturation map generated by cDC-GAN and that simulated by CMG-GEM.

constant-rate CO_2 injection there. The CO_2 plume tends to migrate along the higher permeability direction, leading to elongated plume shapes along the south-north direction in this case.

Figure 5 shows the CO_2 plumes generated by the trained cDC-GAN model (i.e., the surrogate model) for the same reference permeability field as shown in Figure 2. In general, the surrogate model captures the main features of the simulated CO_2 plume over time, also showing an elongated pattern in the south-north direction. The last subplot in the last row of Figure 5 shows the SSIM value calculated between the simulated and cDC-GAN-generated CO_2 saturation maps, which decreases from a value close to 1.0 to 0.96 with time. At larger transport times, the CO_2 plume "experiences" more heterogeneity, and its shape becomes more difficult to predict because of the larger degree of freedom. Another reason is related to the design of SSIM metric, which is related to the mean and variance of the sliding windows (equation (13)). As Figures 4 and 5 show, most of the pixel values located outside the plume are zero at early times (e.g., at 15 days), the resulting SSIM value is high because of the "dilution" effect of zero-valued cells. At later times, the number of zero-valued pixels decreases as the plume size increases. Overall, the cDC-GAN does a relatively good job in approximating the shape and value of CO_2 plumes. The example also suggests that the use of time step as the conditioning information is instrumental for helping the cDC-GAN to learn the cross mappings between the static permeability field and dynamic model outputs.

To further investigate the difference between saturation maps simulated by CMG-GEM and that generated by the cDC-GAN, Figure 6 plots the residual maps between the two sets of results at selected time steps. As Figure 6 suggests, the cDC-GAN model achieves a high accuracy for predicting the CO_2 plume, with most of the error residuals close to 0.

Figure 7 plots the SSIM value for the entire testing ensemble consisting of 200 realizations. In general, we observe similar temporal patterns as we have seen in the example case—as time progresses, the SSIM values tend to decrease. The ensemble of SSIM curves tends to have a larger spread with time, as shown by the different SSIM histograms in Figures 7b1–7b6 corresponding to time steps 30, 75, 135, 200, 280, and 380 days.

In addition to SSIM, we also performed the statistical moment analysis on the testing ensemble. Statistical moment analysis is typically used to measure the quality of surrogate modeling against Monte Carlo simulations conducted using the actual models. In Figure 8, the ensemble mean and standard variation of CO_2 saturation maps generated by cDC-GAN are compared to those simulated using CMG-GEM for the testing ensemble. The first row of Figure 8 shows the mean (μ) and standard variation (σ) of CO_2 saturation simulated by CMG-GEM at time steps of 105 and 340 days, respectively. The second row of Figure 8 shows the results of statistical moment analysis obtained for a horizontal cross section a–a′ at time steps of 105 and 340 days, and the third row shows the results obtained for a south-north cross section b–b′ for the same output times. The ensemble mean CO_2 saturation obtained by the surrogate model (dashed line) overlays almost exactly on top of that obtained by the CMG-GEM simulations (solid line with asterisks). The ensemble standard deviation shows a close match at 105 days, although slight deviations near the σ peaks can be observed in the near field of the injector at 340 days. An interesting observation from Figure 8 is that the matching quality in the far field stays relatively unaffected. This indicates that the surrogate model achieves good performance in tracking the plume front, which is especially encouraging for CCS applications because of the need to demonstrate closure (i.e., safe containment) of the injected CO_2 in CCS projects. To highlight the dynamic change of μ and σ of CO_2 saturation along injection time, μ and σ at grid block of (60, 40) and (100, 50) are plotted in Figure S1 in the supporting information. A close match between CMG-GEM simulation result and cDC-GAN prediction result at grid blocks (60, 40) and (100, 50) indicates that the cDC-GAN can predict the CO_2 plume at any injection time. From a deep learning design perspective, the use of GAN in this case plays an important role in helping to reduce blurriness in the generated images (i.e., the saturation maps in this case; Isola et al., 2017). Our results also suggest that cDC-GAN may be used as a reasonable alternative to full numerical model Monte Carlo simulations for UQ tasks.

3.2.2. Interpolation Capability

In the previous example, the cDC-GAN model was trained by using model outputs from all 22 time steps for each input permeability field in the training set (i.e., 600×22 target training samples). In practice, it is important for a surrogate model to be able to predict model outputs at arbitrary intermediate simulation time steps. To assess such interpolation ability of the cDC-GAN, here we retrain the cDC-GAN model by using only a subset of all 22 time steps. Specifically, we leave time steps (45, 105, 150, 200, 260, 320, and 360 days)

