Delft University of Technology

Conditioning of deep-learning surrogate models to image data with application to reservoir characterization

Xiao, Cong; Leeuwenburgh, Olwijn; Lin, Hai-Xiang; Heemink, Arnold

DOI
10.1016/j.knosys.2021.106956

Publication date
2021

Document Version
Final published version

Published in
Knowledge-Based Systems

Citation (APA)
Xiao, C., Leeuwenburgh, O., Lin, H.-X., & Heemink, A. (2021). Conditioning of deep-learning surrogate models to image data with application to reservoir characterization. *Knowledge-Based Systems*, *220*, 1-25. Article 106956. https://doi.org/10.1016/j.knosys.2021.106956

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.
For technical reasons the number of authors shown on this cover page is limited to a maximum of 10.

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Conditioning of deep-learning surrogate models to image data with application to reservoir characterization

Cong Xiao [a,*], Olwijn Leeuwenburgh [b,c], Hai-Xiang Lin [a], Arnold Heemink [a]

[a] Delft Institute of Applied Mathematics, Delft University of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands
[b] Civil Engineering and Geosciences, Delft University of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands
[c] TNO Princetonlaan 6 PO Box 80015 3508 TA Utrecht, The Netherlands

ARTICLE INFO

Article history:
Received 14 July 2020
Received in revised form 3 February 2021
Accepted 13 March 2021
Available online 16 March 2021

Keywords:
Reservoir simulation
Data parameterization
Deep convolutional neural network
Image segmentation
Data assimilation

ABSTRACT

Imaging-type monitoring techniques are used in monitoring dynamic processes in many domains, including medicine, engineering, and geophysics. This paper aims to propose an efficient workflow for application of such data for the conditioning of simulation models. Such applications are very common in e.g. the geosciences, where large-scale simulation models and measured data are used to monitor the state of e.g. energy and water systems, predict their future behavior and optimize actions to achieve desired behavior of the system. In order to reduce the high computational cost and complexity of data assimilation workflows for high-dimensional parameter estimation, a residual-in-residual dense block extension of the U-Net convolutional network architecture is proposed, to predict time-evolving features in high-dimensional grids. The network is trained using high-fidelity model simulations. We present two examples of application of the trained network as a surrogate within an iterative ensemble-based workflow to estimate the static parameters of geological reservoirs based on binary-type image data, which represent fluid facies as obtained from time-lapse seismic surveys. The differences between binary images are parameterized in terms of distances between the fluid-facies boundaries, or fronts. We discuss the impact of the choice of network architecture, loss function, and number of training samples on the accuracy of results and on overall computational cost. From comparisons with conventional workflows based entirely on high-fidelity simulation models, we conclude that the proposed surrogate-supported hybrid workflow is able to deliver results with an accuracy equal to or better than the conventional workflow, and at significantly lower cost. Cost reductions are shown to increase with the number of samples of the uncertain parameter fields. The hybrid workflow is generic and should be applicable in addressing inverse problems in many geophysical applications as well as other engineering domains.

© 2021 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

1. Introduction

Imaging-type monitoring techniques are relevant for monitoring of dynamic processes in many application domains, and include for example X-ray, computed tomography (CT) and magnetic resonance imaging (MRI) techniques for medical imaging [1], satellite remote sensing for earth observation [2,3], and seismic and electromagnetic imaging of the subsurface [4]. Applications in the earth observation domain include the prediction of spreading of air pollution [5,6] and e.g. typhoon tracks [7]. Geophysical applications include the monitoring of CO_2 storage in aquifers [8] and the displacement of fluids in hydrocarbon reservoirs [9]. Imaging techniques deliver pixel-wise information in 2D or 3D and may be used to identify static features or anomalies or changes over time. We are especially interested in the application of such data for the conditioning of simulation models. Such applications are very common in the geosciences, where large-scale simulation models and measured data are used to monitor the state of e.g. energy and water systems, predict their future behavior and optimize actions to achieve desired behavior of the system.

A challenge with the use of imaging techniques for these purposes is that they tend to deliver very large number of data points that may have complex relationships to underlying (poorly-known) model parameters. Therefore compromises are often needed towards the data assimilation methods that are used to integrate the data into models, or towards the description of the data and the associated measurement noise. We intend to use state-of-the-art data assimilation (in fact, parameter estimation) methods that are able to deliver a full uncertainty characterization, especially Iterative Ensemble Smoothers (IES) [10]. Such

[*] Corresponding author.
E-mail address: c.xiao@tudelft.nl (C. Xiao).

https://doi.org/10.1016/j.knosys.2021.106956
0950-7051/© 2021 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

methods characterize uncertainty in the model by a large ensemble of model realizations, where each model realization is defined by a different set of randomly sampled values for uncertain model parameters.

