



Convolutional neural networks (CNN) for feature-based model calibration under uncertain geologic scenarios

Syamil Mohd Razak¹ · Behnam Jafarpour¹

Received: 5 August 2019 / Accepted: 28 April 2020 / Published online: 22 June 2020
© Springer Nature Switzerland AG 2020

Abstract

This paper presents convolutional neural network architectures for integration of dynamic flow response data to reduce the uncertainty in geologic scenarios and calibrate subsurface flow models. The workflow consists of two steps, where in the first step the solution search space is reduced by eliminating unlikely geologic scenarios using distinguishing salient flow data trends. The first step serves as a pre-screening to remove unsupported scenarios from the full model calibration process in the second step. For this purpose, a convolutional neural network (CNN) with a cross-entropy loss function is designed to act as a classifier in predicting the likelihood of each scenario based on the observed flow responses. In the second step, the selected geologic scenarios are used in another CNN with an ℓ_2 -loss function (as a regression model) to perform model calibration. The regression CNN model (step 2) learns the inverse mapping from the production data space to the low-rank representation of the model realizations within the feasible set. Once the model is trained off-line, a fast feed-forward operation on the observed historical production data (input) is used to reconstruct a calibrated model. The presented approach offers an opportunity to utilize flow data in identifying plausible geologic scenarios, results in an off-line implementation that is conveniently parallelizable, and can generate calibrated models in real time, i.e., upon availability of data and without in-depth technical expertise about model calibration. Several synthetic Gaussian and non-Gaussian examples are used to evaluate the performance of the method.

Keywords Inverse problems · Machine learning · Uncertainty · Geologic scenarios · Convolutional neural networks · Model calibration

1 Introduction

Developing a computer model that can be used to simulate and predict the performance of subsurface flow systems often constitutes a complex multi-stage process that involves acquisition, processing, integration, and interpretation of various types of data. The model development process is composed of hierarchical sequential steps, each with its own sources of uncertainty and subjectivity that must be captured and propagated along the chain. Some of the major sources of uncertainty tend to be present at early

stages, where very limited data is used to develop the reservoir structure and the conceptual geologic model for the reservoir. Faced with limited data, geoscientists often have to resort to subjective assumptions in developing a conceptual model that can be used by subsurface flow modelers. Figure 1 shows a typical model development process, in which at the early stages geoscientists include structural uncertainties in defining the top and base of a reservoir due to uncertainties associated with seismic time-depth conversion. This uncertainty arises because of the limited number of well penetration points that are available to obtain the conversion coefficients. Efforts to consider structural and fault placement uncertainties require complex parameterization techniques [5]. As the process continues down the chain, other sources of uncertainty present themselves; for example, in the case of a fluvial environment, channel thickness, width, and azimuth are some of the common sources of uncertainty. In practice, stochastic representation of these parameters is informed by similar fluvial systems and through qualitative expert elicitation [3, 41].

✉ Behnam Jafarpour
behnam.jafarpour@usc.edu

Syamil Mohd Razak
mohdraza@usc.edu

¹ Mork Family Department of Chemical Engineering and Material Science, Viterbi School of Engineering, University of Southern California, Los Angeles, CA 90089, USA

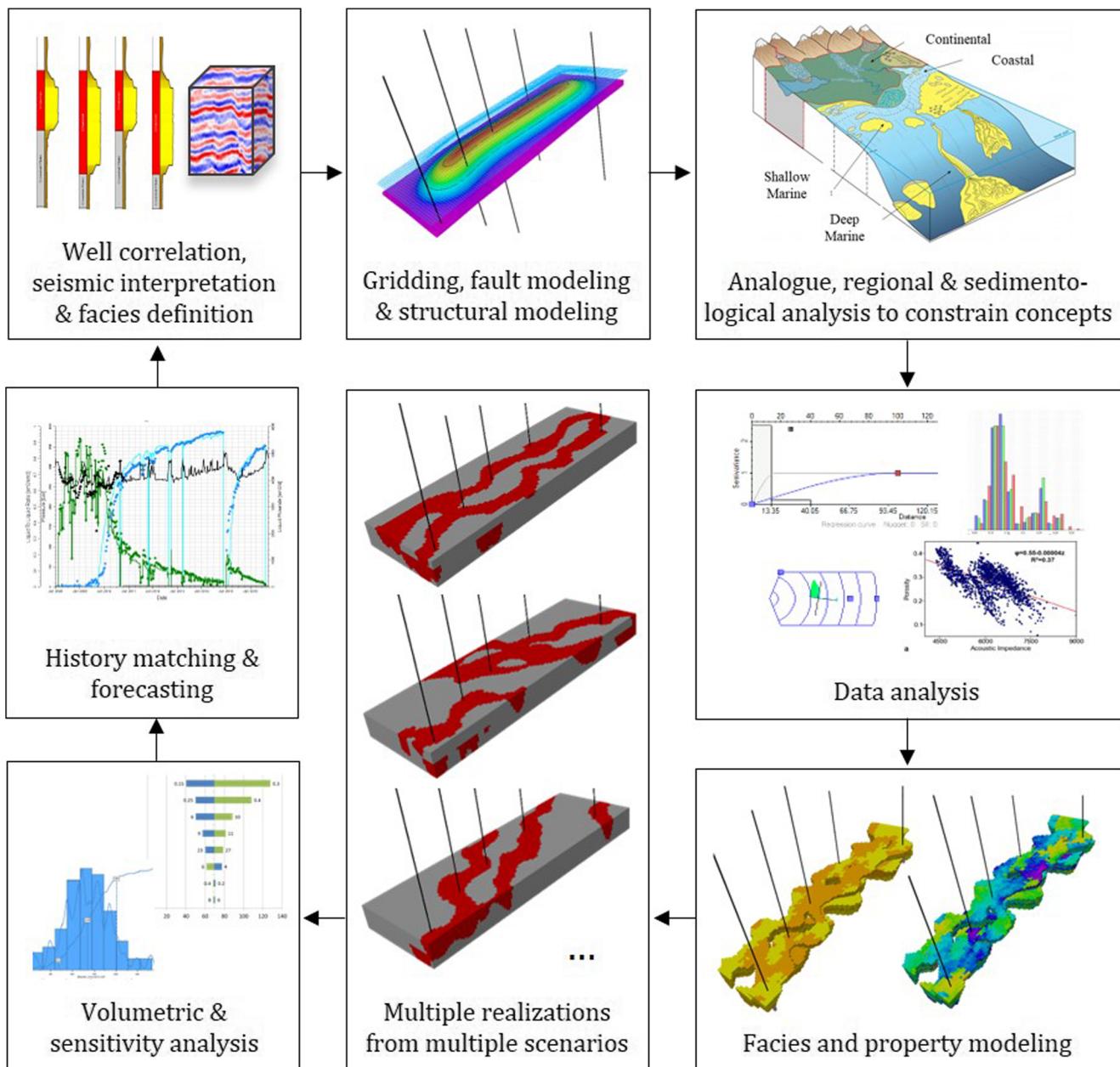


Fig. 1 Conventional workflow that considers multiple geologic scenarios and other sources of uncertainty in constructing prior models for history matching to generate reliable dynamic models

Conventional workflows typically do not include dynamic response data in reducing the uncertainty and subjectivity that the geologists deal with. In most cases, only well testing data from the exploration and appraisal wells are available at early stages. Hence, pressure transient analysis is performed to provide coarse insight on the shape and size of the reservoir, which is used by the geologist to ascertain reservoir parameters. As production and injection wells are drilled, the subsurface team will have more direct samples from the formation and inferred (from production data) information about the connectivity between wells [7]. As

a green field matures to a brown field, fine-scale reservoir models become highly relevant as a tool for guiding future infill well placement and developing reservoir management strategies. Traditionally, integration of dynamic data into reservoir models is performed by reservoir engineers to update the initial models while honoring a single conceptual geologic model that is provided by the geologists, a task that is performed either manually or in an assisted way using automatic model calibration workflows [9, 13, 14]. This assumption has been present in many of the promising history matching methods that have been developed through

the years [36]. The underlying assumption in these methods is that the conceptual geologic model is known and is used to generate realizations with alternative spatial distribution of the corresponding geologic features (spatial patterns). However, since each scenario is characterized by distinct geologic features that define the connectivity in the reservoir [40, 49], disregarding the uncertainty and subjectivity in the geologic scenario can result in significant underestimation of the uncertainty and can introduce bias in predicting the flow performance [39]. As an example, a meandering fluvial environment tends to have sinuous curves and point bars while an anastomosing fluvial environment will have straight channels meeting at acute angles. The resulting models based on these two conceptual environments can lead to very different fluid flow and displacement patterns that play a critical role in planning infill drilling and overall field development strategies.

To enable the feedback from dynamic data to rank conceptual geologic models, novel data integration methods that do not assume a single geologic scenario are needed. A common approach is to adopt the Popper-Bayes philosophy that observations can only be used to falsify, and not deduce, models or theories [48]. In the context of subsurface flow models, this approach leads to falsification of geologic scenarios that are not supported by additional data. While dynamic flow response data are known to contain low-resolution (aggregate) information support that are sensitive only to large-scale features in the reservoir, they can provide important information to support or reject some of the proposed geologic scenarios. Two classes of methods have been proposed in the literature to implement model calibration under uncertainty in the geologic scenario. The first class performs scenario falsification before model calibration [6, 12, 17, 37, 38, 43]. The second class of methods performs scenario falsification after (and in some methods, simultaneously) model calibration [4, 10, 16, 18, 20].

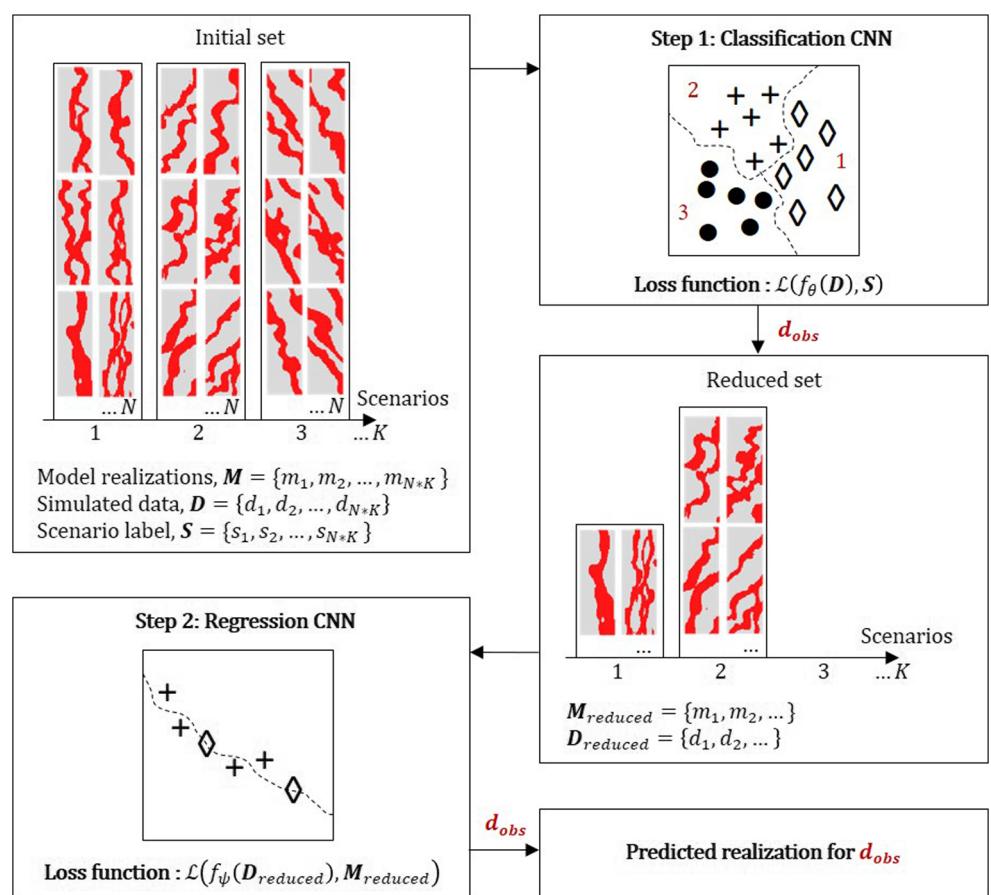
In the first class of methods, some authors have used kernel density estimation on simulated data (from realizations of multiple scenarios) where distance of observed data to distributions of labeled data is used to infer its likelihood (online method) [6, 12, 37, 38]. In Scheidt et al. [43], decomposition via wavelet transform is used to analyze differences in patterns for measuring global similarity between simulated and observed data. Classification and regression tree (CART) method has also been utilized to find the probability of each outcome (e.g., water breakthrough trend) given uncertain geologic parameters [17]. Pirot et al. [38] use multiple global spatial statistics of simulated data from multiple scenarios to the observed data to identify plausible scenarios. In the second class of methods, Brunetti et al. [4] perform scenario selection after posterior sampling and

acknowledge that the method is computationally expensive for large models. In Khodabakhshi and Jafarpour [20], mixture modeling is combined with ensemble-based data assimilation to include multiple plausible scenarios (that are simultaneously screened) as prior knowledge for model calibration. In [10, 18, 19], inverse modeling with sparsity-promoting regularization techniques and sparse geologic dictionaries are used to identify relevant geologic scenarios and obtain inversion solutions.

In this paper, we introduce a machine learning-based workflow to perform model calibration under uncertainty in the geologic scenario. This approach is illustrated in Fig. 2 where given K possible geologic scenarios, each scenario is used to simulate (N) realizations of the reservoir model properties (e.g., permeability distribution) to represent within-scenario variabilities. The resulting realizations summarize the geologic features that exist in each geologic scenario and can be used to generate the corresponding dynamic flow responses (as training data labels). The presented approach consists of two steps, where the first step is used to reduce the number of geologic scenarios and the second step performs model calibration based on the selected scenarios in step 1. The main motivation behind geologic scenario selection is to prune the search space prior to performing detailed model calibration. For the first step, a convolutional neural network (CNN) [25] is trained as a classifier for screening geologic scenarios based on their global flow responses. The classification CNN is trained using a cross-entropy loss function to estimate the likelihood that a given historical data (\mathbf{d}_{obs}) belongs to any of the possible geologic scenarios. The resulting likelihoods are used to prepare the realizations that are used in step 2, to train another CNN that learns the inverse mapping between production data and model realizations.

In step 2 of our workflow, a CNN architecture with an ℓ_2 -loss function is used as a regression model. The CNN learns non-linear low-dimensional manifolds between realizations and the associated production data. The hypothesis is that the probability distribution of data is highly concentrated along the manifold [34]. Each realization (within the reduced space of relevant scenarios) and corresponding production data is connected to each other and can be reached along the manifold. Under this hypothesis, if the data is sampled well enough to cover the structure of the learned manifold, then its regression property can be exploited to predict the realization for a given historical production data. Furthermore, CNN can be viewed as a non-linear generalization of the standard PCA [11, 44]. In this case, by combining the model realizations with their simulated response data, CNN is trained to learn the correspondence between flow response data and the resulting geologic realizations. Once trained off-line, the resulting model can be used in real time to provide a

Fig. 2 Proposed two-step workflow that predicts a model realization for any given historical data d_{obs}



geologic realization that corresponds to a given observed data.

Since the seminal work of LeCun et al. [27] in using CNN to classify hand-written digits, CNN has become more popular due to the increasing amount of available data and advances in computational hardware. The architecture of CNN has also progressed significantly in terms of capacity (here refers to the ability to learn complex relationships) and computational efficiency [23, 31, 47]. Within the subsurface community, variations of neural network models have been applied to problems such as seismic inversions to predict hydrocarbon prospectivity [1, 28, 42], geologic feature extraction for preserving realism in data assimilation [21], well placement optimization [30], parameterization of geologic models [24], and as a proxy or surrogate model to forecast field production or net-present value (NPV) [2, 8, 15, 29, 32, 45].