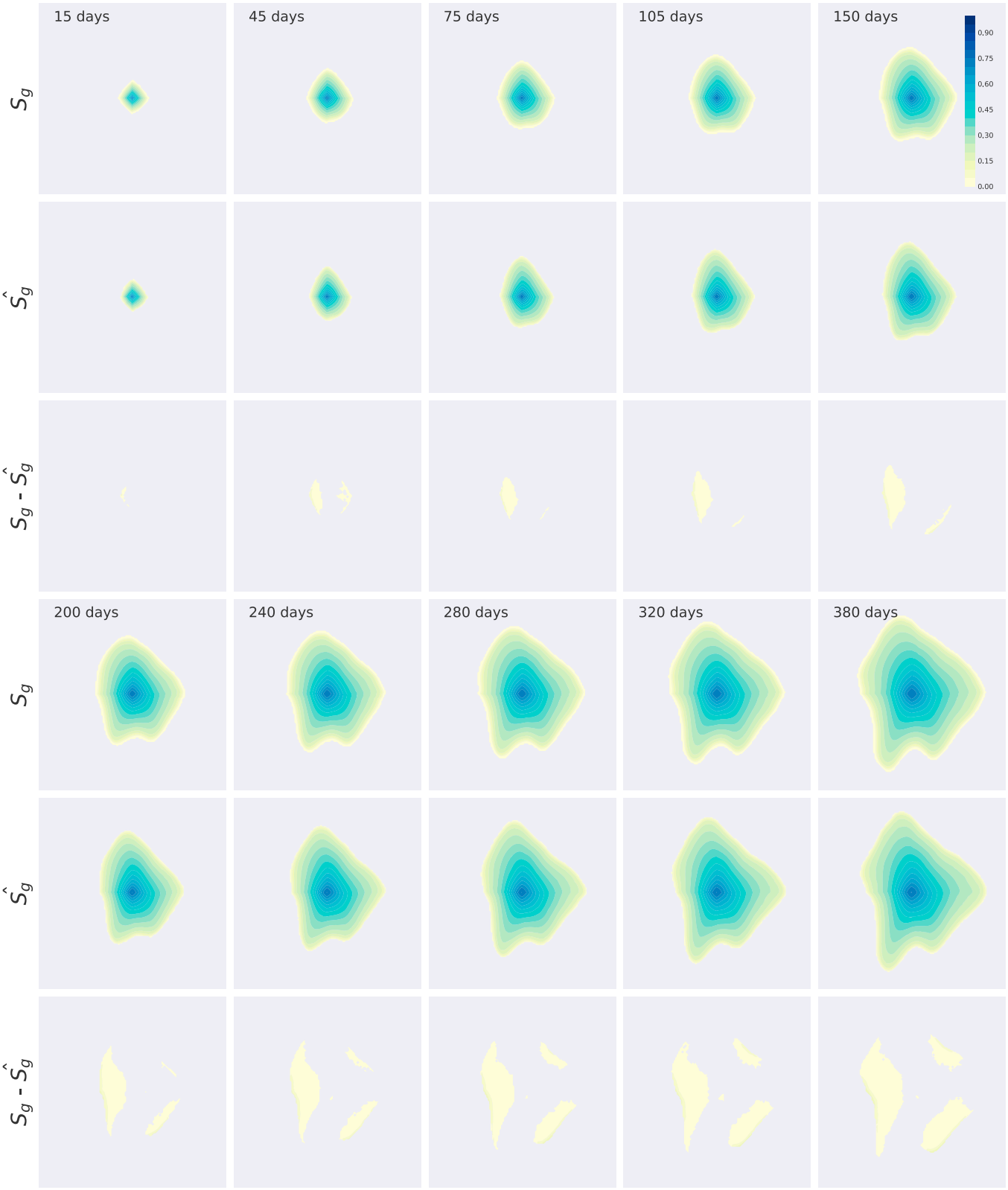

Figure 6. Residual error maps between CMG-GEM and cDC-GAN CO_2 saturation maps at selected time steps. S_g denotes the CMG-GEM results, $\hat{S}_g$ denotes the cDC-GAN results, and $S_g - \hat{S}_g$ corresponds to the difference between each pair of maps. Note the same color bar is applied to all panels.

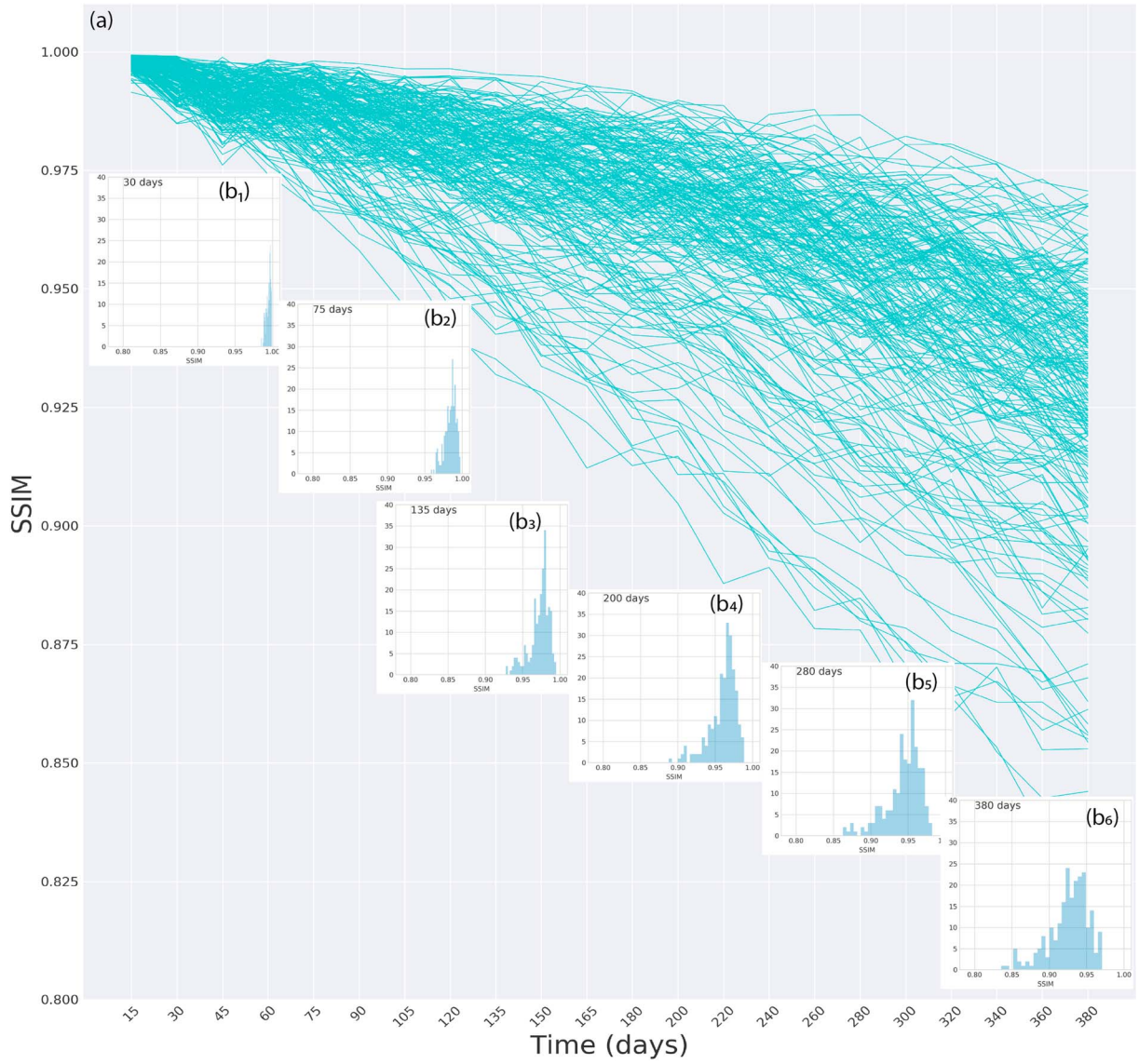

Figure 7. Plots of structural similarity index (SSIM) values obtained on the testing ensemble of 200 realizations. Insets show histograms of SSIM at 30, 75, 135, 200, 280, and 380 days.

out during training. After the cDC-GAN is trained, we test the trained model on predicting the CO_2 plume at 45, 105, 150, 200, 260, 320, and 360 days (the simulation results are already archived) to evaluate its interpolation ability. Figure 9 shows the comparison between the predicted CO_2 saturation field (S_g) and simulated CO_2 saturation field ($\hat{S}_g$) for time steps used for this interpolation test. Results show that the cDC-GAN does a satisfactory job in approximating CO_2 saturation fields at different "new" time steps not used during training, suggesting its strong interpolation skill and the consistency of the cDC-GAN performance. Thus, the cDC-GAN model can be used as an efficient surrogate model for dynamic systems.

3.2.3. Model Comparison

Here the performance of our cDC-GAN surrogate model (for the base case) is evaluated against a naïve surrogate model, the mean-plume predictor, that is obtained by simply taking the ensemble average of CMG-GEM CO_2 saturation maps obtained for the training set at all output times. We then evaluate the performance of this mean-plume predictor on all test realizations by calculating the difference (residual) between CO_2 saturation maps of the mean-plume predictor and that obtained by the CMG-GEM runs (note that no extra model runs are needed for this comparison because all forward simulations have already been completed for the training and testing sets).