Two challenges are commonly recognized in the application of such data assimilation methods to imaging data: (1) the use of large data sets can lead to artificial collapse of the ensemble; (2) given the computational expense of simulating each model, the large number of uncertain grid-based parameters, and the complexity in data-parameter relationships, many iterations of the IES with a large ensemble may be required, resulting in huge over-all computational costs.

Several approaches have been attempted to deliver reduced representations of large data sets, including coarse representations [11], wavelet decomposition [12], and nonlinear reduction methods based on machine learning techniques [13]. In many cases, including the earth observation and geophysical examples mentioned earlier, the purpose of the monitoring is the identification of changes over time, which can often be characterized by the displacement of a contour value in the image. Leeuwenburgh and Arts and Zhang and Leeuwenburgh [14,15] proposed a parameterization of monitoring data for such situations in terms of distances between corresponding contours (or iso-surfaces in 3D) in the simulated and measured images respectively, and showed that the resulting reduction in the number of data can help avoid ensemble collapse. (See Trani et al. [16,17] and [18] for related approaches.) In its most basic form, the contours separate the domain into regions belonging to one of two possible classes, effectively resulting in a binary image.

Here we address the second challenge, namely the high computational demand imposed by iterative parameter estimation workflows involving imaging data. In introducing the methodology we will focus on an application of time-lapse seismic monitoring for reservoir model parameter estimation, also referred to as seismic history matching in the field of reservoir engineering [19]. Seismic data are obtained as waves that are registered in grid-based distributions of sensor locations, after being first emitted into the ground at source locations on the surface and subsequently reflected at so-called impedance contrasts in the subsurface, typically reflecting spatial changes in rock properties or fluid content. When a seismic survey is repeated at a later time, the differences between the imaging data sets can often be interpreted as changes in the distribution of different types of fluids. Examples include the displacement of water by CO_2 [8] and the displacement of oil by water or gas [9]. Given that direct access to the reservoirs, which are often found at depths of a few kilometers, is possible only at locations where wells have been drilled into the reservoir, this time-lapse seismic data can be the main source of information about changes in the system.

In order to reduce the computation cost of seismic history matching reservoir models, surrogate (proxy) modeling methods such as upscaled models [20] and reduced-order models [21] have been pursued. Disadvantages of these approaches are the loss of information at high spatial resolutions and non-linearity respectively. Another approach is the use of machine learning surrogates, where especially Artificial Neural Networks (ANN) have recently started receiving renewed interest. This growing interest is related to the appearance of modern architectures that support deep networks with enhanced capability of relating large numbers of inputs and outputs. Several recent studies have explored the use of Deep Neural Network (DNN) surrogates for prediction of single-phase [22,23], two-phase [24–26], and multi-phase [27] subsurface flow dynamics. As far as we are aware, however, such approaches have not been successful yet in the context of grid-based parameter estimation based on large-volume imaging data. We will therefore propose a new surrogate modeling methodology based on machine learning or more specifically deep learning approaches that aims to deliver high-quality parameter characterization at a significantly decreased computational cost. This is motivated by rapid recent advancements in the application of deep neural networks to simulation of dynamic systems, and image processing [28], and wide availability of high-performance processing units (GPU's) and deep-learning frameworks (e.g. Tensorflow [29], PyTorch [30]).

The successful applications of deep neural networks to many application domains have been reported. In the community of subsurface flow simulations, Jin et al. proposed a new deep-convolutional auto-encoder structure with a successful application in reservoir simulation problem [31]. Tripathy et al. proposed a fully-connected deep neural network to approximate the flow dynamic of a single phase subsurface water flow in porous media [22]. Zhong et al. proposed a generative surrogate models for the dynamic plume prediction of CO_2 capture and storage problem [26]. This type of surrogate model was built on a generative adversarial network (GAN). In addition to the subsurface flow discipline, the hybridization workflow of neural network has also been presented in image reconstruction and classification problem. For example, Dawid and Gautam reported hybrid solutions by training different types of neural networks with heuristic validation mechanism in the problem of image reconstruction [32]. Dawid et al. presented an effective way to approach image classification problems based on splitting the images into smaller parts and classifying them into vectors that can be identified as features using a convolutional neural network (CNN) [33]. Ruttgers et al. improved the prediction of a typhoon track using a generative adversarial network and satellite imaginary [7]. Time series of satellite images of typhoons are used to train the neural network, which then can be employed to predict the movements of typhoon with past satellite images as inputs.