The application of CNN to our problem is motivated by its superior pattern-learning performance and its convenient adaptability to scenario selection and inverse mapping, simply by modifying the loss function that allows the convolutional filter to learn salient information. For scenario falsification, this is an efficient direct approach that does

not require pre-processing step such as multi-dimensional scaling (MDS) and applying kernel density estimation on the metric space to approximate the probability of each scenario. Multiple non-linear convolutional filters automatically extract distinguishing salient trends between each scenario from the dynamic data through the cross-entropy loss function without any need for a pre-defined distance metric. The probability of each scenario is readily given by the trained model and information loss is minimized as the trained model can be tuned to prevent over-fitting. For model calibration, CNN allows direct inversion by extracting salient production data features and learning the complex manifold between these features and geologic model realizations. During training, CNN implicitly learns the existing trends in Gaussian and non-Gaussian models without the need to specify any pre-defined probability density function (pdf) of the realizations (whether two-point or multi-point statistics). CNN is a convenient and versatile tool that does not require the intervention of domain experts. In this work, we assume that production data contains information that can falsify irrelevant scenarios and there is sufficient support between the production data and model realizations.

2 Methodology

In this section, a brief overview of the CNN fundamentals is presented along with a review of its key components, leaving detailed discussions to relevant references in the computer science literature (e.g., [26, 33]). CNN is a class of supervised learning algorithms that are inspired by the behavior of the visual cortex in the brain, where individual cortical neurons respond to stimuli only in a restricted region of the visual field. The CNN architecture mimics the same behavior by using many filters in several layers to detect different features in the input signal that are important in describing the output. Each layer of a CNN architecture consists of a convolution layer, followed by an activation layer and a pooling layer. The main distinguishing component of the CNN is the convolution layer, which performs mathematical convolution (filtering) operation using a large number of filters, each responsible for detecting specific features of the input data. During training of a CNN, the parameters (weights) that describe each convolution filter are adapted to the specific training data that is provided. Figure 3

a illustrates a convolution operation for extracting salient (spatial/temporal) information from an input matrix. The output of the convolution layer is used as input into the activation layer. Activation layers are an abstraction representing the rate of potential firing action in a neuron. In CNN, activation layers (Fig. 3b) apply non-linear transformations to their input signals to generate the corresponding outputs, which determine whether individual neurons will fire or not (given their input). The transformed outputs of the activation layer are then transferred as input signal to the next layer (pooling). Pooling (or down-sampling) layers (Fig 3c) are designed to reduce the number of parameters and the computation in the network. The main functionality of the pooling layers in CNN is to replace high-dimensional inputs (from the activation layer) with low-dimensional approximations that will be transferred to the next layer in the network. Compared with dense artificial neural networks and other machine learning algorithms, CNN represents complex non-linear systems using reduced number of parameters through weight sharing (filters) and by taking advantage of local spatial coherence and distributed representation.

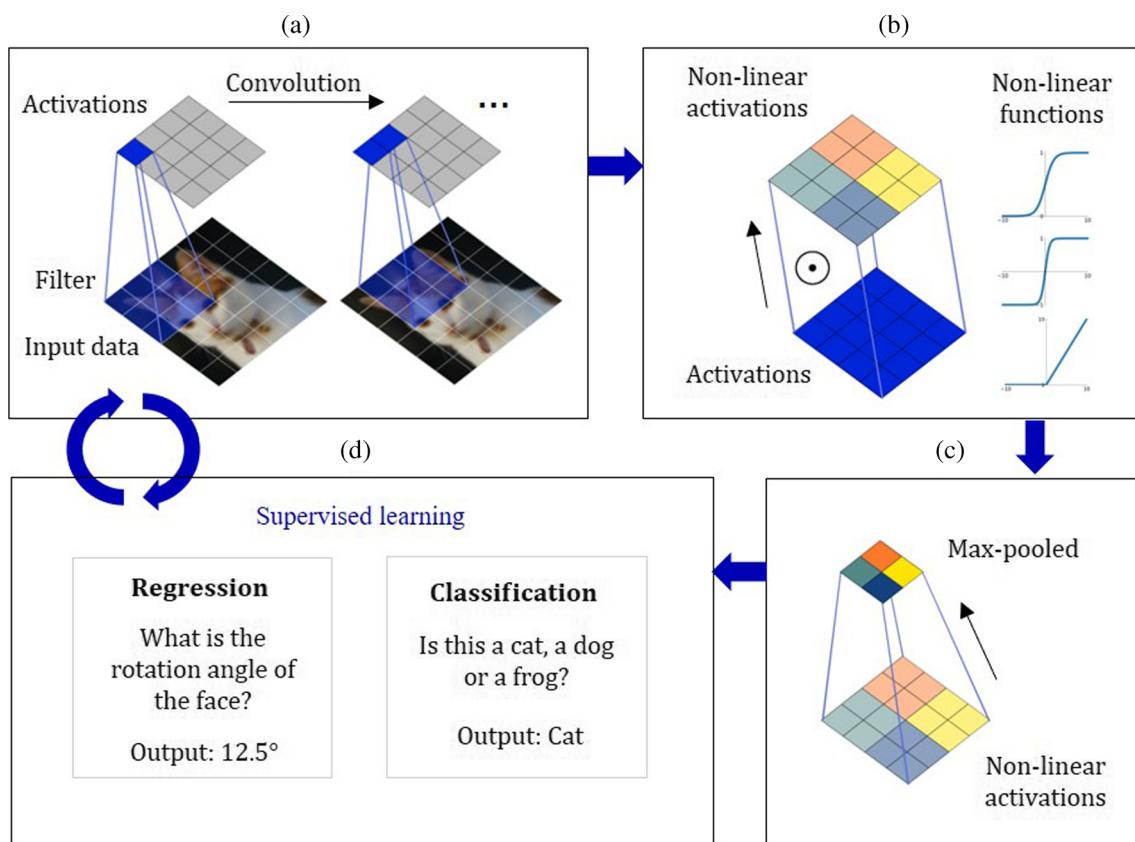


Fig. 3 (a) Convolution operation (of a trainable filter) on an input results in activations. (b) Non-linear transformation on activations to generate non-linear activations. (c) Pooling or down-sampling layer reduces the number of parameters. (d) In a supervised learning

setting, these non-linear activations are compared with the expected corresponding output variable (classification for a discrete variable or regression for a continuous variable) and the filter is updated to minimize the difference. *Images not drawn to scale*

The power of CNN for pattern recognition and classification with complex datasets is well established in the literature. In this paper, we take advantage of this strength of CNN to classify geologic scenarios based on their dynamic flow responses (classification with CNN) and to associate the flow responses with the spatial distribution of important rock flow properties (through a regression CNN). The CNN implementations are based on the open-source machine learning framework Keras (version 2.2.4) with Tensorflow library as the backend (version 1.15). Table 1 lists the actual Keras functions and hyperparameters used for examples 1 and 2 in Section 3. For examples 3 and 4, simply modify the sizes of the input and output according to the general formulation provided below. In this paper, the training is performed using the Adam stochastic optimization method [22] combined with the dropout regularization method [46] on the dense layer (to prevent over-fitting) with dropout rate of 0.2.

2.1 Classifier model

In this paper, the focus is on using production response data as input to either identify the geologic scenario as output (classification model, $\mathcal{D}(\cdot)$) or the corresponding geologic realization as output (regression model, $\mathcal{H}(\cdot)$). Figure 4 (Top) shows the actual architecture used for both models (Gaussian and non-Gaussian 2D examples). Each box represents the output of a mathematical operation on the preceding box. In this section, we explain each of these operations. To keep our notations general for any depth of CNN architecture, we consider for the l -th layer with input $\mathbf{x}^l \in \mathbb{R}^{H^l \times W^l \times D^l}$ as a 3-dimensional tensor where for the first convolution layer ($l = 1$), H^1 is the production time interval and W^1 is the features considered (i.e., water-cut, oil production rate, and bottom-hole pressure, see Section 3.1). As the depth of the input is 1 ($D^1 = 1$), the input tensor is reduced to a matrix. In our implementation, the input to the architecture is provided in mini-batches of size b . Hence, the input to the first convolutional layer becomes a 4-dimensional tensor of $\mathbf{x}^1 \in \mathbb{R}^{H^1 \times W^1 \times D^1 \times b}$. For simplicity, we assume that $b = 1$. With this definition,

any element in \mathbf{x}^l has indices (i^l, j^l, d^l) such that $0 \leq i^l < H^l$, $0 \leq j^l < W^l$, $0 \leq d^l < D^l$. The output of layer l is denoted as x^{l+1} and indices $(i^{l+1}, j^{l+1}, d^{l+1})$ such that $0 \leq i^{l+1} < H^{l+1}$, $0 \leq j^{l+1} < W^{l+1}$, $0 \leq d^{l+1} < D^{l+1}$, where $H^{l+1} \times W^{l+1} \times D^{l+1}$, are its dimensions.

The first convolution layer has D kernels each of size $H \times W \times D^1$. A collection of kernels are denoted as filters of $f^{H \times W \times D^1 \times D}$ and each element is accessed with (i, j, d^l, d) such that $0 \leq i < H$, $0 \leq j < W$, $0 \leq d^l < D^l$, $0 \leq d < D$. For clarity, we omit any bias term and assume a filter of stride 1 with no padding, and the convolution operation for all d is mathematically defined as:

$$\mathbf{a}_{i^{l+1}, j^{l+1}, d} = \sum_{i=0}^H \sum_{j=0}^W \sum_{d^l=0}^{D^l} f_{i, j, d^l, d} \times \mathbf{x}^l_{i^{l+1}+i, j^{l+1}+j, d^l} \quad (1)$$

In Fig. 4 (Bottom), sample activation of kernels/filters of dimension $5 \times 2 \times 1$ is shown. The activation $\mathbf{a} \in \mathbb{R}^{(H^l-H+1) \times (W^l-W+1) \times D}$ is scaled through a channel-wise batch-normalization layer to increase stability in training before passing it through a rectified linear unit (ReLU) as a state-of-the-art non-linear layer. The ReLU operation is element-wise and mathematically defined as:

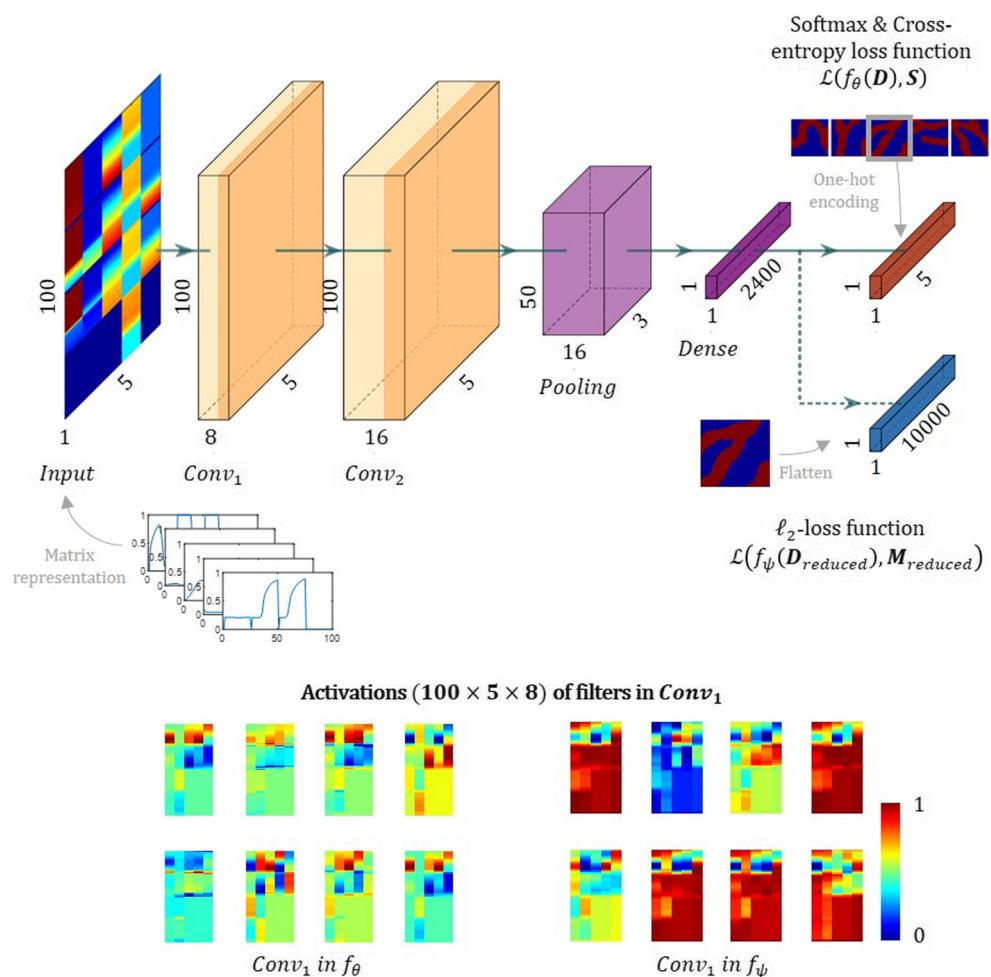
$$\mathbf{h}_{i, j, d}^{l+1} = \max \{0, \mathbf{a}_{i, j, d}^l\} \quad (2)$$

A hidden layer is defined as a series of convolution, batch-normalization, and ReLU layers. The combined mechanism (represented as *Conv_l* in Fig. 4 (Top)) is such that each convolution filter extracts selective spatial and temporal information. For example, in convolving the filters on a geologic realization of a meandering fluvial channel, filters that detect curved edges will result in positive regions of activation while other filters that detect sharp acute edges will have non-positive regions of activation. When this activation is passed through a ReLU layer, only positive regions remain active. When we convolve these filters on a matrix representation of time-series data within a fixed period of time, the filters extract salient features such as water breakthrough time, pressure trend, and cumulative oil production. Next, we utilize a max-pooling (a form of

Table 1 Keras functions corresponding to Fig. 4 and parameters used in this study (for examples 1 and 2)

Shorthand notation	Keras function (version 2.2.4)	Parameters
<i>Conv_{1/2}</i>	keras.layers.Conv2D	filters = 8/16, kernel_size = 5×2, activation = ‘relu’
<i>Pooling</i>	keras.layers.MaxPooling2D	pool_size = 2, padding = ‘same’
<i>Dense</i>	keras.layers.Dense, keras.layers.Dropout	units = 5/10000, rate = 0.2
$\mathcal{L}(f_\theta(\cdot), \cdot)$	keras.losses.categorical_crossentropy	
$\mathcal{L}(f_\psi(\cdot), \cdot)$	keras.losses.mean_squared_error	
Optimizer	keras.optimizers.Adam	$\alpha = 0.001, \beta_1 = 0.9, \beta_2 = 0.9$

Fig. 4 (Top) Architecture and dimensions of the CNNs. (Bottom) Activations of the 5×2 filters in the first convolution layer



subsampling) layer to reduce a sub-region of an activation block and keep only its maximum value (*Pooling* in Fig. 4 (Top)). The max-pooling operation takes the form:

$$\mathbf{p}_{i^{l+1}, j^{l+1}, d} = \max_{\{0 \leq i < H, 0 \leq j < W\}} \mathbf{h}_{i^{l+1} \times H+i, j^{l+1} \times W+j, d}^l \quad (3)$$

In this case, $\mathbf{p} \in \mathbb{R}^{H^{l+1} \times W^{l+1} \times D^{l+1}}$ where the pooling filter of size $H \times W$ evenly divides the input. The hidden layer can be repeated depending on the complexity of the dataset. In our implementation, two hidden layers were used to capture the complex non-linear relationship within the data set. At this stage, tensor \mathbf{p} has distributed representations of our input and is reshaped into a tensor of order one (vector) and is passed through a dense layer (*Dense* in Fig. 4 (Top)) to enforce the global structure of the data. We denote the output vector of this dense layer as $\mathbf{z} \in \mathbb{R}^{H^{l+1} \cdot W^{l+1} \cdot D^{l+1}}$

If there are C classes of geologic scenarios to be considered (where $C = H^{l+1} \cdot W^{l+1} \cdot D^{l+1}$), the training output is represented as a one-hot vector encoding $\mathbf{y}_d \in \mathbb{R}^C$, where each column represents the likelihood of each sample

to belong to a scenario. The cross-entropy loss function is computed as:

$$\mathcal{L}(\mathbf{z}, \mathbf{y}_d) = \sum_{i=0}^C -y_{di} \log \sigma(z_i) + (1 - y_{di}) \log(1 - \sigma(z_i)) \quad (4)$$

For ease of discussion, the optimized weights of $\mathcal{D}(\cdot)$ are collectively denoted as θ . The feature extraction step (from production data, \mathbf{D}) is denoted as f_θ and the loss function as $\mathcal{L}(f_\theta(\mathbf{D}), \mathbf{S})$ where \mathbf{S} is the provided data label. For each scenario label, the cross-entropy loss minimizes the Kullback-Leibler divergence between the empirical distribution of the input data and the predicted distribution to learn decision boundary between scenario labels.