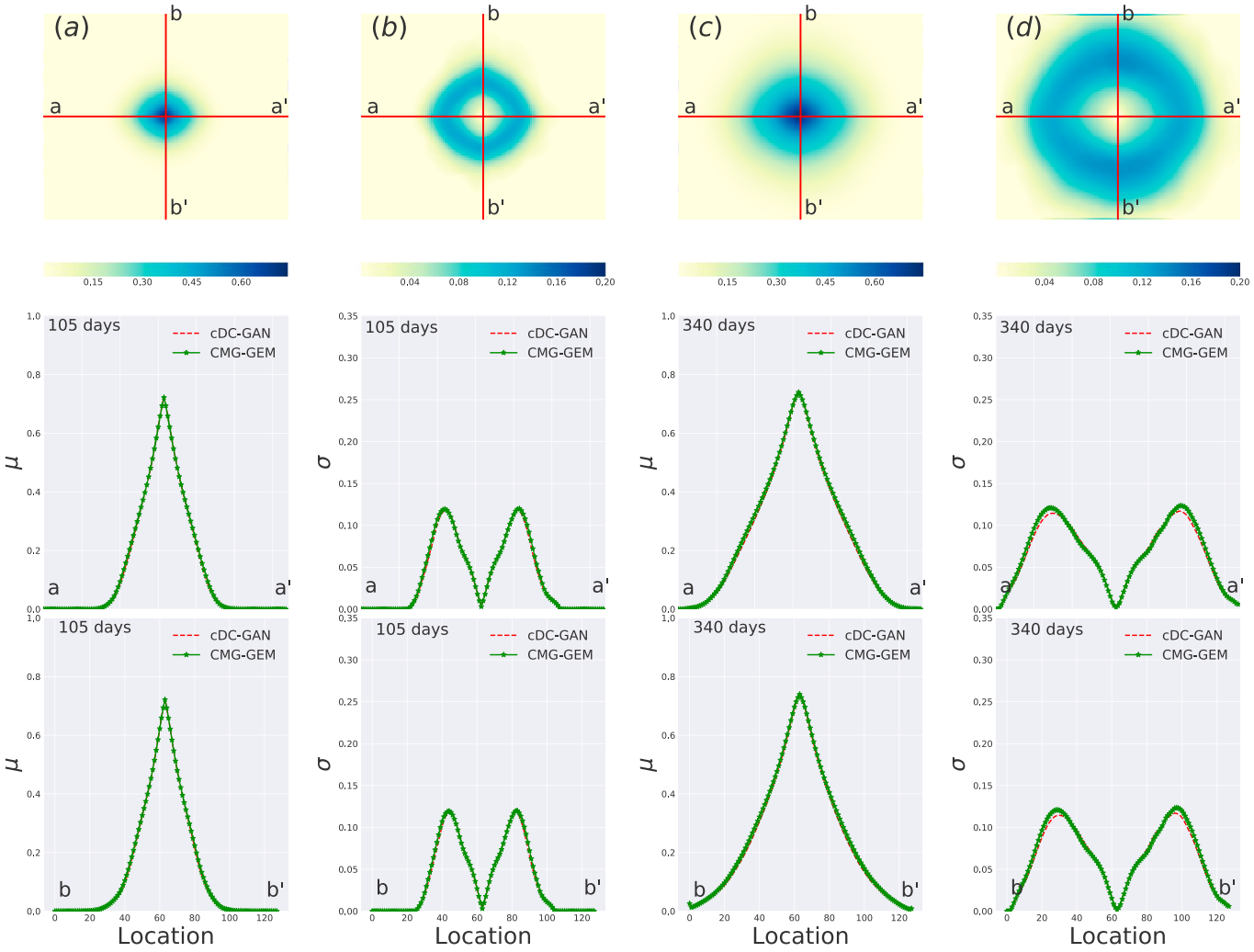

Figure 8. Statistical moment analysis of CO_2 saturation maps (S_g) obtained by surrogate model and numerical model on the testing ensemble of 200 realizations. Top row: (a) mean CO_2 saturation simulated by CMG-GEM at 105 days, (b) standard deviation of CO_2 saturation at 105 days, (c) mean CO_2 saturation at 340 days, and (d) standard deviation of CO_2 at 340 days. Middle row: from left to right, ensemble mean and standard deviation obtained at 105 and 340 days using surrogate model (dashed line) and CMG-GEM simulations (solid line with asterisks) at the cross section a–a′ in a, b, c, and d. Bottom row: from left to right, ensemble mean and standard deviation obtained at 105 and 340 days at the cross section b–b′ in a, b, c, and d.

The top panel in Figure S3 shows the CO_2 saturation maps of the mean-plume predictor for time steps 40, 105, 150, 200, 260, and 340 days. The bottom panel in Figure S3 shows the residual maps (ΔS_g) obtained by applying the mean-plume predictor on all test realizations. As the color bar in Figure S3 (bottom panel) suggests, the residual is the greatest (~0.2) near the injector and then gradually decreases toward the edges. For comparison, Figure S4 shows the same predictions obtained by using the trained cDC-GAN on the testing set. The bottom panel of Figure S4 suggests that the residual ΔS_g obtained by cDC-GAN is almost zero near the injector and only becomes visible around the plume edges but is generally less than 0.03. This comparison highlights the need for developing an accurate surrogate model, especially when model parameters are spatially heterogeneous.

3.2.4. Computational Cost

In Figure S2, we compare the total computational time of using CMG-GEM to that of using the cDC-GAN surrogate model for various number of realizations. In this case, each numerical simulation takes about 10 min to finish by running CMG-GEM. As Figure S2 shows, CMG-GEM and cDC-GAN take the same amount of time for the first 600 simulations, because the training data of cDC-GAN model come from running the CMG-GEM simulations. Then, training of the cDC-GAN takes about 2.5 hr, which is equivalent to another 15 CMG-GEM runs (assuming 10 min for each run). After training, the cDC-GAN proxy model