Motivated by the successful applications of hybridization workflow using neural network methods in a variety of research domains, the aim of this work is to propose a hybrid deep learning-based workflow to characterize reservoir heterogeneities. Specifically, a cheap-to-run surrogate model is constructed to efficiently predict imaging-type seismic data. Representation of fluid-fronts position using binarized images is simply equivalent to an image segmentation problem in the field of computer vision and image processing [34–36]. Deep learning as a promising approach has shown potentials to effectively address this kind of problem. This paper explores the potential of deep neural network to predict the spatially discontinuous fluid fronts based on the concept of image segmentation. It has been extensively acknowledged that a deeper network may approximate the predictions with higher complexity, but at the cost of difficulty in training [37,38]. Inspired by the successful applications of image super-resolution problems [39,40], the state-of-the-art residual-in-residual dense block (RRDB) acts as an effective means to train a deeper neural network. On the basis of conventional U-Net [41] which has shown prominent advantage and applicability for object segmentation task, we adopt an improved U-Net architecture via multiple RRDB structures.

The remainder of the paper is structured as follows: The definition of seismic history matching problem is provided in Section 2. Section 3 describes a hybrid deep-learning workflow for reservoir heterogeneity characterization. In Section 4, a systematic accuracy assessment of the proposed deep-learning segmentation model for predicting binarized fluid-fronts is presented. This section also discusses and evaluates an application of proposed hybrid workflow to characterize a synthetic 2D non-Gaussian facies model and a benchmark 3D reservoir model [42]. Finally, Section 5 highlights our contribution and points out some potential works.

2. Problem description

We address the problem of computationally efficient estimation of a large number of spatially varying and uncertain gridblock parameters in simulation models. The relationship between directly or indirectly observable dynamic (i.e., time-varying) variables and static model parameters can generally be described by a nonlinear operator $\mathbf{h}^n$: $R^{N_d} \to R^{N_\mathbf{m}}$ as follows

$$\mathbf{y}^n = \mathbf{h}^n(\mathbf{m}, t^n), \quad n = 1, \ldots, N_t. \tag{1}$$

where $\mathbf{y}^n$ is the vector of observable data, which we will treat as images, $\mathbf{m}$ denotes the vector of spatially varying parameters, $N_\mathbf{m}$ is the total number of gridblocks, t^n is the time at the nth timestep where the measurements are taken, N_d is the number of measurements taken at each timestep. In the process of Bayesian inference, ensemble-based data assimilation methods [43] enable us to assess the parameter uncertainty by sampling multiple solutions from the posterior distribution. Each of these samples is an acceptable posterior realization, which honors available measurements. These posterior realizations can be generated via minimizing the perturbed objective functions $J^i(\mathbf{m})$

$$\begin{aligned} J^i(\mathbf{m}) = & \frac{1}{2}(\mathbf{m} - \mathbf{m}_b^i)^T {\mathbf{C_m}}^{-1}(\mathbf{m} - \mathbf{m}_b^i) \\ & + \frac{1}{2}\sum_{n=1}^{N_t}[\mathbf{d}_{obs}^{n,i} - \mathbf{h}^n(\mathbf{m}, t^n)]^T {\mathbf{R}_n}^{-1}[\mathbf{d}_{obs}^{n,i} - \mathbf{h}^n(\mathbf{m}, t^n)]. \end{aligned} \tag{2}$$

where the uncertain parameters and measurements are assumed to satisfy Gaussian distributions. Specifically, $\mathbf{m}_b^i \sim N(\mathbf{m}_b, \mathbf{C_m})$ and $\mathbf{d}_{obs}^{n,i} \sim N(\mathbf{d}_{obs}^n, \mathbf{R}_n)$ for i = 1, …, N_e where N_e represents the total number of solutions that are computed.