2.2 Regression model

If the objective is to use the flow response data to estimate the corresponding geologic realization for any given historical data \mathbf{d}_{obs} , the last layer of network discussed for classification will be set up to solve a regression problem. In this case, a subset of geologic realizations ($\mathbf{M}_{reduced}$) and their corresponding simulated production data ($\mathbf{D}_{reduced}$)

Table 2 Proposed two-stage workflow

<i>Data preparation</i>	
1.	Identify geologic uncertainties and partition into K scenarios
2.	For each scenario, generate N realizations
3.	Run $N \times K$ forward simulations to get model-data pair (\mathbf{M} , \mathbf{D})
4.	Construct scenario label S for \mathbf{D}
<i>Stage 1: Geologic scenario selection $\mathcal{D}(\cdot)$</i>	
5.	Train a classifier model $\mathcal{D}(\cdot)$ with cross-entropy loss function, $\mathcal{L}(f_\theta(\mathbf{D}), S)$
6.	Predict the probability of each geologic scenario, $\mathbf{g} = \mathcal{D}(\mathbf{d}_{obs})$
7.	Form subset ($\mathbf{M}_{reduced}$, $\mathbf{D}_{reduced}$) according to proportion \mathbf{g}
<i>Stage 2: Model calibration $\mathcal{H}(\cdot)$</i>	
8.	Train regression model $\mathcal{H}(\cdot)$ with ℓ_2 -loss function, $\mathcal{L}(f_\psi(\mathbf{D}_{reduced}), \mathbf{M}_{reduced})$
9.	Predict the realization, $\hat{\mathbf{m}} = \mathcal{H}(\mathbf{d}_{obs})$

are used for training, where a geologic realization $\mathbf{m} \in \mathbb{R}^{N \times M}$ is flattened to $\mathbf{y}_h \in \mathbb{R}^{N \cdot M}$. With the output of the dense layer denoted as $\mathbf{z} \in \mathbb{R}^{N \cdot M}$, the resulting loss function takes the form

$$\mathcal{L}(\mathbf{z}, \mathbf{y}_h) = \sum_{i=0}^{N \cdot M} (y_{hi} - z_i)^2 \quad (5)$$

In this case, the optimized weights of $\mathcal{H}(\cdot)$ are denoted as ψ and the feature extraction operation is represented by f_ψ and the loss function as $\mathcal{L}(f_\psi(\mathbf{D}_{reduced}), \mathbf{M}_{reduced})$. To improve the effectiveness of the approach, instead of a grid representation of the geologic realizations, a low-rank feature-based representation is used.

2.3 Dimensionality reduction techniques

A geologic realization (continuous Gaussian or discrete facies property field), represented as $\mathbf{u} \in \mathbb{R}^n$, can have a linear expansion in a sub-space defined by specialized basis functions φ_i such that $0 \leq i < n$ as:

$$\mathbf{u} = \sum_{i=0}^n v_i \varphi_i \quad (6)$$

The expansion coefficients $\mathbf{v} = [v_0, v_1, \dots, v_{n-1}]^T$ represent \mathbf{u} in $\{\varphi_i\}_{i=0,1,2,\dots,n-1}$ and can be sparsely approximated by s terms where $s \ll n$ and $(n-s)$ of the coefficients are approximately zero. The approximation

quality depends on the complexity of the geologic realization and the information in φ_j such that $0 \leq j < s$. The linear approximation of \mathbf{u} in $\{\varphi_j\}_{j=0,1,2,\dots,s-1}$ is expressed as:

$$\mathbf{u} = \sum_{i=0}^n v_i \varphi_i \approx \sum_{j=0}^s v_j \varphi_j \quad (7)$$

where Eqs. 6 and 7 can be written in matrix notation as

$$\begin{aligned} \mathbf{u}_{n \times 1} &= \boldsymbol{\phi}'_{n \times n} \mathbf{v}_{n \times 1} = \boldsymbol{\phi}'_n \mathbf{v} \\ &\approx \boldsymbol{\phi}'_{n \times s} \mathbf{v}_{s \times 1} = \boldsymbol{\phi}'_s \mathbf{v} \end{aligned} \quad (8)$$

Here, the columns of $\boldsymbol{\phi}'_s$ contain the significant basis elements $\{\varphi_j\}_{j=0,1,2,\dots,s-1}$. Using an orthonormal basis such as PCA, s is identified as the leading φ_j basis elements with the largest eigenvalues. The low-rank vector of expansion coefficients is used to approximately represent each geologic realization. That is, Eq. 5 can be expressed as

$$\mathcal{L}(\mathbf{z}, \mathbf{v}) = \sum_{i=0}^s (v_i - z_i)^2 \quad (9)$$

where $\mathbf{v} \in \mathbb{R}^s$ and $\mathbf{z} \in \mathbb{R}^s$.

As an example, the total number of parameters (ψ) for $\mathcal{H}(\cdot)$ in Fig. 4 is 24,011,384 where 24,010,000 is within the dense layer. With low-rank representation of $s = 7$ that covers $\sim 75\%$ of the variance in a Gaussian dataset, the number of parameters to be learned in the dense layer is reduced to 16,807.

Table 3 Uncertain geologic parameters as input to SGS algorithm

Scenario	Variogram azimuth (°)	Variogram (major, minor) range (meters)
1	Isotropic	500 ± 100
2	0 ± 5	$(500 \pm 100, 200 \pm 50)$
3	45 ± 5	$(500 \pm 100, 200 \pm 50)$
4	90 ± 5	$(500 \pm 100, 200 \pm 50)$
5	135 ± 5	$(500 \pm 100, 200 \pm 50)$

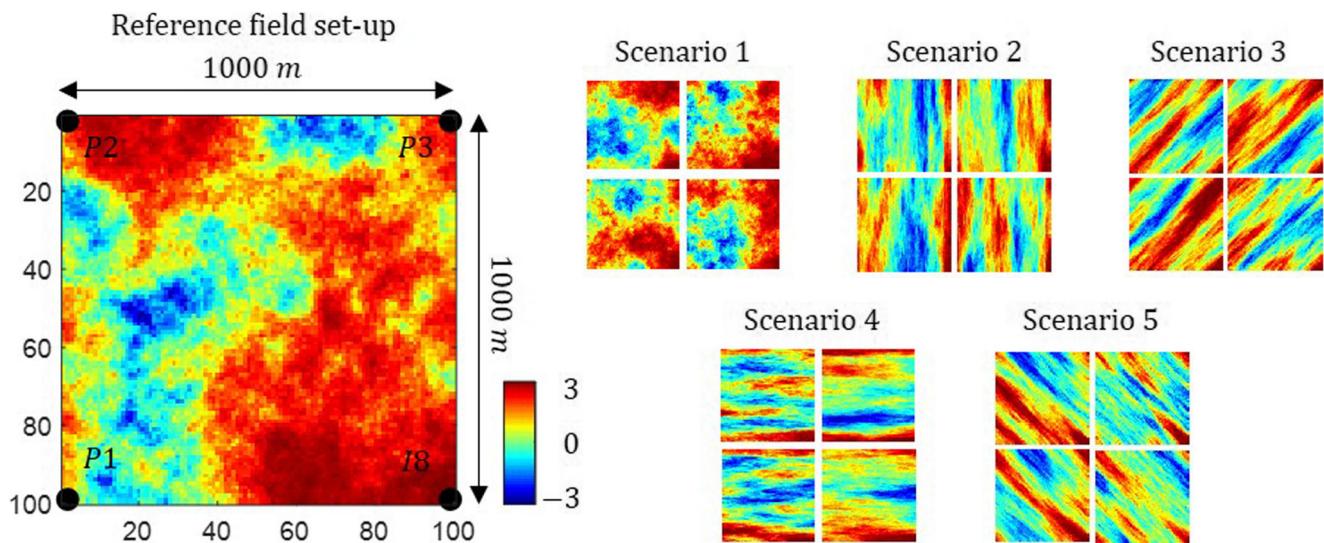


Fig. 5 Reference field set-up and samples of conditioned Gaussian realizations

3 Numerical experiments and results

In this section, the performance of the proposed workflow (Table 2) is demonstrated with four numerical experiments. The first experiment is based on synthetic multi-Gaussian realizations with uncertain variogram parameters that represent the uncertainty in the continuity model. In the second and third experiments, five 2D and four 3D scenarios of fluvial reservoirs are considered. For the fluvial channels, the uncertainty in the geologic scenario is reflected through channel azimuth, thickness-to-width ratio, and connectivity patterns. In the fourth experiment, we test the workflow on a field-like example based on Volve field in the North sea. In these experiments, the reference cases are assumed to be known and different modeling methods are used to represent uncertain scenarios.

3.1 Example 1: Synthetic 2D Gaussian model

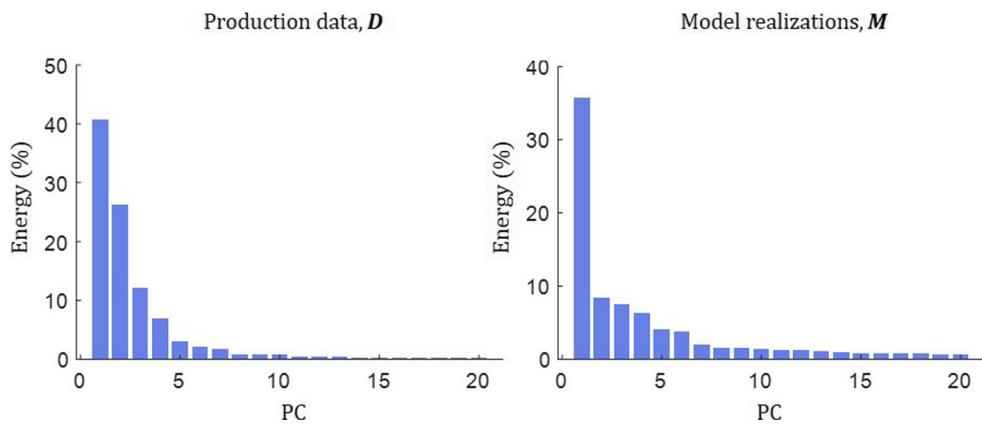
In this example, a two-dimensional reservoir of dimension 1000 m × 1000 m that is discretized into a 100 × 100 domain is considered. The experiment is based on a two-phase flow system with four wells, each located at a corner of the reservoir. Three of the wells are producers and one is an injector located at the southeastern corner. The uncertain parameters of the variogram model for this example are listed in Table 3. For each of the five scenarios ($K = 5$), 500 conditioned realizations ($N = 500$) are generated as in Fig. 5. For each realization, a Gaussian field z' is simulated using the Sequential Gaussian Simulation (SGS) algorithm with Petrel and transformed to get the corresponding porosity model as $\phi = 0.05z' + 0.25$, and the permeability model as $k = 5 + 1000000\phi^{10}$ [35]. The porosity values range from 0.1 to 0.4 while the permeability values range

from 5 to 1054 mD. Approximately one pore volume is injected into the reservoir over 6 years of simulation time, during which production data is collected every 3 months. Since the amount of production data collected for each realization is the same, we represent the data as a matrix (*Input* in Fig. 4 (*Top*)) and assign a label that corresponds to the related scenario. If data is missing for some time steps, the time steps are omitted from the matrix representation. The labeled data set is shuffled and split into training and validation data set with a ratio of 4 : 1. Our testing data set has the same number of elements as the validation data set.

	1	2	3	4	5	90.0%	10.0%
1	90 18.0%	1 0.2%	3 0.6%	2 0.4%	4 0.8%	90.0%	10.0%
2	0 0.0%	92 18.4%	0 0.0%	0 0.0%	0 0.0%	100%	0.0%
3	0 0.0%	0 0.0%	100 20.0%	0 0.0%	0 0.0%	100%	0.0%
4	1 0.2%	0 0.0%	1 0.2%	104 20.8%	0 0.0%	98.1%	1.9%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	102 20.4%	100%	0.0%
	98.9% 1.1%	98.9% 1.1%	96.2% 3.8%	98.1% 1.9%	96.2% 3.8%	97.6% 2.4%	
Actual	1	2	3	4	5		

Fig. 6 Confusion matrix of the classifier on the testing data set (Gaussian realizations)

Fig. 7 Energy contribution of principal components (PCs) for production data and models (Gaussian realizations)



The CNN classifier model $\mathcal{D}(\cdot)$ is trained with the labeled data set until the accuracy of both the training and validation data sets converge to a similar value, as a model validation step to prevent over-fitting.

The trained classifier is evaluated using the class-balanced testing data set to verify its robustness in extracting salient information from unseen samples. In Fig. 6, we show that the model is capable of achieving 97.6% accuracy (sum of true positive values along the matrix diagonal) on unseen data, suggesting that it has learned to identify, for each scenario, unique features in production data. We observe that scenario 1 has the highest false positives (sum of row 1 less true positive, 10.0%) where the incorrectly classified samples are perceived to be from scenario 1. This precision (90.0%) is attributed to the relatively high similarity in terms of simulated production response of realizations from the other four scenarios.

From Eq. 4, once the classifier is trained, feeding any data to the classifier in a feed-forward operation results in the vector \mathbf{g} that represents the likelihood of it belonging to different scenarios. In the second stage $\mathcal{H}(\cdot)$, we select random samples of realizations from each

scenario according to the proportion in \mathbf{g} and perform inverse mapping. This consequent data set is denoted as $(\mathbf{D}_{reduced}, \mathbf{M}_{reduced})$ and has 500 elements. In Fig. 7, significant amount of variance for this particular synthetic data set is observed in the first 7 leading components. In $\mathcal{H}(\cdot)$, the goal is to learn the non-linear inverse mapping between production data and the geologic realizations in a complex solution space as shown in Fig. 8, and to determine data support for the geologic features. Similar to $\mathcal{D}(\cdot)$, the regression model is trained with the reduced training data set until convergence and validated on a validation data set. The reference case comes from the testing set and is not included in any of these two sets.