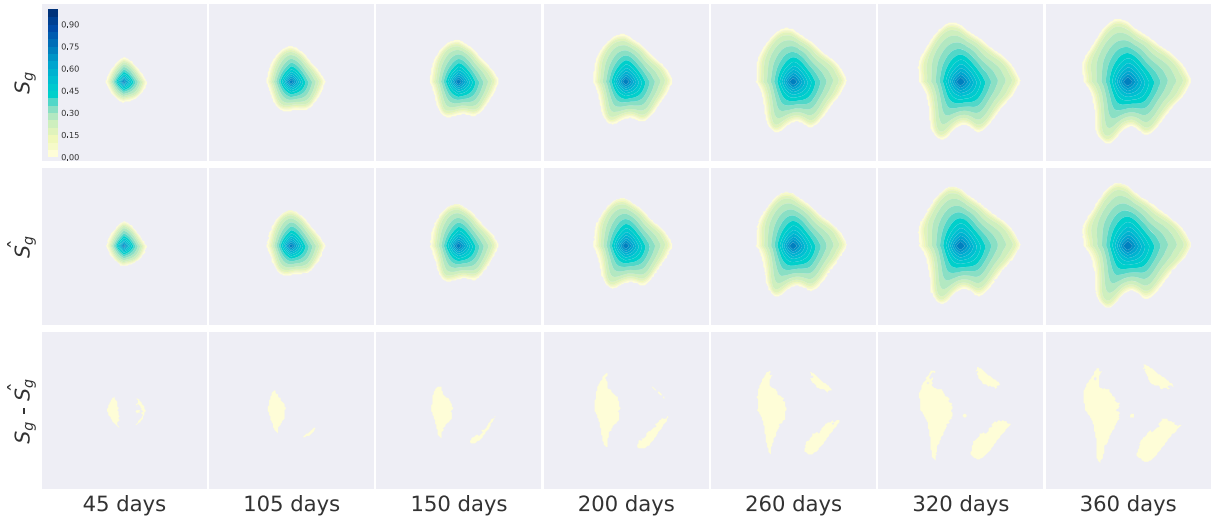

Figure 9. Test of conditional deep convolutional generative adversarial network (cDC-GAN) interpolation ability at time steps not included in training (45, 105, 150, 200, 260, 320, and 360 days). (Top row) CO_2 saturation field simulated by CMG-GEM (S_g), (middle row) CO_2 saturation field ($\hat{S}_g$) generated by cDC-GAN, and (bottom row) differences between the top and middle rows. The cDC-GAN model is trained using 600 permeability realizations.

can get CO_2 saturation at any time instance and for any permeability distribution with a few seconds, but CMG-GEM still takes 10 min to run a single simulation. Thus, if the number of required Monte Carlo simulations is more than 615, which is often the case for subsurface UQ (Li, 2014), using cDC-GAN as a surrogate model will be more efficient computationally.

3.3. Sensitivity Studies

3.3.1. Effect of ln *k* Correlation Range

In the base case, the correlation ranges used for generating the ln k realizations are relatively large as compared to the domain dimensions. As a result, the generated fields exhibit smooth large-scale spatial patterns which, in turn, may have helped the pattern extraction and learning by cDC-GAN. A remaining question is whether the performance of cDC-GAN will be affected when the spatial heterogeneity is increased. To address this concern, we ran additional sensitivity studies using smaller correlation ranges.

In the first test, the correlation ranges were changed to 25 (in major direction) and 15 (in minor direction) grid blocks, while all other configurations were kept the same (Table 2). Figure S5 shows an example of the generated permeability field and its histogram, which shows smaller-scale spatial heterogeneity and thus higher spatial variations. We trained a cDC-GAN model for this case using 600 realizations. Figure S6 shows the comparison between CO_2 plumes simulated by CMG-GEM and those generated by cDC-GAN at selected simulation time steps, while the results of statistical moment analyses are presented in Figure 10 for the same two cross sections as we have considered previously in the base case. Results show that the performance of cDC-GAN is largely unaffected when the correlation range is reduced.

In the second test, we reduced the correlation ranges of ln k further to 10 grid blocks in both major and minor directions. Figure S7 shows an example of the generated permeability field and its histogram, which exhibits even more spatial heterogeneity than the previous test. Again, a cDC-GAN was trained using 600 ln k realizations. Figure S8 shows the comparison between CO_2 plumes simulated by CMG-GEM and those

Table 2
Experiment Design of Correlation Range, Standard Deviation, and Injection Period for Sensitivity Analyses

Case no.	Major	Minor	Mean (μ)	Std (σ)	Injection period (days)
Base case	50	25	ln(100)	1	380
Shorter correlation range (1)	25	15	ln(100)	1	380
Larger standard deviation	50	25	ln(100)	2	380
Shorter correlation range (2)	10	10	ln(100)	1	380
Longer injection period	50	25	ln(100)	1	540

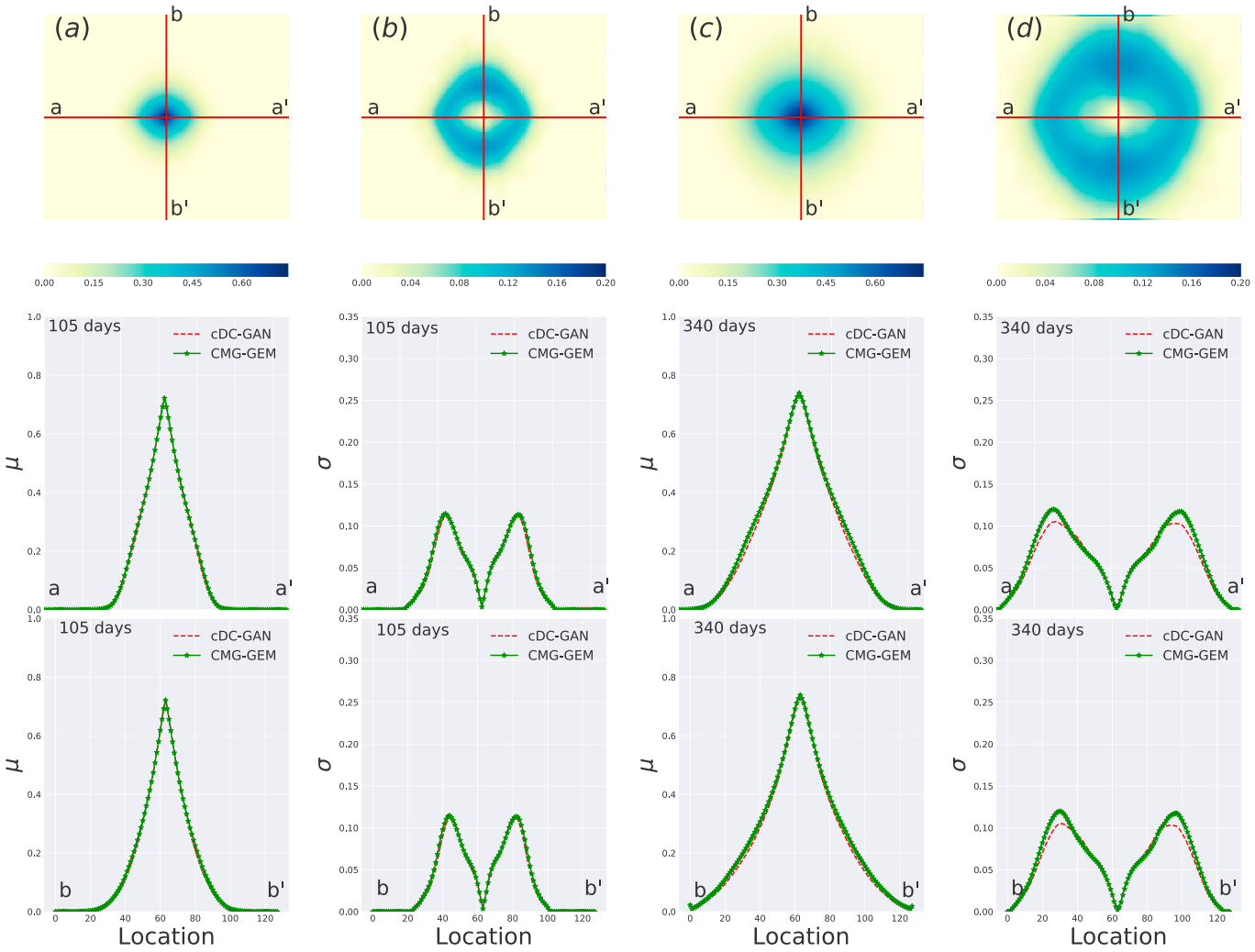

Figure 10. Comparison of statistical moments of CO_2 saturation fields obtained by conditional deep convolutional generative adversarial network (cDC-GAN) and CMG-GEM for the testing ensemble of 200 realizations at 105 and 340 days, for the correlation range sensitivity analysis. (top row) (a) Mean CO_2 saturation simulated by CMG-GEM at 105 days, (b) standard deviation of CO_2 saturation at 105 days, (c) mean CO_2 saturation at 340 days, and (d) standard deviation of CO_2 saturation at 340 days. (middle row) From left to right, ensemble mean and standard deviation obtained at 105 and 340 days using surrogate model (dashed line) and CMG-GEM simulations (solid line with asterisk) at the cross section a–a′ in a, b, c, and d. (bottom row) From left to right, ensemble mean and standard deviation obtained at 105 and 340 days at the cross section b–b′ in a, b, c, and d.