In many cases it may not be feasible to perform this minimization exactly because it requires access to the derivatives of the operator $\mathbf{h}^n$ with respect to the model parameters $\mathbf{m}$, which may not be available, and because it requires numerous computationally costly simulations of the numerical model. This limitation has stimulated research into efficient approximate methods, for example for parameter estimation (also referred to as history matching) in subsurface reservoir engineering applications. Surrogate modeling is currently identified as one of the most promising means to improve the efficiency of parameter estimation procedures. In the following we will discuss a hybrid workflow for ensemble-based model parameter estimation through integration of deep-learning surrogates.

3. Hybrid deep-learning workflow

This section describes a hybrid deep-learning workflow for model parameter estimation, including a variant of the iterative ensemble smoother (ES-MDA), and an image-oriented distance parameterization which is used to extract informative features from the simulated and measured images. Subsequently, we describe a time-conditioning residual-in-residual dense U-Net (cRRDB-U-Net) to capture both spatial and temporal features of the model. In addition, a two-stage training strategy through alternatively minimizing the regression loss and segmentation loss is used to improve the prediction of spatially discontinuous binarized images.

3.1. Ensemble smoother with multiple data assimilation

We use the Ensemble Smoother with Multiple Data Assimilation (ES-MDA) for solving the parameter estimation problem [10]. ES-MDA is an ensemble-based iterative minimization method for solving parameter estimation problems in a Bayesian setting. Given N_e randomly sampled a prior model parameter realization vectors $\mathbf{m}_i^0$, a data vector $\mathbf{d}_{obs}$ (in our case an image), with associated error covariance matrix $\mathbf{R}_n$. The ES-MDA update (analysis) step at iteration steps k=1, …, $\mathbf{N}_a$ for each realization i can be written follows:

$$\mathbf{m}_i^k = \mathbf{m}_i^{k-1} + \mathbf{K}^k[\mathbf{d}_{obs} + \mathbf{e}_i^k - \mathbf{h}(\mathbf{m}_i^{k-1})], \quad i = 1, \ldots, N_e. \tag{3}$$

At each analysis step, the measurement error covariance $\mathbf{R}_n$ is inflated to $\epsilon_k \mathbf{R}_n$, where ϵ_k are user-chosen inflation coefficients satisfying the constraint that $\sum_{k=1}^{N_a} \frac{1}{\epsilon_k} = 1$. Here we use $\epsilon_k = \mathbf{N}_a$. $\mathbf{e}_i^k$ are random observation errors sampled from the Gaussian distribution $N(\mathbf{0}, \epsilon_k \mathbf{R})$. Here we omit the time index for a simplification. That is, the measurement operator $\mathbf{h}(\mathbf{m})$ is a concatenation of $\mathbf{h}^n(\mathbf{m}, t^n)$ at all N_t timesteps, that is, $\mathbf{h}(\mathbf{m})$ = $[\mathbf{h}^1(\mathbf{m}, t^1), \cdots, \mathbf{h}^{N_t}(\mathbf{m}, t^{N_t})]$ (see Eq. (1)). The Kalman gain $\mathbf{K}^k$ is defined as

$$\mathbf{K}^k = \mathbf{C}_\mathbf{md}^{k-1}(\mathbf{C}_\mathbf{dd}^{k-1} + \epsilon_k \mathbf{R})^{-1}. \tag{4}$$

where $\mathbf{C_{md}}$ denotes the cross-covariance of parameters and predicted measurements and $\mathbf{C_{dd}}$ is the covariance of predicted measurements, which are computed from the ensemble simulation results $\mathbf{m}_i^k$, $\mathbf{g}(\mathbf{m}_i^k)$, $i = 1, \ldots N_e$.

In summary, the ES-MDA procedure is as follows:

1. Set the iteration number $\mathbf{N}_a$ and initialize the inflation coefficients ϵ_k for k=1, …, $\mathbf{N}_a$.
2. For $k = 1, \ldots, \mathbf{N}_a$:
   - Run an ensemble of forward models with parameters $\mathbf{m}_i^k$ from initial time resulting in $\mathbf{g}(\mathbf{m}_i^k)$
   - Sample the observation errors using $\mathbf{e}_i^k \sim N(\mathbf{0}, \epsilon_k \mathbf{R})$ for each ensemble member
   - Update each ensemble member using Eqs. (3) and (4)

As indicated in Eq. (4), the implementation of ES-MDA typically requires the inversion of an $N_{obs} \times N_{obs}$ matrix $\mathbf{C} = \mathbf{C}_\mathbf{dd}^{k-1} + \epsilon_k \mathbf{R}$ (where $N_{obs} = N_d \times N_t$ represents the total number of measurements), which is computed using truncated singular value decomposition (TSVD) [10]. The overall computational cost is proportional to the cost of simulating the model $N_e \times N_a$ times. For large N_m and N_{obs} and highly nonlinear functions $\mathbf{g}(\mathbf{m})$, both N_e and N_a may need to be large in order to obtain results of sufficient accuracy, leading to a prohibitive computational costs. We will consider the possibility of reducing the simulation cost by replacing simulations of the high-fidelity model by simulation of a surrogate.