Due to the ill-posed nature of the problem, it is impossible to achieve near zero loss when training $\mathcal{H}(\cdot)$. In Fig. 9, realizations from multiple scenarios are distinguishable in the leading principal components. Realizations from the test data set and their predicted realizations (when the corresponding simulated production data are fed to $\mathcal{H}(\cdot)$) are compared in the same orthogonal space. We can observe strong correlation between the first few leading principal components of the production

Fig. 8 Production data and models from five scenarios (Gaussian realizations) visualized in the first three principal components (PCs)

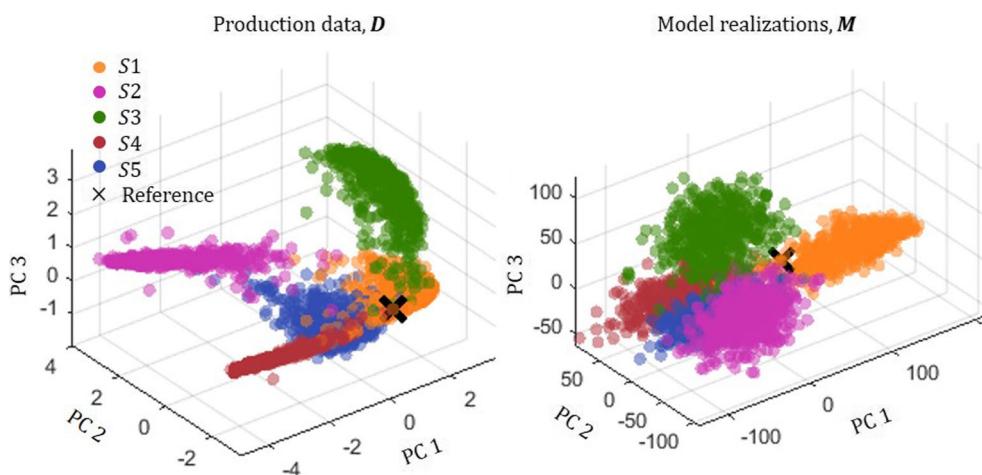
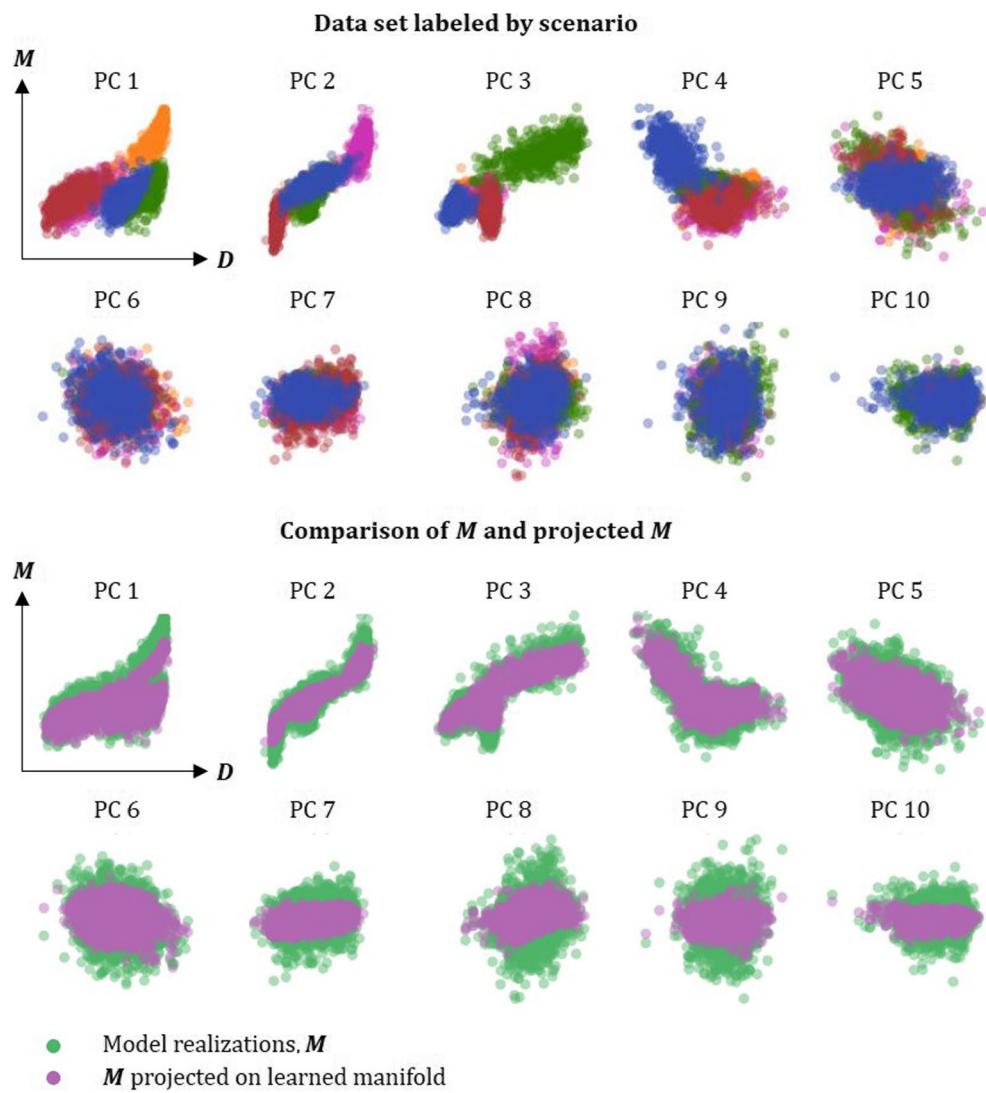


Fig. 9 (Top) Geologic scenarios are only distinguishable in the leading principal components (PCs) of production data and model realizations. (Bottom) Strong correlation exists between leading PCs of production data and model realizations



data and model realizations. This strong correlation fades when increasingly non-leading principal components are analyzed. These distinguishing features and correlation are implicitly extracted and learned by f_θ and f_ψ from the training data set.

When low-rank representation of model realizations is used, an important question is the choice of s , i.e., the number of leading elements to be included. In general, s can be determined by first analyzing the contribution of each eigenvalue to the total variance (Fig. 7 (right)), and by examining the correlation between the PCs of model realizations and production data. It is important to note that for geologic scenario identification, smaller s values will be more effective as the first few leading elements tend to have strong discriminating power. In Fig. 9 (bottom), the correlation is significant for the 7 leading PCs, implying that production data has little

sensitivity to non-leading PCs. For highly non-linear relations, where small changes in geologic features can lead to entirely different production data behavior, it is possible to have significant correlation between non-leading PCs of model realizations and leading PCs of production data.

Figure 10 illustrates the outcome of the workflow for a multi-Gaussian case. The best predicted realization is obtained after the likelihood of each scenario is considered. Without the scenario selection step, accuracy of the prediction reduces significantly and artifact from irrelevant scenarios is introduced in the prediction. The nearest neighbors (in the training data set) to the reference realization is shown to prove that the reference case has not been seen by the model in the training process. The reduced training data set used to train $\mathcal{H}(\cdot)$ honors the proportion of scenario identified by $\mathcal{D}(\cdot)$.

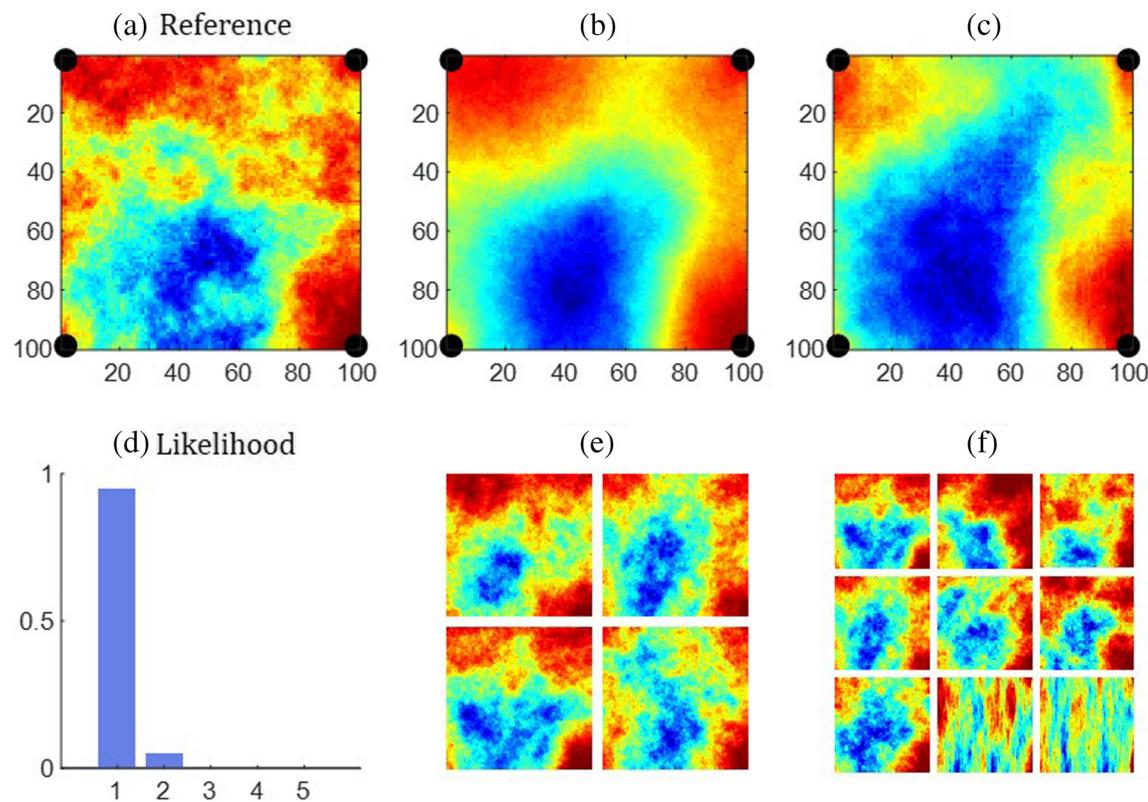


Fig. 10 (a) Reference realization as denoted in Fig. 8. (b) Predicted realization with the workflow. (c) Predicted realization with no geologic scenario selection step. (d) Likelihood of each scenario with

$\mathcal{D}(\mathbf{d}_{obs})$. (e) Nearest neighbors to the reference realization show that it does not exist in the training set. (f) Samples from $\mathbf{M}_{reduced}$

3.2 Example 2: Synthetic 2D fluvial model

In this example, we demonstrate the workflow using a non-Gaussian two-dimensional reservoir of size 1000 m \times 1000 m, which is discretized into a 100 \times 100 domain. A two-phase flow system with uncertainties in fluvial features, listed in Table 4, is also considered for this example. For each of the five scenarios ($K = 5$), training images that capture the uncertainties are generated using

object-based simulation. With these training images, 500 conditioned realizations ($N = 500$) are simulated using Multi-point Statistics (MPS) algorithm in Petrel. There are two producers and two injectors in the reservoir as seen in Fig. 11. The fluvial field is composed of binary facies, where the sand and non-sand facies are assigned (ϕ, k) pairs of (0.23, 500 mD) and (0.03, 3 mD), respectively. As in previous example, approximately one pore volume is injected into the reservoir over 6 years of simulation time

Table 4 Uncertain geologic parameters as input to 2D training image for MPS algorithm

Scenario	Channel type	Channel azimuth (°)	Channel width (meters)
1	Meandering	0	[70, 400]
2	Anastomosing	0 ± 5	[70, 400]
3	Anastomosing	45 ± 5	[70, 400]
4	Anastomosing	90 ± 5	[70, 400]
5	Anastomosing	135 ± 5	[70, 400]

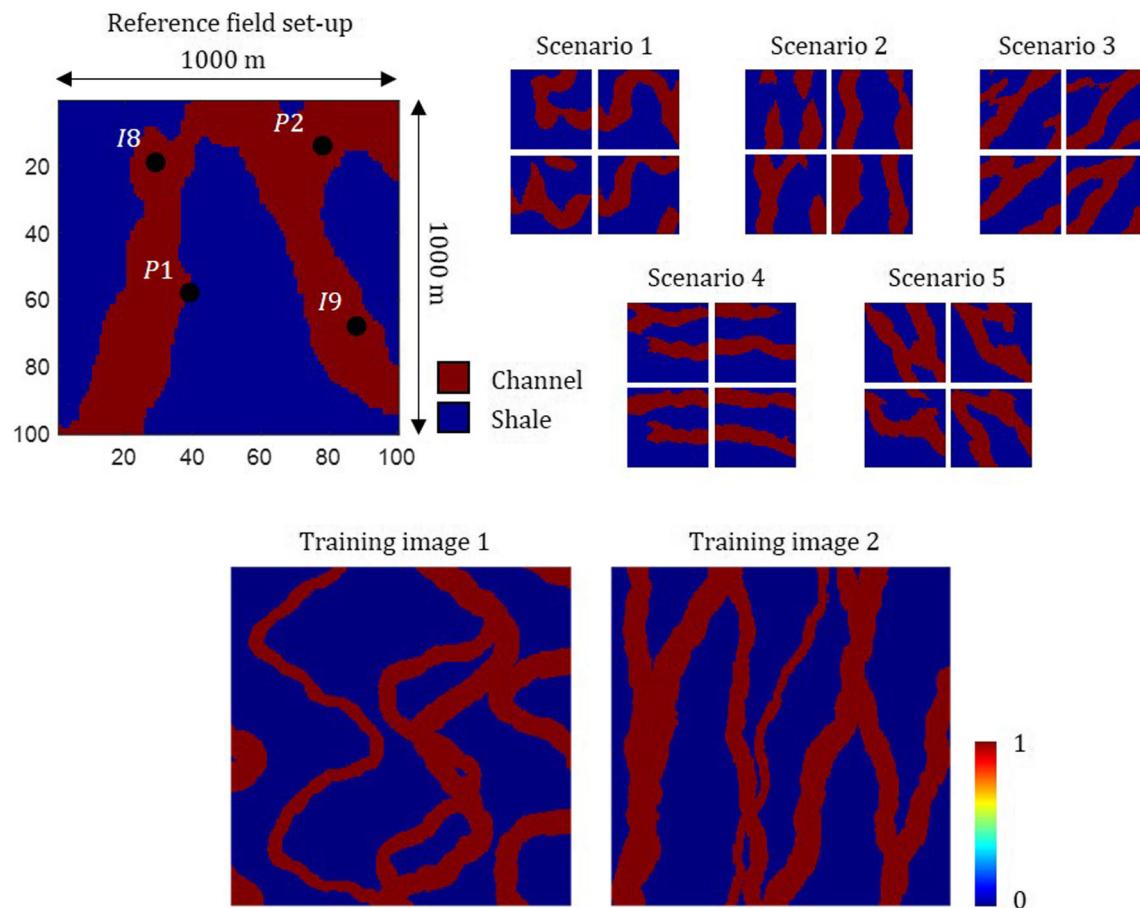


Fig. 11 (Top) Reference field set-up and samples of conditioned fluvial realizations. (Bottom) Training image 1 used for scenario 1 and training image 2 used (by rotation) for scenarios 2–5

and production data is collected every 3 months. The labeled data set is shuffled and split into training and validation data set with a ratio of 4 : 1 and the training of $\mathcal{D}(\cdot)$ and $\mathcal{H}(\cdot)$ proceeds as described earlier.

Figure 11 shows the realizations generated for each scenario, capturing the uncertainty in channel geometry (sinuous or straight), azimuth, thickness-to-width ratio, and connectivity (isolated or intersecting). From Fig. 12, the predicted realization $\hat{\mathbf{m}}$ with scenario selection step sufficiently represents the reference case. As can be seen, the finer geologic features are absent mainly because they are not supported by the production data. From the likelihood predicted by $\mathcal{D}(\cdot)$, the irrelevant scenarios are pruned and the solution space is constrained. Without $\mathcal{D}(\cdot)$, the predicted realization contains artifacts from other scenarios that when compared with historical production data in Fig. 13, degrades the quality of the production data match (compared with the case where only

relevant scenarios are included in training $\mathcal{H}(\cdot)$). Table 7 tabulates the root-mean-square error (RMSE) of data match for the examples shown in this paper.