generated by cDC-GAN at selected simulation time steps. Figure S9 displays the results of statistical moment analysis. The mean value between cDC-GAN and CMG-GEM at both 105 and 340 days is matched very well in both the base case (Figure 8) and the smaller correlation range studies (Figures 10 and S9). However, the match in standard deviation between cDC-GAN and CMG at 105 days is better than that at 340 days in both the base case (Figure 8) and sensitivity cases (Figures 10 and S9). By comparing the standard deviation between cDC-GAN and CMG-GEM at 340 days for all three cases (Figures 8, 10, and S9), it can be observed that the match in saturation map standard deviation only decreases slightly when the correlation range is decreased.

Previously, it was shown, through eigenvalue analysis, that the number of eigenvalues required to represent a log-normal permeability field at the same energy level (i.e., in KLE sense) increases with the decrease in correlation length (Li, 2014; Zhang & Lu, 2004). Thus, smaller correlation lengths tend to be more challenging for stochastic surrogate modeling techniques that are based on KLE. In the case of cDC-GAN, small correlation lengths may imply that the dimensionality of the underlying latent space (or intrinsic dimensionality) is higher. The series of sensitivity studies presented herein suggest that the performance of cDC-GAN is little affected when the correlation length is reduced, indicating that the end-to-end learning behind cDC-GAN

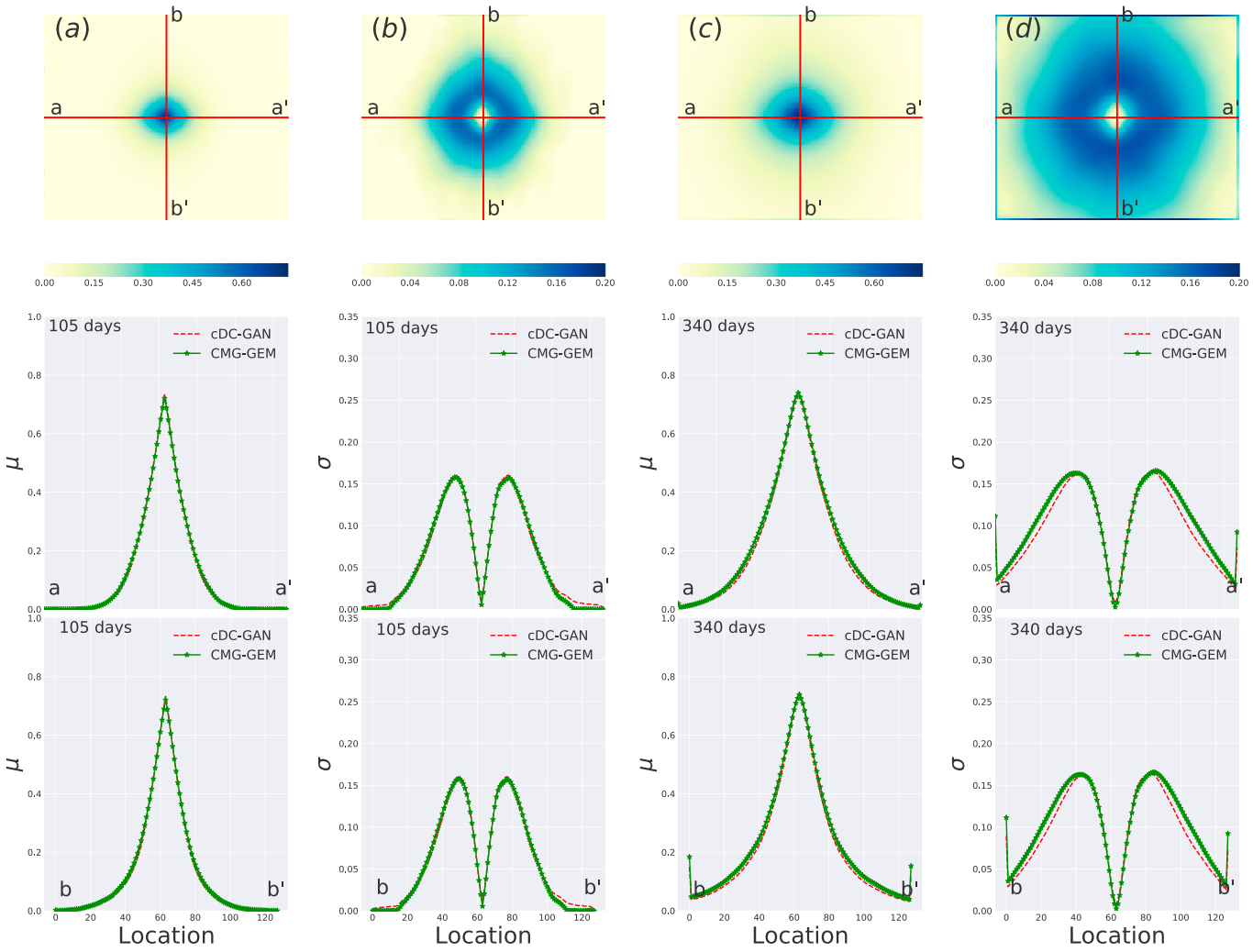

Figure 11. Comparison of statistical moments of CO_2 saturation fields obtained by cDC-GAN and CMG-GEM for the testing ensemble with larger $\ln k$ standard deviation ($\sigma_{\ln k} = 2$) at time steps 105 and 340 days. (top row) From left to right, mean and standard deviation of CO_2 saturation simulated by CMG-GEM at 105 and 340 days, respectively. (middle row) Results for a horizontal cross section a–a′. (bottom row) Results for south-north cross section b–b′.

can be trained to accommodate the increased latent space dimension while maintaining the same training effort (i.e., 600 realizations in our study).