3.2. Image-oriented distance parameterization

We will consider the situation in which the relevant information contained by the data can be captured by contour (or iso-surfaces), which segments the image into binary categories (for similar applications in e.g. computer vision, see e.g. [34–36]). In order to characterize and minimize the differences between binary images originating from different sources (from high-fidelity model simulations and surrogate model simulations, and from surrogate model simulations and measured data respectively) we must adopt a similarity measure (or metric). An example is the Euclidean distance, which is the L_2 norm of point-wised discrepancy of two images. In our work, we employ a feature-oriented distance parameterization to characterize the similarity that was designed for application to seismic history matching of subsurface reservoirs [14,15]. In that application the contours represent the position of the saturation front between a displacing and a displaced fluid, which is more reliable and informative than the information contained in the amplitude of individual grid cell values. The dimension of the resulting data space (e.g. the

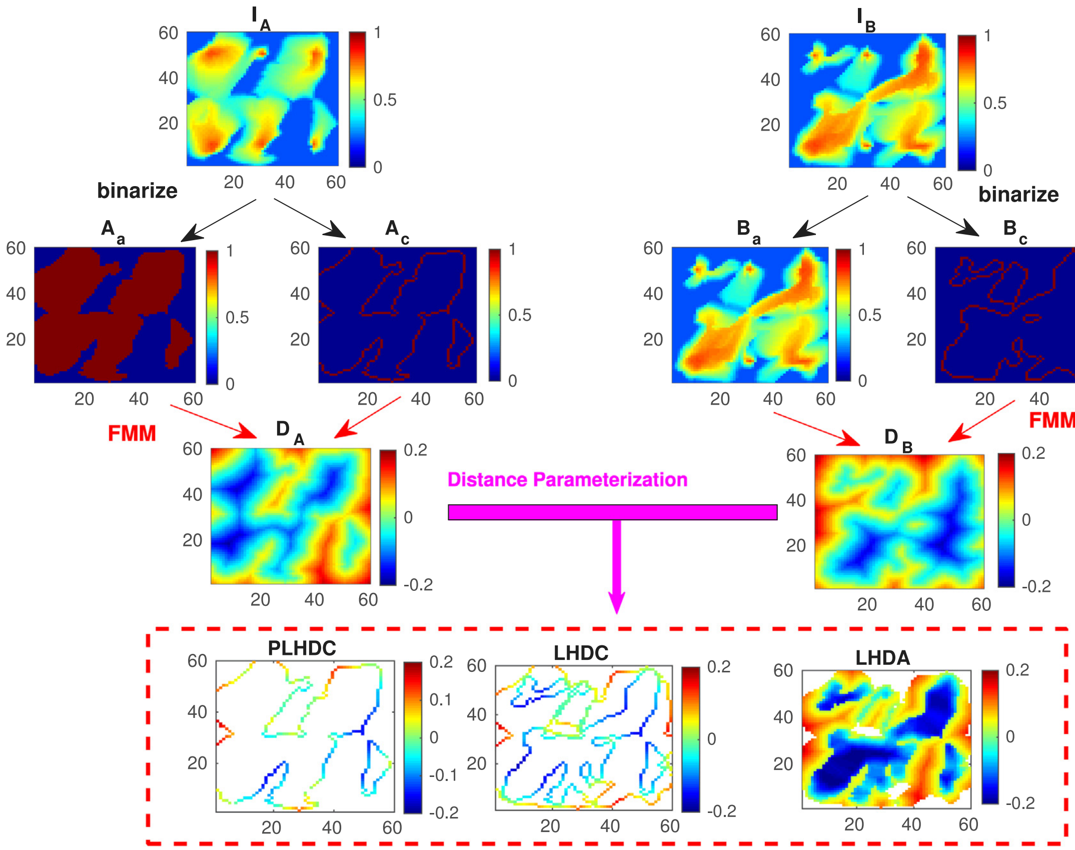


Fig. 1. Illustrative diagram of the *PLHDC*, *LHDC* and *LHDA* schemes given the two images $\mathbf{I}_A$ and $\mathbf{I}_B$ corresponding to the continuous saturation distribution. The first three rows show the original continuous saturation images, the corresponding binarized contour and area maps, and the derived distance maps respectively. The last row shows a spatial representation of the similarity metric using the *PLHDC*, *LHDC* and *LHDA* schemes. Only the colored pixels enter the difference vector.

number of points on fronts) is typically much lower than that of the original data space which is equal (or proportional) to the number of grid cells. In the following the essential elements of the parameterization are described in more detail.