Four additional cases (case A, B, C, and D) are shown in Fig. 14 to demonstrate the impact of confidence in the prediction of $\mathcal{D}(\cdot)$ on the solution. In general, the designed $\mathcal{D}(\cdot)$ for this specific example achieves 89.0% accuracy on the testing data set. In case A, a wrong label (1) is predicted with very high confidence which suggests that the behavior in d_{obs} is more prevalent in scenario 1 rather than its assigned scenario (label 5). As such, the inverse mapping in $\mathcal{H}(\cdot)$ is learned between geologic features and production behavior in the selected scenarios, resulting in a solution that visually appears to belong to scenario 1 and has a reasonable data match. This case occurs with very low frequency and can be attributed to the ill-posed nature of the problem.

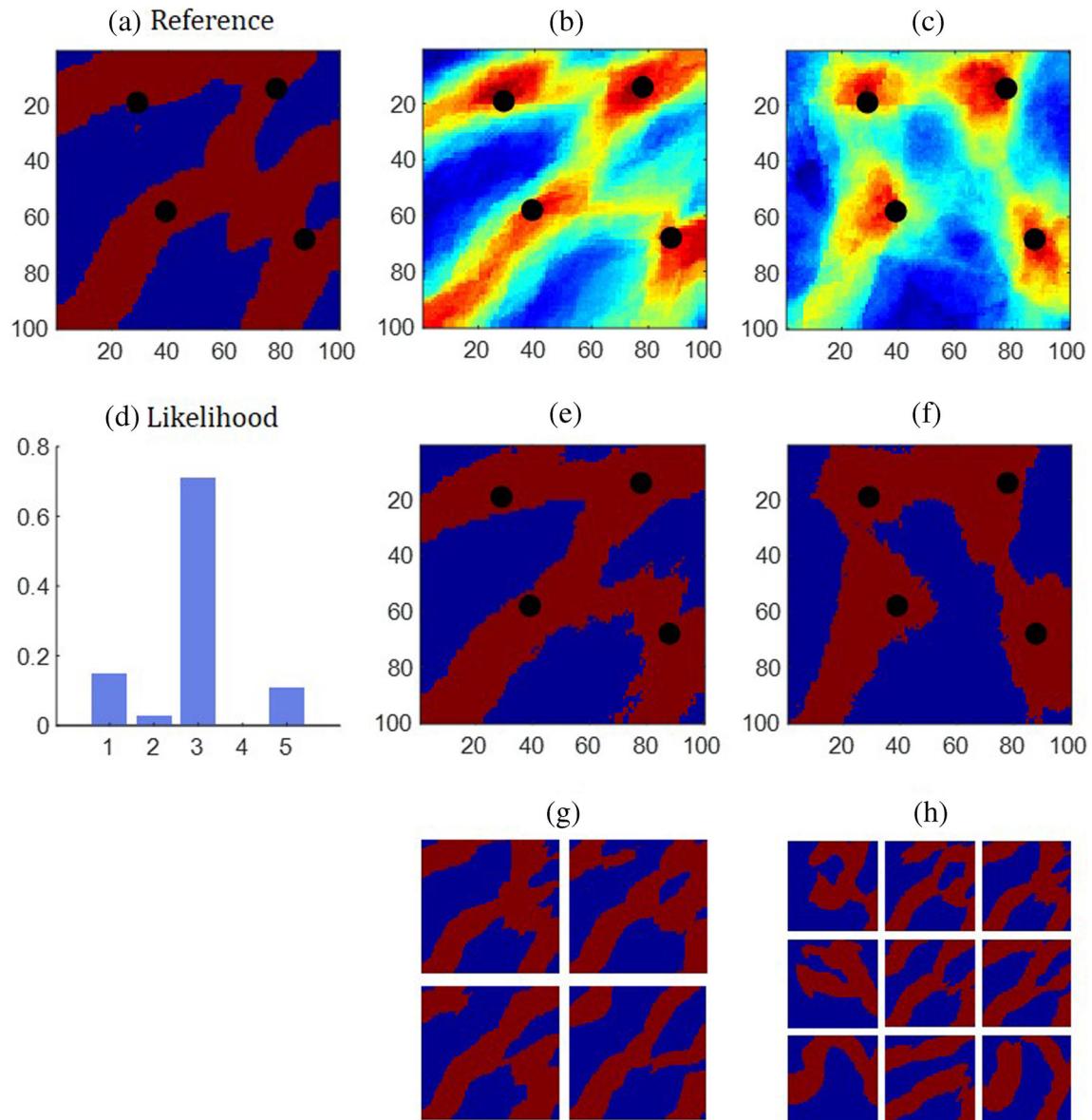


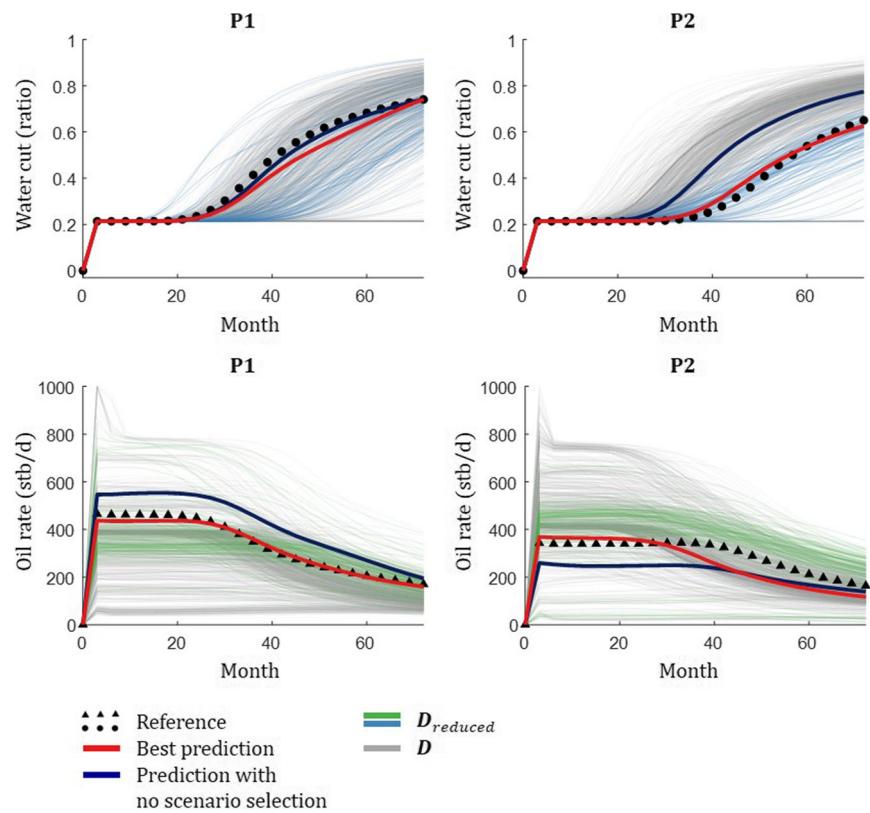
Fig. 12 (a) Reference realization. (b) Predicted realization with the workflow. (c) Predicted realization with no geologic scenario selection step. (d) Likelihood of each scenario with $\mathcal{D}(\mathbf{d}_{obs})$. (e) Discretized prediction by taking a threshold (determined by the mid-point of facies

code of channel and non-channel facies). (f) Discretized prediction. (g) Nearest neighbors to the reference realization show that it does not exist in the training set. (h) Samples from $\mathbf{M}_{reduced}$

In case B , scenario 1 is predicted instead of its correct scenario 2 albeit with very low confidence such that the likelihood is almost evenly spread across 4 different scenarios. In this situation, $\mathcal{D}(\mathbf{d}_{obs})$ suggests learning inverse mapping from 4 scenarios to increase the chance of finding geologic features that can explain the production behavior where two outcomes are possible. In the first outcome, relevant features are found and the solution preserves

geologic realism where in this example, curvilinear features are present due to the relatively higher contribution from scenario 1. In the second possible outcome, relevant features are found but when combined through $\mathcal{H}(\cdot)$, realism is not preserved and production data match is compromised. When this happens, the best course of action is to use the selected scenarios to help construct new scenarios (by consulting domain experts) that may be able to explain \mathbf{d}_{obs} .

Fig. 13 Lowest mismatch (RMSE tabulated in Table 7) is observed between reference (d_{obs}) and simulated data from the realization predicted by the workflow



In case C , the correct scenario 4 is selected with very high confidence resulting in a solution with very low data misfit. In our data set, this particular scenario has relatively low variability compared with other scenarios, hence very distinguishable production behavior. In case D , the correct scenario 3 is selected with medium confidence between two scenarios. Therefore, $\mathcal{H}(\cdot)$ learns from the two most likely scenarios resulting a solution that includes features from the two selected scenarios and reasonably low data mismatch. As is expected, features with no sensitivity to production data are not constructed.

3.3 Example 3: Synthetic 3D fluvial model

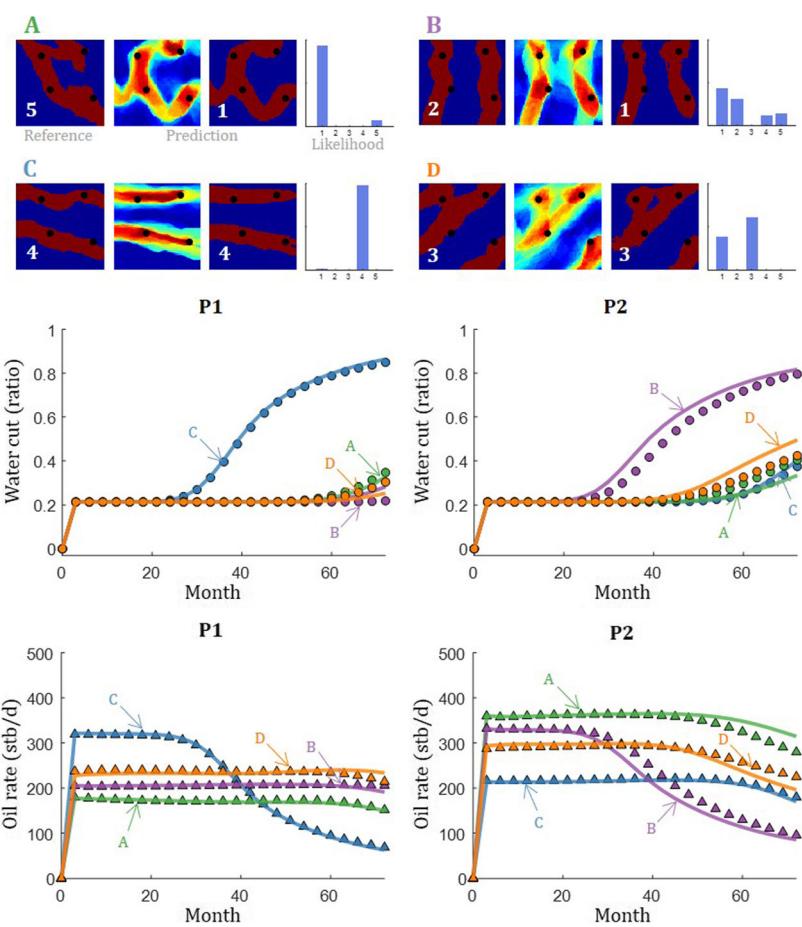
To further test the workflow on a more realistic geological setting, a three-dimensional reservoir of dimension 3000 m \times 9000 m \times 80 m that is discretized into a $40 \times 120 \times 16$ domain is considered. As before, a two-phase flow system is considered and the uncertain parameters for the geologic scenarios are listed in Table 5. For each of the four scenarios ($K = 4$), 500 realizations ($N = 500$) are generated using object-based modeling algorithm in Petrel. The realizations are conditioned to three producers and two injectors that are shown in Fig. 15. The petrophysical properties in this example are similar to those in example 2. Additional uncertainty about channel thickness within each scenario is introduced. For this example, approximately one pore

volume is injected over 8 years of simulation time, during which production data is collected every 3 months. The training of $\mathcal{D}(\cdot)$ and $\mathcal{H}(\cdot)$ proceeds as described earlier.

We observe in Fig. 16 that the best prediction is obtained when irrelevant scenarios have been removed. Figure 17 shows satisfactory production data match using the best prediction, although finer geologic features (i.e., high sinuosity meander east of I8) are smoothed out due to lack of sensitivity of production data to those features. In Fig 18, the area southwest of P3 is not drained by the producer nor entirely swept by the injector I9—resulting in almost no data sensitivity. Hence, the network predicts the mean map as predicted map of this area.

The production data match in Fig. 19 suggests that the best prediction has integrated the relevant geologic features according to the degree of sensitivities learned by $\mathcal{H}(\cdot)$. The negative impact of not selecting relevant scenarios before performing inverse mapping is observed in Fig. 16, where the area between I8 and P2 is not predicted accurately as seen in (ii). When relevant scenarios are selected according to coarse-scale temporal features in production data (extracted by training $\mathcal{D}(\cdot)$), the distinguishing information typically corresponds only to coarse-scale geologic features. Since the production data has little sensitivity to the area between I8 and P2, the predicted solution presents an averaged solution of the relevant prior realizations.

Fig. 14 Performance of the proposed workflow on four cases of 2D fluvial realizations with varying degree of confidence in the predicted scenarios of $\mathcal{D}(\cdot)$ (RMSE tabulated in Table 7)



We visualize the oil-in-place grid over simulation period where the area between I8 and P2 is partially flooded at the end suggesting low data sensitivity to this particular region. Due to this, when higher-order information is extracted using $\mathcal{D}(\cdot)$ from high sensitivity regions and used to select relevant scenarios (in this particular case, mostly from north-south fluvial scenario), the same geologic features would be observed in low-sensitivity regions as abrupt changes in geology is not expected. As mentioned earlier, in region with low data sensitivity, the prediction would be close to the mean map of the priors considered.