3.3.2. Effect of Spatial Variability

This sensitivity analysis investigates the effect of higher spatial variability on the performance of cDC-GAN. Specifically, we increase the standard deviation of $\ln k$ from 1.0 to 2.0 while keeping all other parameters the same. Figure S10a illustrates an example of permeability realization generated based on Gaussian semi-variogram with the larger standard deviation (Table 2), and Figure S8b displays the corresponding histogram of permeability map. A total of 800 simulations is simulated by CMG-GEM, of which 600 simulations are used for training the cDC-GAN, and the remaining 200 simulations are used for testing. Figure S11 shows the resulting CO_2 saturation distributions that are simulated by CMG-GEM and by cDC-GAN at different times for the higher standard deviation case. To quantify the performance of cDC-GAN, we calculate the mean and standard deviation on the 200 testing realizations at the time instance of 105 and 340 days (Figure 11). The result indicates that the magnitude of standard deviation has little impact on the performance of cDC-GAN. The magnitude of mean CO_2 saturation at 105 and 340 days is the same in both the base case (Figure 8) and the larger standard deviation case (Figure 11); however, because of the increased spatial heterogeneity, the magnitude of standard deviation of CO_2 saturation at 105 and 340 days increases compared with the base case. Again, the robust cDC-GAN performance seen here may be attributed to the strong capacity of cDC-GAN to learn the underlying functional mapping through pattern extraction, not

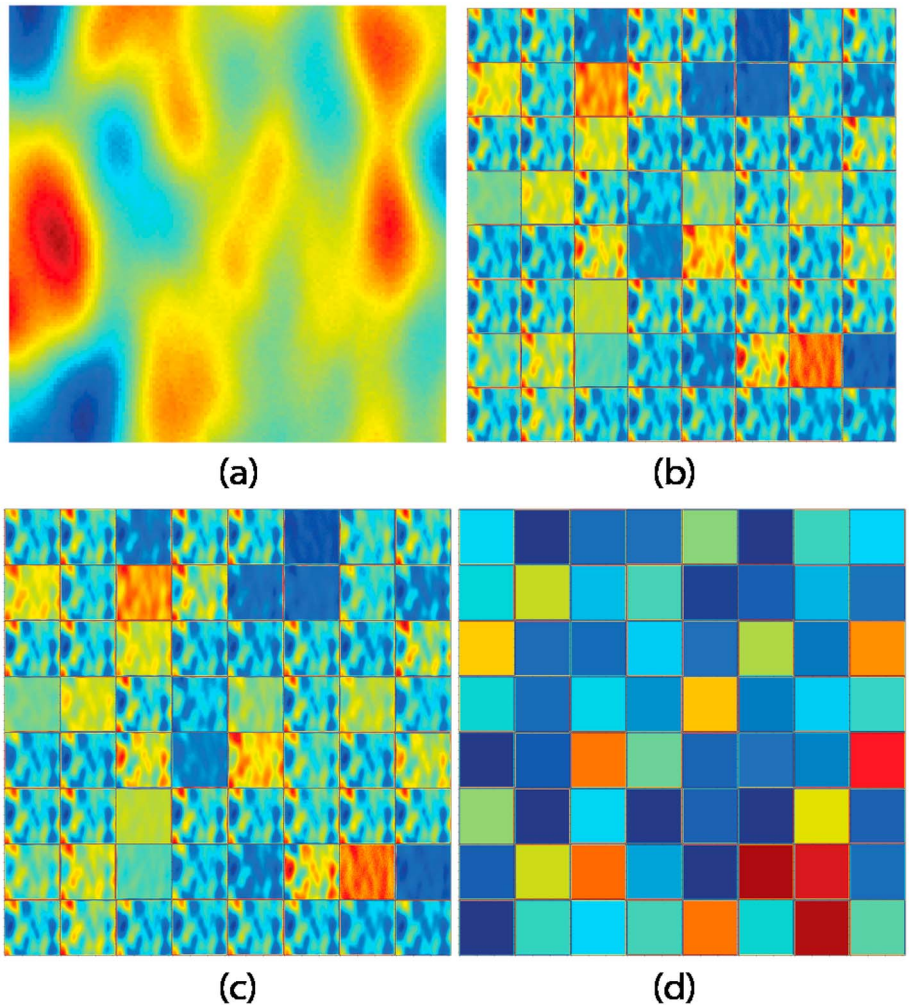

Figure 12. Illustration of conditional deep convolutional generative adversarial network feature extraction. (a) A sample permeability realization used as input, (b) feature maps outputted from the first layer of convolution calculation with $t = 15$ days as conditioning data, (c) feature maps after the first layer convolution calculation with $t = 380$ days as conditioning data, and (d) residual map between feature maps calculated in (b) and (c). The last three subplots suggest that permeability determines the pattern while time controls the magnitude (or intensity) of the feature maps.

by approximation through polynomial expansions that invariably lose information after truncation (Xiu & Karniadakis, 2002).

3.3.3. Effect of Injection Duration

For the current problem, the duration of injection time affects the size of CO_2 plumes and thus the SSIM metric (see also the discussion under section 3.2.1). To investigate the impact of different injection time lengths on the performance of cDC-GAN, we increase the total injection period from 380 to 540 days while keeping all other parameters the same as in the base case. The same sensitivity analysis is repeated. Figure S12 shows the SSIM values, which are obtained on 200 testing realizations. The max SSIM value is around 0.99, which is obtained on the first time instance. The minimum SSIM value is 0.825, which is obtained for the end of time instance. It can be seen that the SSIM value does not decrease too much as compared to the base case (Figure 7). The result suggests that although the deviation (as measured by SSIM) between the CO_2 saturation simulated by CMG-GEM and CO_2 saturation predicted by cDC-GAN increases, it is still within an acceptable range.

3.4. What Has the cDC-GAN Learned?

As mentioned before, the cDC-GAN is a pattern-driven deep learning method. To help elucidate this point, we compare feature maps obtained after the first layer of convolutional operation (i.e., the 64×64 feature maps in Figure 1). Figure 12a shows an example of input permeability map, and Figure 12b shows the

corresponding feature maps extracted from the input after the first convolutional layer and for the time step of 15 days. Recall that the time step is used as conditioning data. Comparing Figures 12a and 12b, it can be seen that some subplots in Figure 12b exhibit the same spatial patterns as shown in the original permeability map (Figure 12a). Figure 12c shows the feature maps calculated after the first convolutional layer at 380 days. In Figure 12d, the difference between Figure 12b and Figure 12c is shown. Each subplot in Figure 12d turns out to show a constant value image, which means that the patterns extracted by the first layer do not change between Figures 12b (15 days) and 12c (380 days), and the only change between those two figures are magnitudes, as reflected in the color intensity of the subplots in Figure 12d. These results imply that the conditioning data (time step) mainly affects the color intensity of the features extracted from the first layer, while the image patterns are determined by the spatial patterns in the permeability map. The results shown here may be compared to the KLE, which also uses a combination of eigenmaps to represent a stochastic process (Zhang & Lu, 2004). The difference is that the end user does not need to control the order of expansion, nor worry about the distribution of the input. In the supporting information (Figures S13–S15), feature maps extracted after the second convolutional operation are also plotted, which show that more detailed information is extracted from the permeability (Figure 12a) for different time steps.