In Fig. 1, scalar values 0 and 1 are used to define binarized images obtained by pre-processing underlying images $\mathbf{I}_A$ and $\mathbf{I}_B$ (as has been shown in the first row of Fig. 1). Contours defining the boundaries between the two categories contained in the images are shown as well. The similarity between the two images is characterized by a map computing from the local Hausdorff distance (*LHD*) [44]. In the literature, three parameterization approaches developed in [15] are as follows

$$\begin{aligned}
&PLHDC(\mathbf{A}_c, \mathbf{B}_c) = \mathbf{A}_c \circ \mathbf{D}_B, \\
&LHDC(\mathbf{A}_c, \mathbf{B}_c) = \mathbf{A}_c \circ \mathbf{D}_B + \mathbf{B}_c \circ \mathbf{D}_A, \\
&LHDA(\mathbf{A}_a, \mathbf{B}_a) = \mathbf{A}_a \circ \mathbf{D}_B + \mathbf{B}_a \circ \mathbf{D}_A. \qquad (5)
\end{aligned}$$

where $\mathbf{D}_A$ and $\mathbf{D}_B$ represent distance maps for shapes A and B respectively, which are computed here using a fast marching method (FMM) [15]. the subscripts c and a denote that the shapes in the images are represented by contours (boundaries or fronts) or areas (for example, a flooded area) respectively. In essence, these metrics quantify the similarity of two images with two directed distance maps in complementary directions, i.e. $\mathbf{A}_c \circ \mathbf{D}_B$ (distance from B to A) and $\mathbf{B}_c \circ \mathbf{D}_A$ (distance from A to B). *PLHDC* is the partial *LHDC* measuring the distance only from the simulated fronts to the "observed" fronts. Because nonzero distance values exist only on the "observed" fronts, the number of data is reduced from the number of grid points on the whole image to the number of grid points on the "observed" fronts. Furthermore, the binary character of the image representation is transformed into continuous data (distances) that can be handled by the data assimilation methodology. These three distance parameterization approaches each have their own advantages and disadvantages. *PLHDC* leads to the strongest reductions in the number of data, but not does capture the information in both images as well as LHDC [15]. *LHDA* provides the most complete description of similarity and differences but does not reduce the number of data. The "colored" regions in Fig. 1 identify the spatial locations of image pixels that enter the difference calculation. Pixels that are colored white are not used. Thus, the data dimension is largely reduced, especially for PLHDC and LHDC schemes. Therefore, in the remainder of this paper *LHDC* will be used. In the ES-MDA procedure, the measurement innovation (the vector containing the differences between predicted and measured data) $[\mathbf{d}_{obs}+\mathbf{e}_i^k-\mathbf{g}(\mathbf{m}_i^{k-1})]$ defined in Eq. (3), can now be replaced by the image dissimilarity, i.e., *LHDC*, which is a vector of values that jointly measure the differences between the simulated and measured images.

In the next section we will describe an efficient deep-learning segmentation model for predicting binarized images as accurately as possible.

3.3. Conditional residual-in-residual dense block U-Net

This section introduces the procedures of using a deep neural network to perform predictions of spatially discontinuous shape features. The overall neural network architecture and its subsections are illustrated in Fig. 2–Fig. 5. The section discussing the preparation of the training dataset introduces a pre-processing step to convert continuous maps into binarized images. In the network training section, we define the training losses for both