3.4 Example 4: Large-scale model (based on Volvo field)

The workflow is applied to a large-scale example based on Volvo field in the North sea. This three-dimensional reservoir of approximate dimension 5000 m \times 4000 m \times 80 m is discretized into a 78 \times 87 \times 15 domain. A complex fault system in the field creates varying degrees of sand juxtaposition that divides the reservoir into hydraulically separated producing regions. Figure 20 shows the field set-up (2D map view) with 10 producers and 4 water

Table 5 Uncertain geologic parameters as input to 3D object-based modeling algorithm

Scenario	Channel type	Channel azimuth ($^{\circ}$)	Thickness-width ratio (ratio)
1	Meandering	0 ± 5	0.1 ± 0.05
2	Meandering	30 ± 5	0.1 ± 0.05
3	Meandering	330 ± 5	0.1 ± 0.05
4	Braided	0 ± 15	0.2 ± 0.05

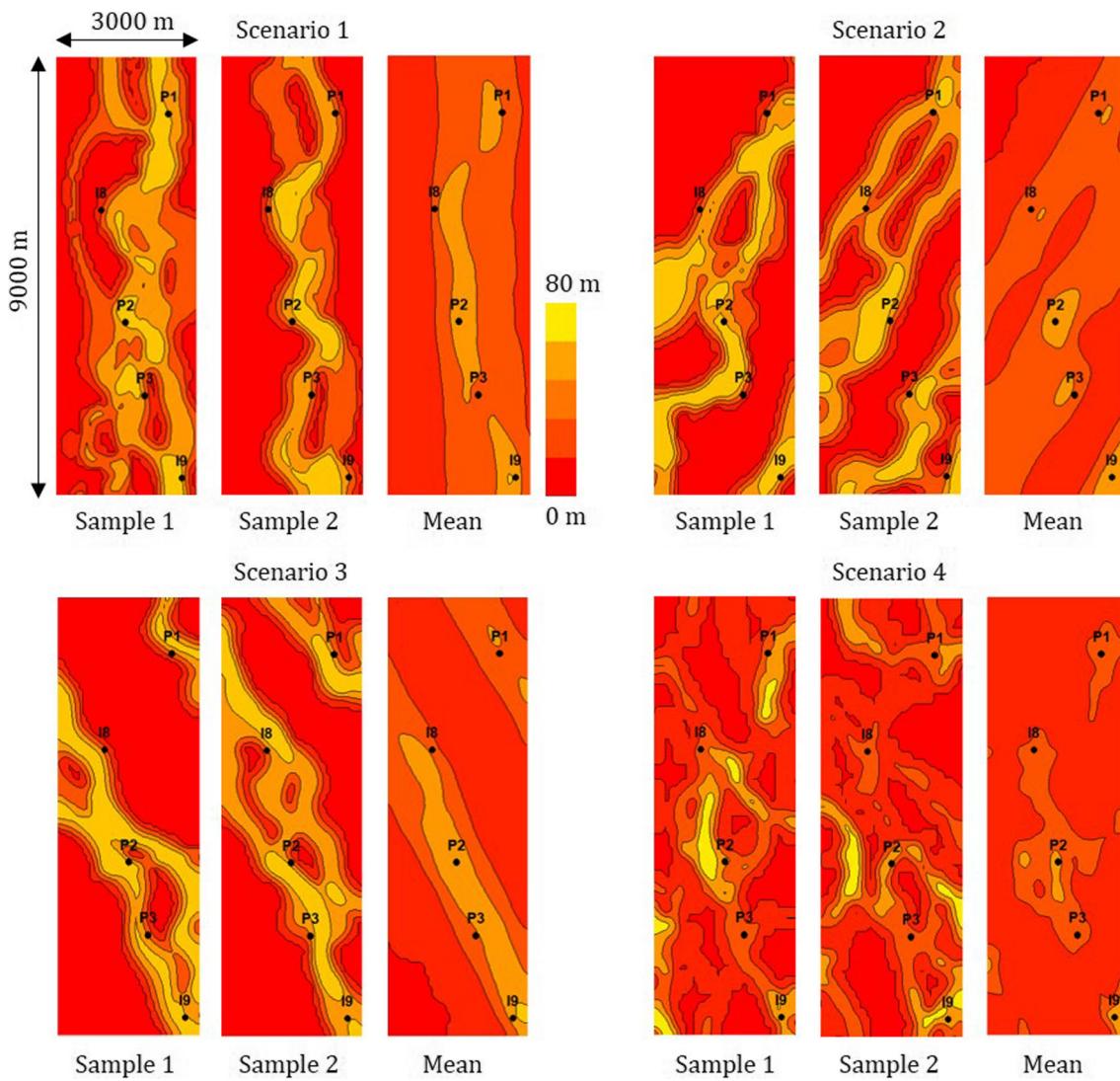


Fig. 15 Samples and mean of conditioned 3D realizations shown as isochore maps with contour interval of 16 m (measured along true vertical thickness between the top and base horizons of this synthetic field). The thickness of channel facies at each well location is varied

injectors (marked in red) that penetrate several of the fault blocks. The porosity field is populated (with Petrel) using Sequential Gaussian Simulation (SGS) algorithm conditioned to the wells (porosity logs for select wells are shown) with thicker sand accumulation in the northern area. The uncertain scenarios ($K = 3$) considered in this example are defined by variogram azimuth (Table 6) and for each scenario, 500 realizations ($N = 500$) are generated. We assume that the structural and fault framework are certain. The permeability field is calculated using a simple transform function $k = 5 + 10000000\phi^{8.7}$ to mimic the ranges seen in the actual field. The porosity values range from 0.05 to 0.35 while the permeability values range from 5 to 1085 mD.

A water-flooding system is considered for a total simulation time of 13 years, with approximately 35% oil recovery factor at the end of simulation. Using production data collected every month, $\mathcal{D}(\cdot)$ achieves a testing accuracy of 89.7% when trained and validated with $N = 500$. Our sensitivity analysis shown in Fig. 21 supports that $N = 100$ is able to yield reasonable testing accuracy of 83.0% and increasing the number of N for the scenario selection step introduces marginal improvement in classification accuracy on the testing set. Similar to the previous examples, we randomly divide the data for $\mathcal{D}(\cdot)$ in a 4:1 ratio for the training and validation data sets. The class-balanced testing data set for the sensitivity analysis (Fig. 21) is

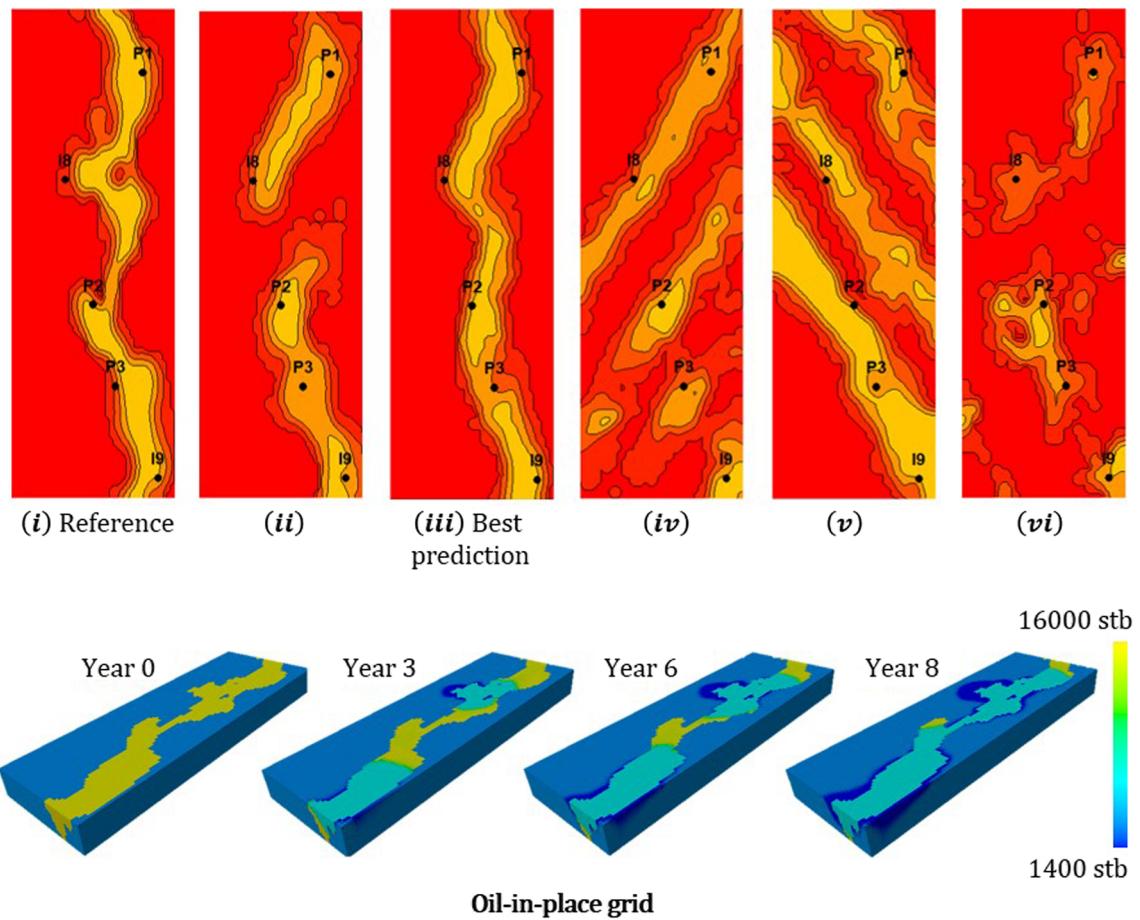


Fig. 16 (Top) (i) Reference realization. (ii) Predicted realization with no geologic scenario selection step. (iii) Predicted realization with the workflow. (iv) Prediction using models from scenario 2. (v)

Prediction using models from scenario 3. (vi) Prediction using models from scenario 4. (Bottom) Time steps of oil-in-place grid of the reference realization

kept constant (with 300 elements) for a fair comparison. With this observation, to screen unsupported scenarios, the total simulation run (computational cost) can be reduced to $N \cdot K = 300$. For the inversion step, 500 realizations are sampled from the relevant scenarios specified in \mathbf{g} for training $\mathcal{H}(\cdot)$ to provide more information on the complex manifold of inverse mapping.

In Fig. 22, a cross section (\mathbf{x} to \mathbf{x}') across key wells and a 2D map view (of the 11th layer) of the reference porosity grid (a) are compared with the predicted porosity grid (b). While spatial continuity patterns are predicted well for all layers, vertical features (fining-upwards trend) appear smooth especially in areas that are not water-flooded and away from wells. This is expected as spatial patterns can be informed by inter-well water cut and oil rate trends, whereas there is not enough sensitivity to vertical heterogeneity within the dynamic data, since all the wells have single-string completion. The upscaled well porosity logs (hard data) in the predicted realization ($\hat{\mathbf{m}}$) are also reproduced,

while the extreme values are not predicted accurately (as shown on the histograms of porosity grid).

Figure 22 c shows the initial and final oil-in-place for the reference case and predictions, where some peripheral fault blocks are filtered (defined as non-oil-bearing). The water-flooded areas are visible around the injectors, where production data has the highest sensitivity to geologic features. The northern area of the reservoir (compartment penetrated by P4) with thicker and higher porosity sand shows good prediction performance as more production from this area results in more sensitivity of production data to geologic features. For areas away from the wells with no data sensitivity, the prediction would be close to the mean values of the priors considered.

Figure 23 shows good data match between the reference production data (\mathbf{d}_{obs}) and data simulated from $\hat{\mathbf{m}}$ even when predicted spatial features are not identical to those in the reference case, which is attributed to the non-uniqueness of the inverse mapping captured by $\mathcal{H}(\cdot)$. For

Fig. 17 Comparison of production data match (RMSE tabulated in Table 7) between predictions from Fig. 16

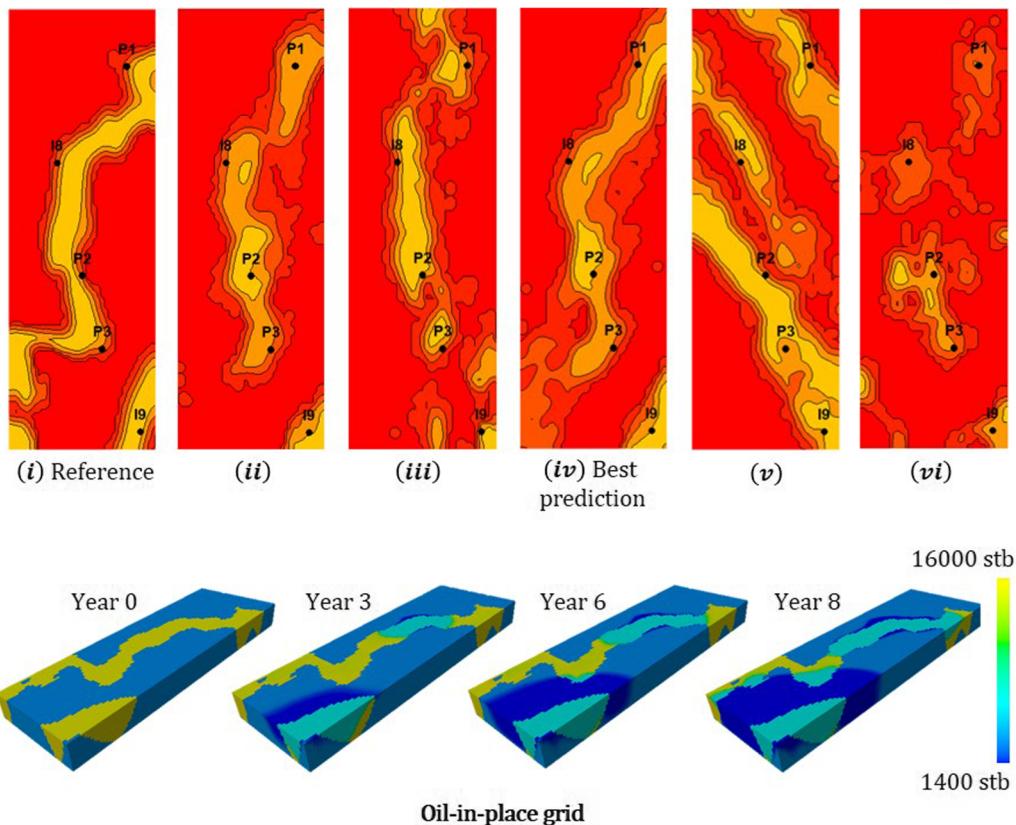
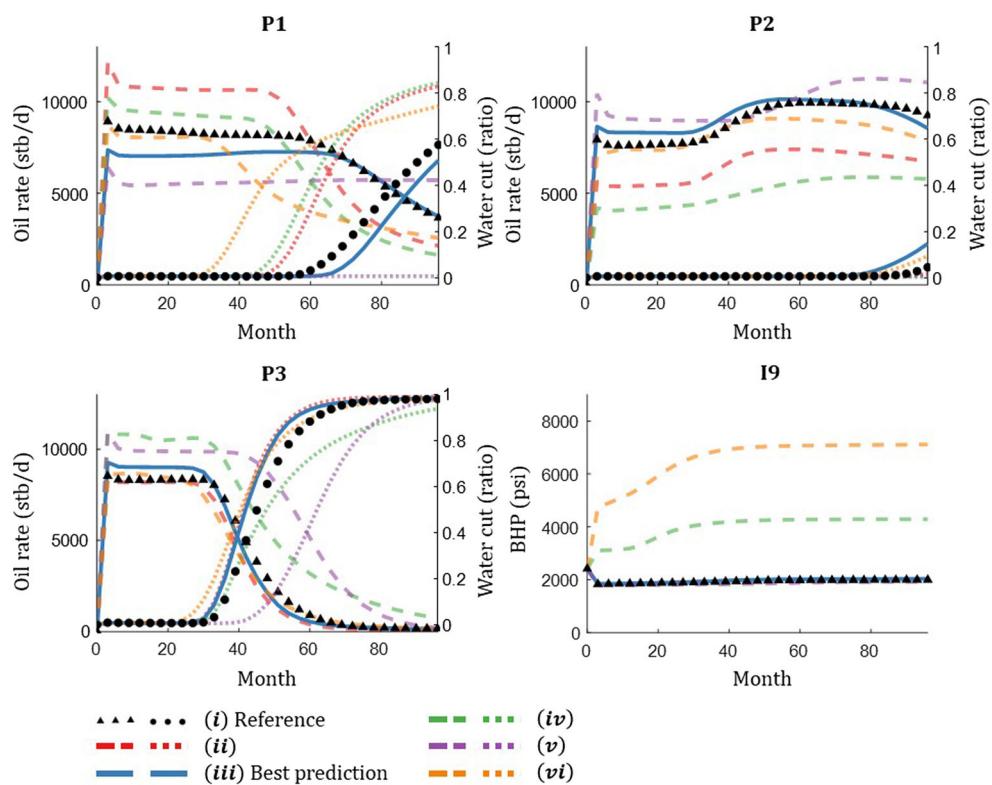


Fig. 18 (Top) (i) Reference realization. (ii) Predicted realization with no geologic scenario selection step. (iii) Predicted realization using models from scenario 1. (iv) Prediction with the workflow. (v)