4. Conclusions

The GANs, a type of deep learning models, have shown promising performance in learning cross-domain mappings. In this work, we adopted a GAN framework to develop surrogate models of high-dimensional dynamical numerical models, which has been extensively studied but remains a challenging task under the conventional surrogate modeling frameworks. In particular, we have developed a cDC-GAN for stochastic surrogate modeling. The developed cDC-GAN model is demonstrated for a carbon capture and storage (CCS) use case, for which we seek to predict the spatial and temporal evolution of injected CO_2 plume in heterogeneous carbon storage reservoirs. The underlying multiphase flow and transport problem is highly nonlinear and normally solved via compositional reservoir simulation, which is computationally expensive even on high-performance clusters. The cDC-GAN is trained to learn the functional mappings between two domains: the model input domain is permeability field and the output domain is the CO_2 plume. The conditioning data is the model output time. Our results indicate that cDC-GAN has strong skills in learning the cross mappings between the permeability fields and simulated CO_2 plumes. By feeding it with different stochastic realizations of the permeability, the cDC-GAN is trained to learn a generalized mapping that can handle different cases from the same data distribution class. The performance of the cDC-GAN stays relatively robust when the heterogeneity structure changes (e.g., smaller $\ln k$ correlation ranges or larger $\ln k$ standard deviation), without requiring the increase of training samples at the same time. This suggests the strong capacity of cDC-GAN to adjust to the change in latent space dimensionality. It also interpolates well for time steps not used in training, which is an important attribute to have for high-quality surrogate models. As part of the study, we also investigate the feature maps extracted by cDC-GAN and show that the feature maps can be interpreted meaningfully. It is worth pointing out that unlike many surrogate modeling techniques that either assume a linear system or require parametric probability distribution functions, cDC-GAN imposes few assumptions on the input data and is entirely data (or pattern) driven. Thus, it can be potentially applied to a large class of physical simulation problems for developing surrogate models for risk assessment and UQ purposes.

Acknowledgments
We are grateful to the AE and three anonymous reviewers for their constructive comments. Z. Zhong and A. Y. Sun were supported by the U.S. Department of Energy, National Energy Technology Laboratory (NETL) under Grant DE-FE0026515. H. Jeong was partially supported by the National Research Foundation of Korea (NRF) under Grant 2018R1C1B5045260. Computing resources are provided by the Texas Advanced Computing Center at UT Austin. We are grateful to the Computer Modeling Group (Calgary, Canada) for free access to their CMG-GEM software. Python codes of the proposed cDC-GAN are available at the GitHub (https://github.com/danilecug/CO2-GAN/).

References

Agarwal, K., Sharma, P., Ma, J., Lo, C., Gorton, I., & Liu, Y. (2014). Reveal: An extensible reduced-order model builder for simulation and modeling. *Computing in Science & Engineering*, *16*(2), 44–53.

Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein GAN. arXiv preprint arXiv 170107875.

Baddar, W. J., Gu, G., Lee, S., & Ro, Y. M. (2017). Dynamics transfer GAN: Generating video by transferring arbitrary temporal dynamics from a source video to a single target image. arXiv preprint arXiv 171203534.

Chan, S., & Elsheikh, A. H. (2017). Parametrization and generation of geological models with generative adversarial networks. arXiv preprint arXiv 170801810.

Costa, A., & Nannicini, G. (2018). Rbfopt: An open-source library for black-box optimization with costly function evaluations. *Mathematical Programming Computation*, *10*(4), 597–629.

Cotter, S. L., Roberts, G. O., Stuart, A. M., & White, D. (2013). MCMC methods for functions: Modifying old algorithms to make them faster. *Statistical Science*, *28*, 424–446.

Dagan, G., & Neuman, S. P. (2005). *Subsurface flow and transport: A stochastic approach*. Cambridge, England: Cambridge University Press.

Dai, Z., et al. (2018). Heterogeneity-assisted carbon dioxide storage in marine sediments. *Applied Energy*, *225*, 876–883.

De Silva, G., Ranjith, P., Perera, M., & Chen, B. (2016). Effect of bedding planes, their orientation and clay depositions on effective re-injection of produced brine into clay rich deep sandstone formations: Implications for deep earth energy extraction. *Applied Energy*, *161*, 24–40.

Doughty, C., & Pruess, K. (2004). Modeling supercritical carbon dioxide injection in heterogeneous porous media. *Vadose Zone Journal*, *3*(3), 837–847.

Dupont, E., Zhang, T., Tilke, P., Liang, L., & Bailey, W. (2018). Generating realistic geology conditioned on physical measurements with generative adversarial networks. arXiv preprint arXiv 180203065.

Gelhar, L. W. (1993). *Stochastic subsurface hydrology*. Upper Saddle River, NJ: Prentice-Hall.

Goodfellow, I. (2016). Nips 2016 tutorial: Generative adversarial networks. arXiv preprint arXiv 170100160.

Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). *Deep learning*. Cambridge: MIT press.

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

Huan, X., & Marzouk, Y. M. (2013). Simulation-based optimal Bayesian experimental design for nonlinear systems. *Journal of Computational Physics*, *232*(1), 288–317.

Isola, P., Zhu, J.-Y., Zhou, T., & Efros, A. A. (2017). arXiv preprint.

Jeong, H., & Srinivasan, S. (2016). Fast assessment of CO_2 plume characteristics using a connectivity based proxy. *International Journal of Greenhouse Gas Control*, *49*, 387–412.

Jeong, H., & Srinivasan, S. (2017). Fast selection of geologic models honoring CO_2 plume monitoring data using Hausdorff distance and scaled connectivity analysis. *International Journal of Greenhouse Gas Control*, *59*, 40–57.

Jeong, H., Sun, A. Y., Lee, J., & Min, B. (2018). A learning-based data-driven forecast approach for predicting future reservoir performance. *Advances in Water Resources*, *118*, 95–109.

Jiang, X. (2011). A review of physical modelling and numerical simulation of long-term geological storage of CO_2. *Applied energy*, *88*(11), 3557–3566.

Keating, E. H., Harp, D. H., Dai, Z., & Pawar, R. J. (2016). Reduced order models for assessing CO_2 impacts in shallow unconfined aquifers. *International Journal of Greenhouse Gas Control*, *46*, 187–196.

Kleijnen, J. P. (2009). Kriging metamodeling in simulation: A review. *European journal of operational research*, *192*(3), 707–716.

Laloy, E., Hérault, R., Jacques, D., & Linde, N. (2017). Efficient training-image based geostatistical simulation and inversion using a spatial generative adversarial neural network. arXiv preprint arXiv 1708 04975.

Lewicki, J., Oldenburg, C., Dobeck, L., & Spangler, L. (2007). Surface CO_2 leakage during two shallow subsurface CO_2 releases. *Geophysical Research Letters*, *34*, L24402. https://doi.org/10.1029/2007GL032047

Li, H. (2014). Conditional simulation of flow in heterogeneous porous media with the probabilistic collocation method. *Communications in Computational Physics*, *16*(4), 1010–1030.

Li, H., & Zhang, D. (2007). Probabilistic collocation method for flow in porous media: Comparisons with other stochastic methods. *Water Resources Research*, *43*, W09409. https://doi.org/10.1029/2006WR005673

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3431–3440).

Lucia, D. J., Beran, P. S., & Silva, W. A. (2004). Reduced-order modeling: New approaches for computational physics. *Progress in aerospace sciences*, *40*(1-2), 51–117.