Prediction using models from scenario 3. (vi) Prediction using models from scenario 4. (Bottom) Time steps of oil-in-place grid of the reference realization

Fig. 19 Comparison of production data match (RMSE tabulated in Table 7) between predictions from Fig. 18

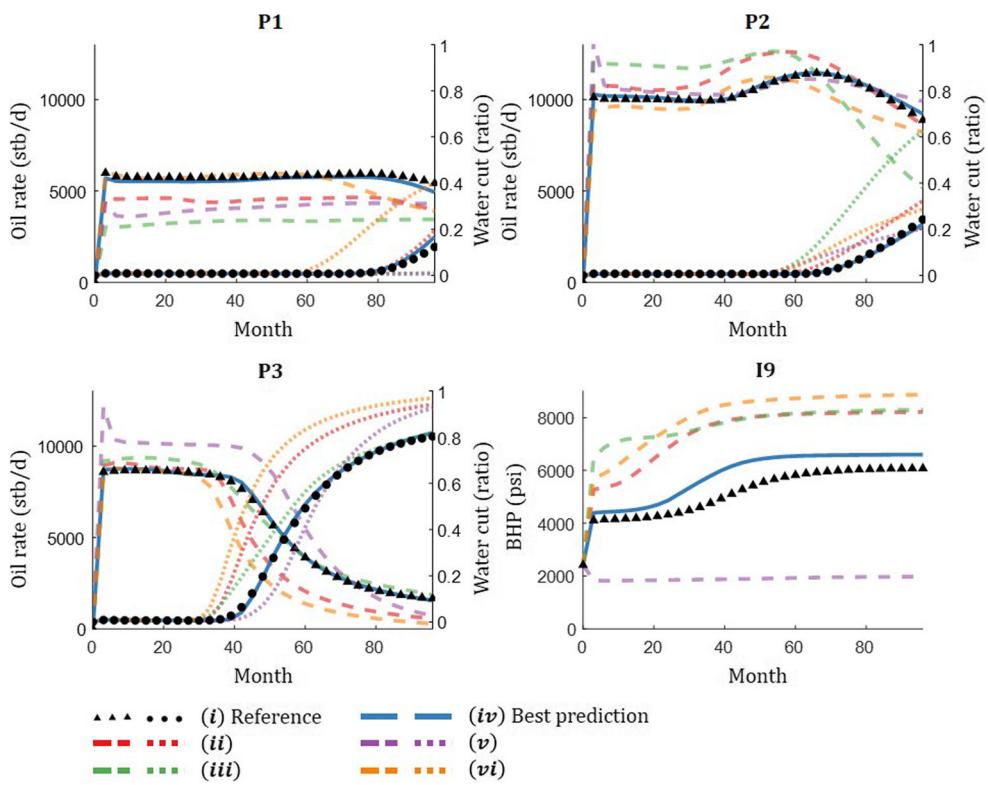


Fig. 20 Field set-up (2D map view), porosity logs (true vertical depth) for select wells, and samples of Volvo conditioned Gaussian realizations from three scenarios

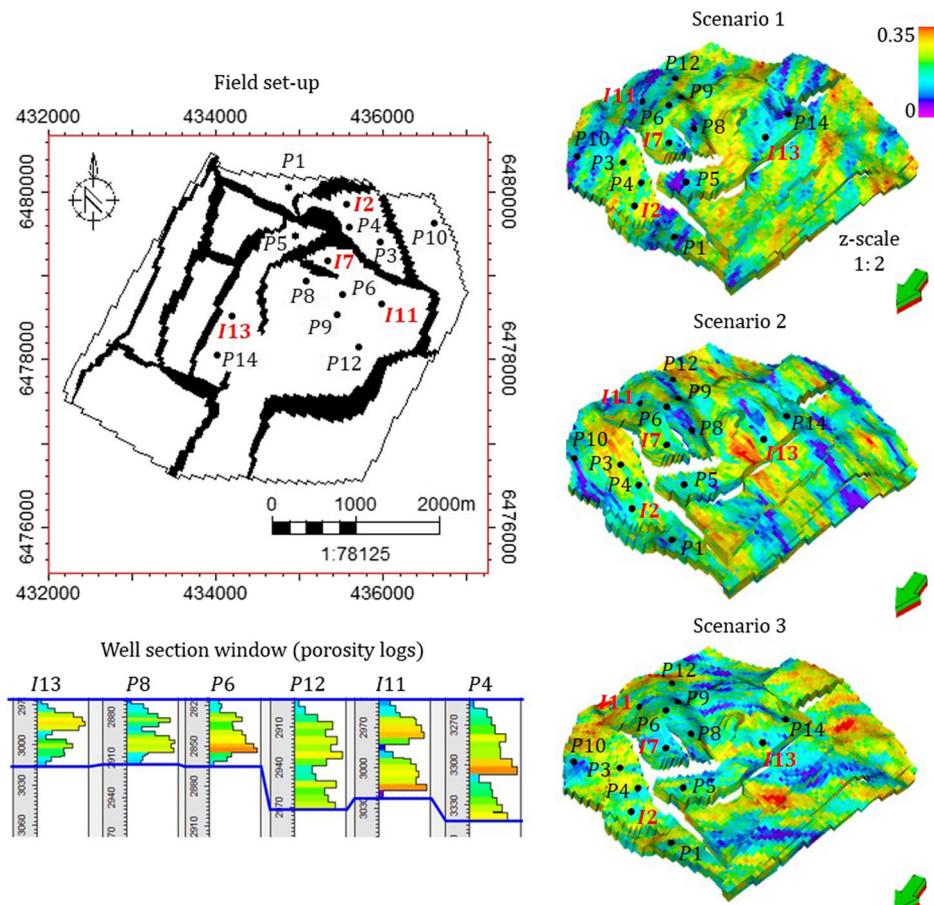


Table 6 Uncertain geologic parameters for Volvo field-like case

Scenario	Variogram azimuth (°)	Variogram (major, minor, vertical) ranges (meters)
1	0 ± 5	(1000 ± 100 , 300 ± 50 , 50)
2	120 ± 5	(1000 ± 100 , 300 ± 50 , 50)
3	240 ± 5	(1000 ± 100 , 300 ± 50 , 50)

this multi-Gaussian example, geologic realism of \hat{m} across hydraulic compartments is preserved when inverse mapping is done with production data and model realizations from relevant scenarios. The information in d_{obs} is used to identify supported scenarios and allows reliable prediction of geologic features in virgin compartments with no well penetration (Table 7).

4 Discussion and conclusions

In this paper, we demonstrate the feature learning capability of CNN as a classifier to distinguish between geologic scenarios and as a regression model to perform inverse mapping from flow response data to salient features of reservoir property distribution. Combined together, the two CNNs can be used as a two-stage workflow for geologic scenario selection and history matching, with promising outcomes. The first step of the workflow applies the CNN as a classifier by extracting salient information in production data to distinguish relevant geologic scenarios. Geologic scenario selection increases the efficiency of the

inverse mapping step by reducing the number of possible geologic scenarios. The second step uses CNN as a multi-dimensional non-linear regression model to learn the inverse mapping between production data and important features of reservoir property distribution (permeability in this work) in the reduced set from the previous step. Once the model is trained, it can be used to estimate model parameters for any set of production data if the training data is reliable. The results show that the geologic scenario selection step is an important step to remove possible artifact (due to irrelevant scenarios) from the solution. Despite no formal mechanism to ensure a good data match for the predicted models by this approach, the resulting solutions could reproduce the observed data and contained geologically consistent features.

The main computation for the developed data-driven workflow in this work is associated with generating the data sets and training the CNN, which are performed off-line. Once a trained model is constructed, it can be used in real time to map flow response data to reservoir property distribution. Unlike conventional methods that use field data to perform model calibration, through gradient-based or gradient-free methods, the developed approach in this paper does not require the field data to construct a model for inverse mapping. As such, history matching of field data with the resulting trained model is fast and straightforward, making the framework easy to deploy and implement. Therefore, the workflow can be seamlessly integrated into current practices in the industry, where it can be used as “black-box” inversion proxy model.

The presented workflow, however, has its limitations too. As in many other machine learning workflows, the off-line computation for training requires several model realizations and forward simulation runs. For the examples presented, the computational cost of training the CNN models was insignificant compared with the computational burden associated with running reservoir simulation. In general, the workflow requires a minimum number of training data to effectively represent the manifold of production data and geologic realization. Computational efficiency can be gained by incrementally increasing the number of training data and evaluating the performance of the CNN classifier for falsification. Additionally, no formal guarantees exist for the trained model to honor the field data. However, in the experiments presented in this paper, the simulated

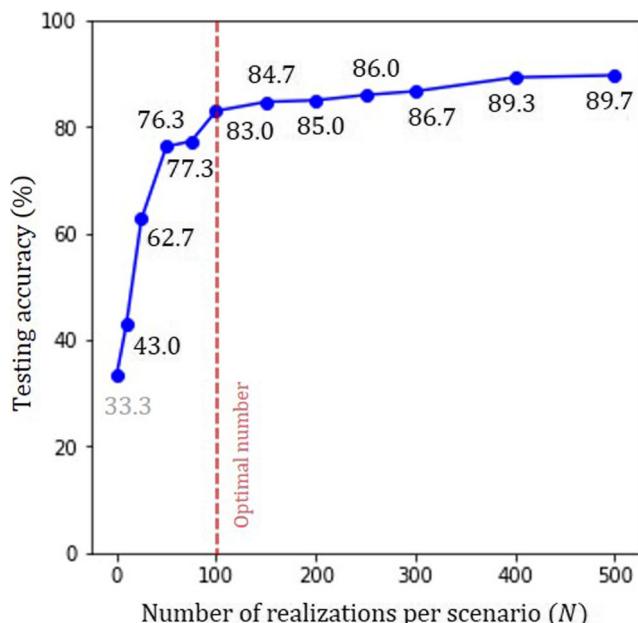
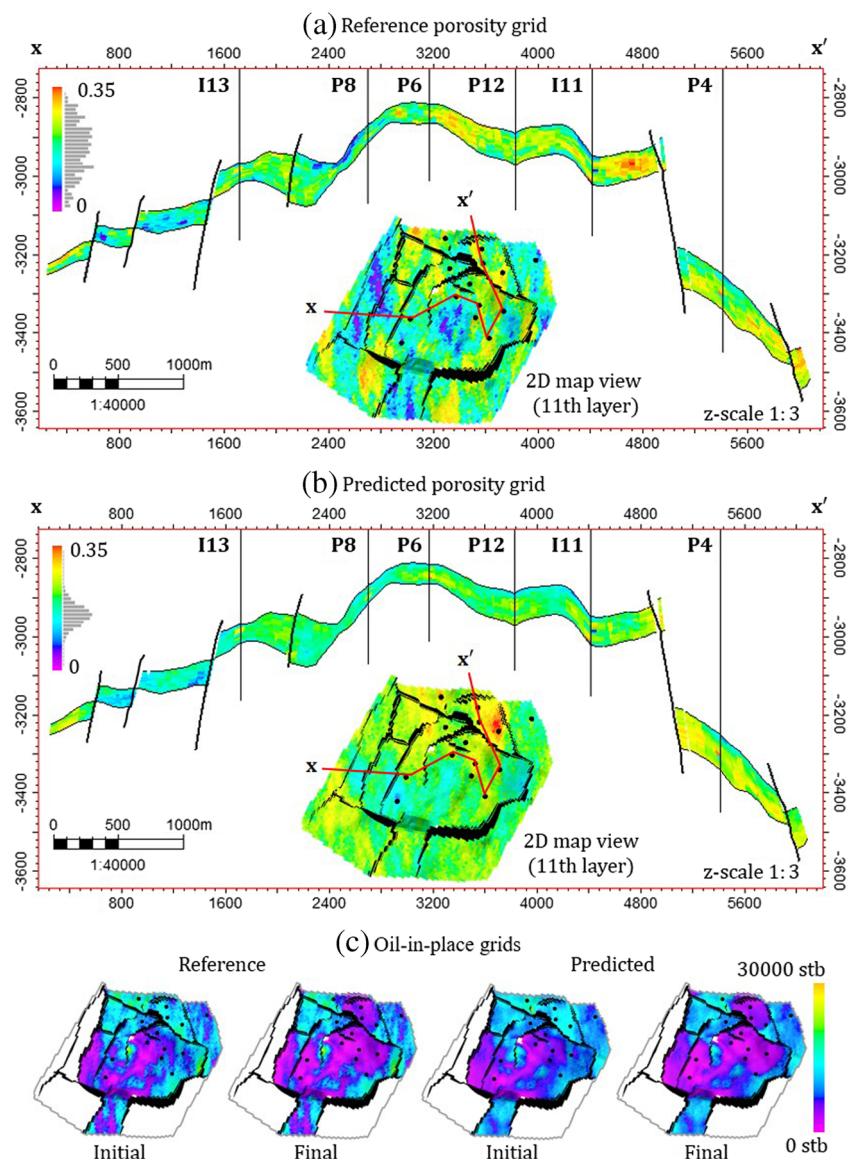


Fig. 21 Sensitivity analysis on testing accuracy as a function of number of realizations per scenario used for training and validation of $D(\cdot)$

Fig. 22 (a) Cross-section (x to x') of reference porosity grid. (b) Cross-section (x to x') of predicted porosity grid. (c) Initial and final oil-in-place grids for reference and predicted cases (map view of 11th layer)



data from the solution was reasonably close to the observed production data. Similar to other methods of automated history matching, the prediction or reconstruction ability depends on several factors such as well density, injector-producer configuration, length of production period, and the contrast in petrophysical parameters.