Luo, F., Xu, R.-N., & Jiang, P.-X. (2013). Numerical investigation of the influence of vertical permeability heterogeneity in stratified formation and of injection/production well perforation placement on CO_2 geological storage with enhanced CH_4 recovery. *Applied energy*, *102*, 1314–1323.

Ma, X., & Zabaras, N. (2009). An efficient Bayesian inference approach to inverse problems based on an adaptive sparse grid collocation method. *Inverse Problems*, *25*(3), 035013.

Ma, X., & Zabaras, N. (2011). Kernel principal component analysis for stochastic input model generation. *Journal of Computational Physics*, *230*(19), 7311–7331.

Marrel, A., Iooss, B., Van Dorpe, F., & Volkova, E. (2008). An efficient methodology for modeling complex computer codes with Gaussian processes. *Computational Statistics & Data Analysis*, *52*(10), 4731–4744.

Mirza, M., & Osindero, S. (2014). Conditional generative adversarial nets. *arXiv preprint arXiv 14111784*.

Mo, S., Zhu, Y., Zabaras, N., Shi, X., & Wu, J. (2018). Deep convolutional encoder-decoder networks for uncertainty quantification of dynamic multiphase flow in heterogeneous media. arXiv preprint arXiv 180700882.

Müller, J., Shoemaker, C. A., & Piché, R. (2013). So-mi: A surrogate model algorithm for computationally expensive nonlinear mixed-integer black-box global optimization problems. *Computers & Operations Research*, *40*(5), 1383–1400.

Oladyshkin, S., Class, H., Helmig, R., & Nowak, W. (2011). An integrative approach to robust design and probabilistic risk assessment for CO_2 storage in geological formations. *Computational Geosciences*, *15*(3), 565–577.

Oliver, D. S., & Chen, Y. (2011). Recent progress on reservoir history matching: A review. *Computational Geosciences*, *15*(1), 185–221.

Pawar, R. J., Bromhal, G. S., Carey, J. W., Foxall, W., Korre, A., Ringrose, P. S., et al. (2015). Recent advances in risk assessment and risk management of geologic CO_2 storage. *International Journal of Greenhouse Gas Control*, *40*, 292–311.

Queipo, N. V., Haftka, R. T., Shyy, W., Goel, T., Vaidyanathan, R., & Tucker, P. K. (2005). Surrogate-based analysis and optimization. *Progress in Aerospace Sciences*, *41*(1), 1–28.

Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv 151106434.

Remy, N., Boucher, A., & Wu, J. (2009). *Applied geostatistics with SGeMS: A user's guide*. Cambridge: Cambridge University Press.

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

Rubin, Y. (2003). *Applied stochastic hydrogeology*. Oxford, England, UK: Oxford University Press.

Sarma, P., Durlofsky, L. J., Aziz, K., Chen, W. H., et al. (2007). A new approach to automatic history matching using kernel PCA. In *SPE Reservoir Simulation Symposium* (pp. 1–19). Society of Petroleum Engineers.

Schöniger, A., Nowak, W., & Hendricks Franssen, H.-J. (2012). Parameter estimation by ensemble Kalman filters with transformed data: Approach and application to hydraulic tomography. *Water Resources Research*, *48*, W04502. https://doi.org/10.1029/2011WR010462

Sun, A. Y. (2018). Discovering state-parameter mappings in subsurface models using generative adversarial networks. *Geophysical Research Letters*, *45*, 11,137–11,146. https://doi.org/10.1029/2018GL080404

Sun, W., & Durlofsky, L. J. (2017). A new data-space inversion procedure for efficient uncertainty quantification in subsurface flow problems. *Mathematical Geosciences*, *49*(6), 679–715.

Sun, A. Y., Jeong, H., González-Nicolás, A., & Templeton, T. C. (2018). Metamodeling-based approach for risk assessment and cost estimation: Application to geological carbon sequestration planning. *Computers & Geosciences*, *113*, 70–80.

Sun, N.-Z., & Sun, A. (2015). *Model calibration and parameter estimation: For environmental and water resource systems*. Salmon Tower Building, NY: Springer.

Sun, A. Y., Zeidouni, M., Nicot, J.-P., Lu, Z., & Zhang, D. (2013). Assessing leakage detectability at geologic CO_2 sequestration sites using the probabilistic collocation method. *Advances in Water Resources*, *56*, 49–60.

Swischuk, R., Mainini, L., Peherstorfer, B., & Willcox, K. (2018). Projection-based model reduction: Formulations for physics-based machine learning. Computers & Fluids.

Tartakovsky, D. M. (2013). Assessment and management of risk in subsurface hydrology: A review and perspective. *Advances in Water Resources*, *51*, 247–260.

Vondrick, C., Pirsiavash, H., & Torralba, A. (2016). Generating videos with scene dynamics. Advances In Neural Information Processing Systems (pp. 613–621).

Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, *13*(4), 600–612.

Wood, E. F., Roundy, J. K., Troy, T. J., van Beek, L. P. H., Bierkens, M. F. P., Blyth, E., et al. (2011). Hyperresolution global land surface modeling: Meeting a grand challenge for monitoring Earth's terrestrial water. *Water Resources Research*, *47*, W05301. https://doi.org/10.1029/2010WR010090

Xiu, D., & Karniadakis, G. E. (2002). The Wiener–Askey polynomial chaos for stochastic differential equations. *SIAM Journal on Scientific Computing*, *24*(2), 619–644.

Zeng, X., Ye, M., Burkardt, J., Wu, J., Wang, D., & Zhu, X. (2016). Evaluating two sparse grid surrogates and two adaptation criteria for groundwater Bayesian uncertainty quantification. *Journal of Hydrology*, *535*, 120–134.

Zhang, D. (2001). *Stochastic methods for flow in porous media: Coping with uncertainties*. Amsterdam, Netherlands: Elsevier.

Zhang, D., & Lu, Z. (2004). An efficient, high-order perturbation approach for flow in random porous media via Karhunen–Loeve and polynomial expansions. *Journal of Computational Physics*, *194*(2), 773–794.

Zhang, J., Zeng, L., Chen, C., Chen, D., & Wu, L. (2015). Efficient Bayesian experimental design for contaminant source identification. *Water Resources Research*, *51*, 576–598. https://doi.org/10.1002/2014WR015740

Zhong, Z., & Carr, T. R. (2019). Geostatistical 3D geological model construction to estimate the capacity of commercial scale injection and storage of CO_2 in Jacksonburg-Stringtown oil field, West Virginia, USA. *International Journal of Greenhouse Gas Control*, *80*, 61–75.

Zhou, H., Gomez-Hernandez, J. J., Franssen, H.-J. H., & Li, L. (2011). An approach to handling non-Gaussianity of parameters and state variables in ensemble Kalman filtering. *Advances in Water Resources*, *34*(7), 844–864.

Zhu, J.-Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. arXiv preprint.

Zhu, Y., & Zabaras, N. (2018). Bayesian deep convolutional encoder–decoder networks for surrogate modeling and uncertainty quantification. *Journal of Computational Physics*, *366*, 415–447.