In conclusion, the presented workflow exploits the feature learning property of CNN to perform geologic scenario selection and history matching in a two-stage approach. In the first step, CNN is trained to learn the complex relationship between the general data trend and the geologic scenarios to eliminate scenarios that cannot be supported by dynamic flow behavior in the field. The second step uses the flow response data from a reduced set of reservoir model realizations to train a CNN and

construct an inverse mapping from the response data to salient geologic features (spatial patterns). The resulting workflow showed promising performance in a series of 2D and 3D numerical experiments including a realistic field-scale example. Furthermore, the robustness of the workflow in a more complex setting where multi-facies geologic models and other major sources of uncertainty such as structural variation in horizons and fault systems are present in the reservoir needs to be evaluated. Another important extension is related to quantification of uncertainty by developing probabilistic formulations that provide multiple calibrated models as the output of the inverse mapping. All in all, end-to-end automated geologic scenario selection and history matching efforts is an important area of investigation that requires novel formulation. Advances

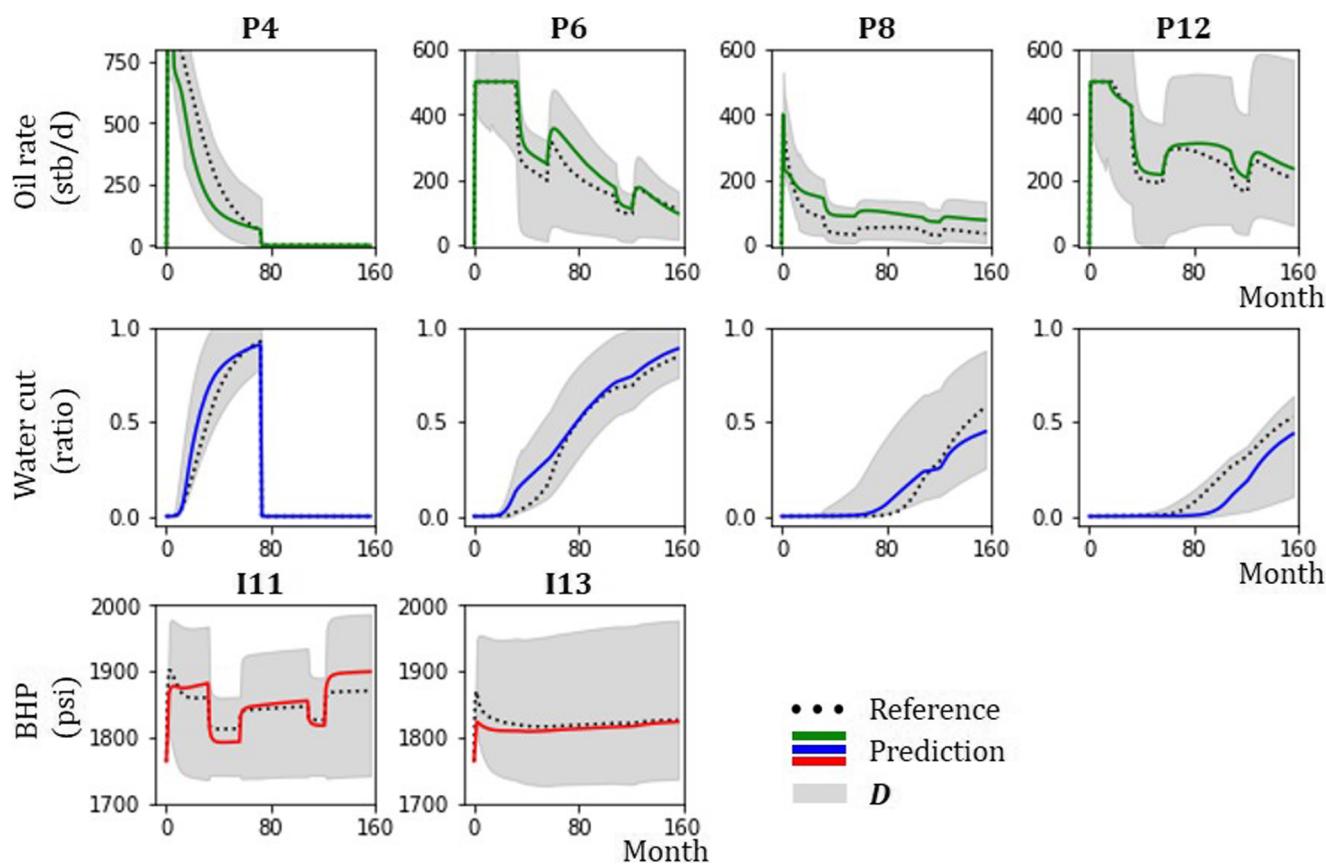


Fig. 23 Production data match (RMSE tabulated in Table 7) for select producers (oil rate and corresponding water cut) and injectors (pressure)

Table 7 Root-mean-square error (RMSE) for examples shown in this paper

Example	Case	RMSE		
		Two-stage workflow	No scenario selection	Prediction with incorrect scenario
Example 2 (Fig. 13)		0.1349	0.1463	-
Example 2 (Fig. 14)	A	0.0217	0.0429	-
	B	0.0496	0.0731	-
	C	0.0728	0.0874	-
	D	0.0564	0.0936	-
Example 3 (Fig. 17)	ii	-	0.2016	-
	iii	0.1673	-	-
	iv	-	-	0.3293
	v	-	-	0.2619
	vi	-	-	0.3829
Example 3 (Fig. 19)	ii	-	0.2039	-
	iii	-	-	0.2813
	iv	0.0568	-	-
	v	-	-	0.2219
	vi	-	-	0.3857
Example 4 (Fig. 23)		0.0924	0.1731	-

in machine learning are also introducing new tools and approaches that may prove effective in solving some of the complex problems in reservoir modeling and geosciences in general. Given the complex nature of oil and gas reservoirs, challenges in inferring the intricate interactions between production data and geologic features, and limitations in available data and computational resources, physical insight may become necessary to effectively tailor data science tools for application in subsurface flow problems.

Nomenclature CNN, convolutional neural network K , number of plausible scenarios CART, classification and regression tree N , number of model realizations per scenario \mathbf{d}_{obs} , historical/observed dynamic data PCA, principal component analysis MDS, multi-dimensional scaling NPV, net-present value $\mathcal{D}(\cdot)$, classification model $\mathcal{H}(\cdot)$, regression model \mathbf{x}^l , input to any l layer in a neural network $H^l \times W^l \times D^l$, height, width, and depth of input of any l layer in a neural network $H^{l+1} \times W^{l+1} \times D^{l+1}$, height, width, and depth of output of any l layer in a neural network b , batch size of training data i^l, j^l, d^l , index for any element in input matrix for l layer $i^{l+1}, j^{l+1}, d^{l+1}$, index for any element in output matrix for l layer $H \times W \times D^l$, height, width, and depth of each filter/kernel of any l layer f , set of filters/kernels \mathbf{a} , output/activation of a convolution layer ReLU, rectified linear unit \mathbf{h} , output of a non-linear function \mathbf{p} , output of a max-pooling (subsampling) layer \mathbf{z} , output vector of a dense (matrix-multiplication) layer C , number of scenarios (class labels) to be classified \mathbf{y}_d , one-hot vector encoding of class labels for each input for classifier loss function σ , sigmoid function θ , weights of the classifier f_θ , trainable kernels/filters of the classifier used for feature extraction S , matrix of one-hot encoding of production data labels $\mathbf{M}_{reduced}$, set of model realizations from relevant scenarios \mathbf{M} , set of model realizations from all scenarios $\mathbf{D}_{reduced}$, set of simulated production data from relevant scenarios \mathbf{D} , set of simulated production data from all scenarios \mathbf{m} , matrix representation of a model realization \mathbf{y}_h , vector representation of each model realization for regression model loss function ψ , weights of the regression model f_ψ , trainable kernels/filters of the regression model used for feature extraction \mathbf{u} , vector representation of a geologic realization φ_i , a basis function v_i , coefficient to a basis function s , number of basis functions retained for approximation n , number of elements in vector representation of a geologic realization ϕ' , matrix of basis functions retained for approximation \mathbf{v} , coefficients to matrix of basis functions retained for approximation \mathbf{z}' , a Gaussian field SGS, sequential Gaussian simulation ϕ , a porosity model k , a permeability model RMSE, root-mean-square error \mathbf{g} , likelihood of scenarios MPS, multi-point statistics $\hat{\mathbf{m}}$, predicted realization with proposed workflow

Funding information This research was supported in part by a grant from the Energi Simulation Foundation.

References

1. Araya-Polo, M., Jennings, J., Adler, A., Dahlke, T.: Deep-learning tomography. *Lead. Edge* **37**(1), 58–66 (2018)
2. Bassamzadeh, N., Ghanem, R.: Probabilistic Data-Driven Prediction of Wellbore Signatures in High-Dimensional Data Using Bayesian Networks Society of Petroleum Engineers (2018)
3. Bond, C.E., Gibbs, A.D., Shipton, Z.K., Jones, S.: What do you think this is?: “Conceptual uncertainty” in geoscience interpretation. *GSA Today* **17**, 4–10 (2007)
4. Brunetti, C., Bianchi, M., Pirot, G., Linde, N.: Hydrogeological model selection among complex spatial priors. *Water Resour. Res.* **55**(8), 6729–6753 (2019). <https://doi.org/10.1029/2019WR024840>
5. Cherpeau, N., Caumon, G., Caers, J., Lévy, B.: Method for stochastic inverse modeling of fault geometry and connectivity using flow data. *Math. Geosci.* **44**(2), 147–168 (Feb 2012)
6. Demyanov, V., Arnold, D., Rojas, T., Christie, M.: Uncertainty quantification in reservoir prediction part 2—handling uncertainty in the geological scenario. *Math. Geosci.* **51**(2), 241–264 (2019)
7. Dromgoole, P., Speers, R.: Managing uncertainty in oilfield reserves. *Middle East Well Eval. Rev.* **12**, 30–41 (1992)
8. Friedmann, F., Chawathe, A., Larue, D.K.: Assessing Uncertainty in Channelized Reservoirs Using Experimental Designs. Society of Petroleum Engineers (2001)
9. Gavalas, G.R., Shah, P.C., Seinfeld, J.H.: Reservoir History Matching by Bayesian Estimation. Society of Petroleum Engineers (1976)
10. Golmohammadi, A., Jafarpour, B.: Simultaneous geologic scenario identification and flow model calibration with group-sparsity formulations. *Adv. Water Resour.* **92**, 208–227 (2016)
11. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press, Cambridge (2016)
12. Hermans, T., Nguyen, F., Caers, J.: Uncertainty in training image-based inversion of hydraulic head data constrained to ERT data Workflow and case study. *Water Resour. Res.* **51**, 5332–5352 (2015)
13. Jacquard, P.: Permeability Distribution from Field Pressure Data. Society of Petroleum Engineers (1965)
14. Jahns, H.O.: A Rapid Method for Obtaining a Two-Dimensional Reservoir Description from Well Pressure Response Data. Society of Petroleum Engineers (1966)
15. Jeong, H., Sun, A.Y., Lee, J., Min, B.: A learning-based data-driven forecast approach for predicting future reservoir performance. *Adv. Water Resour.* **118**, 95–109 (2018). ISSN 0309-1708. <https://doi.org/10.1016/j.advwatres.2018.05.015>
16. Jiang, A., Jafarpour, B.: Combining Regularized Convolutional Neural Networks with Production Data Integration for Geologic Scenario Selection. SPE Annual Technical Conference and Exhibition (2019)
17. Jiang, R., Stern, D., Halsey, T., Manzocchi, T.: Scenario discovery workflow for robust petroleum reservoir development under uncertainty. *International Journal for Uncertainty Quantification*, **6** (2016)
18. Khaninezhad, M.R., Jafarpour, B.: Prior model identification during subsurface flow data integration with adaptive sparse representation techniques. *Comput. Geosci.* **18**(1), 3–16 (2014)
19. Khaninezhad, M.R., Jafarpour, B., Li, L.: Sparse geologic dictionaries for subsurface flow model calibration: part I. Inversion formulation. *Adv. Water Resour.* **39**, 106–121 (2012)
20. Khodabakhshi, M., Jafarpour, B.: A Bayesian mixture-modeling approach for flow-conditioned multiple-point statistical facies simulation from uncertain training images. *Water Resour. Res.* **49**(1), 328–342 (2013)
21. Kim, S., Min, B., Kwon, S., Chu, M.: History Matching of a Channelized Reservoir Using a Serial Denoising Autoencoder Integrated with ES-MDA. *Geofluids*. <https://doi.org/10.1155/2019/3280961> (2019)
22. Kingma, D.P., Adam, J.B.a.: A Method for Stochastic Optimization. arXiv.org (2014)
23. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **25**, 1097–1105 (2012)

24. Laloy, E., Héroult, R., Jacques, D., Linde, N.: Training-image based geostatistical inversion using a spatial generative adversarial neural network. *Water Resour. Res.* **54**(1), 381–406 (2018). <https://doi.org/10.1002/2017WR022148>
25. LeCun, Y., Bengio, Y.: The Handbook of Brain Theory and Neural Networks. MIT Press, Cambridge (1998)
26. LeCun, Y., Bengio, Y., Hinton, G.E.: Deep learning. *Nature* **521**, 436–444 (2015)
27. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. In: Proceedings of the IEEE, pp. 2278–2324 (1998)
28. Leite, E.P., Vidal, A.C.: 3D Porosity prediction from seismic inversion and neural networks. *Comput. Geosci.* **37**(8), 1174–1180 (2011)
29. Maschio, C., Schiozer, D.S.: Bayesian history matching using artificial neural network and Markov chain Monte Carlo. *J. Pet. Sci. Eng.* **123**, 62–71 (2014)
30. Min, B., Park, C., Kang, J., Park, H., Jang, I.S.: Optimal well placement based on artificial neural network incorporating the productivity potential. *Energy Sources Part A Recov. Utilization Enviro. Effect.* **33**, 1726–1738,07 (2011). <https://doi.org/10.1080/15567030903468569>
31. Mishkin, D., Sergievskiy, N., Matas, J.: Systematic evaluation of CNN advances on the ImageNet. *coRR* (2016)
32. Mohaghegh, S.D.: Reservoir simulation and modeling based on artificial intelligence and data mining (AI&DM). *J. Natur. Gas Sci. Eng.* **3**(6), 697–705 (2011)
33. Nair, V., Hinton, G.E.: Rectified Linear Units Improve Restricted Boltzmann Machines. In: Proceedings of the 27Th International Conference on on Machine Learning, pp. 807–814, Omnipress (2010)
34. Narayanan, H., Mitter, S.: Sample complexity of testing the manifold hypothesis. *Adv. Neural Inf. Process. Syst.* **23**, 1786–1794 (2010)
35. Nelson, P.H.: Permeability-porosity Relationships In Sedimentary Rocks. *The Log Analyst - Society of Petrophysicists and Well-Log Analysts*, pp. 38–62 (1994)
36. Oliver, D.S., Chen, Y.: Recent progress on reservoir history matching: a review. *Comput. Geosci.* **15**(1), 185–221 (2011)
37. Park, H., Scheidt, C., Fenwick, D., Boucher, A., Caers, J.: History matching and uncertainty quantification of facies models with multiple geological interpretations. *Comput. Geosci.* **17**(4), 609–621 (2013)
38. Pirot, G., Huber, E., Irving, J., Linde, N.: Reduction of conceptual model uncertainty using ground-penetrating radar profiles: field-demonstration for a braided-river aquifer, vol. 571. ISSN 0022-1694. note=10.1016/j.jhydrol.2019.01.047, (2019)
39. Pirot, G., Renard, P., Huber, E., Straubhaar, J., Huggenberger, P.: Influence of conceptual model uncertainty on contaminant transport forecasting in braided river aquifers, vol. 531. ISSN 0022-1694. <https://doi.org/10.1016/j.jhydrol.2015.07.036> (2015)
40. Posamentier, H.W., Allen, G.P.: Overview: Siliciclastic Sequence Stratigraphy, Concepts and Applications. *SEPM Society for Sedimentary Geology* (1999)
41. Randle, C.H., Bond, C.E., Lark, R.M., Monaghan, A.A.: Uncertainty in geological interpretations effectiveness of expert elicitations. *Geosphere* **15**(1), 108–118 (2019)
42. Roth, G., Tarantola, A.: Neural networks and inversion of seismic data. *J. Geophys. Res. Solid Earth* **99**(B4), 6753–6768 (1994)
43. Scheidt, C., Jeong, C., Mukerji, T., Caers, J.: Probabilistic falsification of prior geologic uncertainty with seismic amplitude data application to a turbidite reservoir case. *Geophysics* **80**, 89–100 (2015)
44. Scholz, M., Fraunholz, M., J. Selbig.: Nonlinear Principal Component Neural Network Models and Applications. Principal Manifolds for Data Visualization and Dimension Reduction. Springer, Berlin (2008)
45. Schuetter, J., Mishra, S., Zhong, M., LaFollette, R.: A Data-Analytics Tutorial, Building Predictive Models for Oil Production in an Unconventional Shale Reservoir. *Society of Petroleum Engineers* (2018)
46. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **15**, 1929–1958 (2014)
47. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the Inception Architecture for Computer Vision. *CoRR* (2015)
48. Tarantola, A.: Popper, Bayes and the inverse problem. *Nat. Phys.* **2**, 492–494 (2006)
49. Van Wagoner, J.C., Posamentier, H.W., Mitchum, R.M., Vail, P.R., Sarg, J.F., Loutit, T.S., Hardenbol, J.: An Overview of the Fundamentals of Sequence Stratigraphy and Key Definitions. In: Sea-Level Changes: an Integrated Approach. *SEPM Society for Sedimentary Geology* (1988)

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.