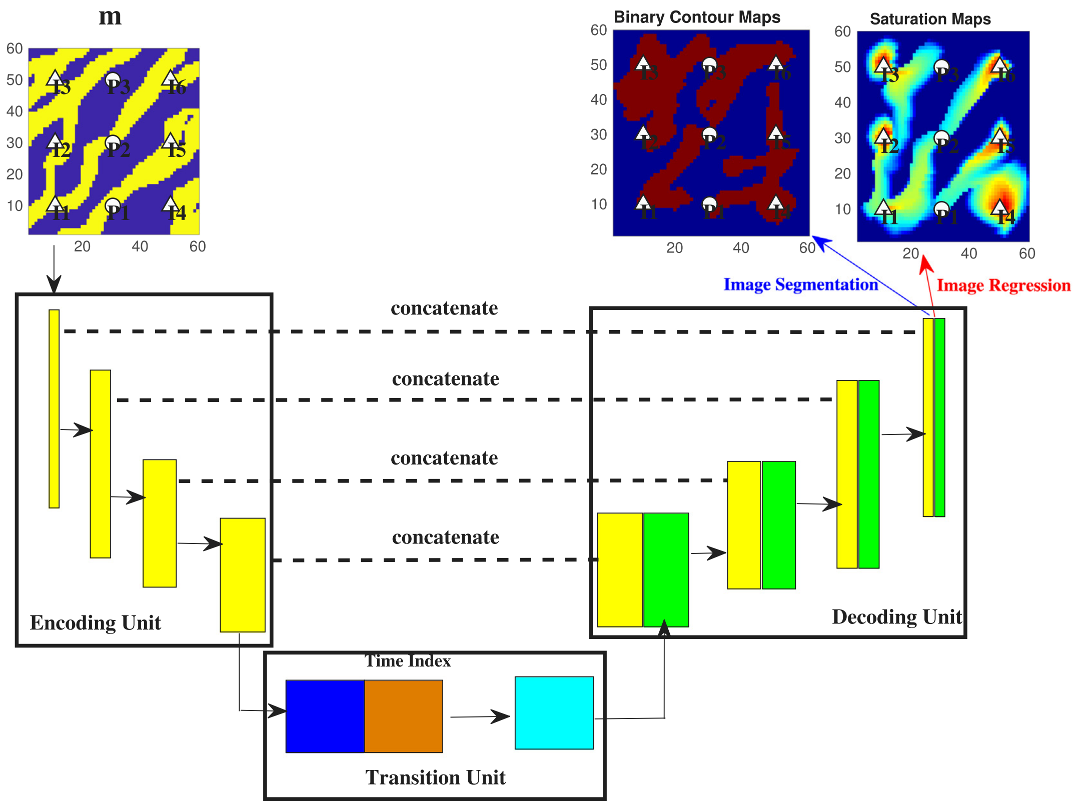


Fig. 2. Illustrative diagram of the cRRDB-U-Net architecture. The proposed cRRDB-U-Net is composed of encoding unit, transition unit and decoding unit. The multi-scale features extracted from the encoding unit are concatenated to the corresponding decoding unit for predicting the final output.

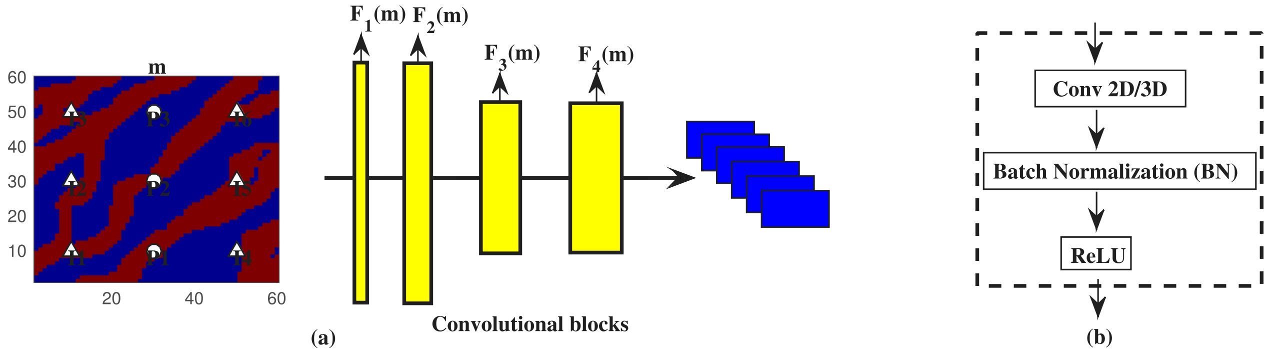


Fig. 3. Schematic illustration of the encoding unit for the cRRDB-U-Net architecture. This unit consists of four convolutional blocks. The encoding unit accepts the logarithmic permeability image as an input. The generated multi-scale features $\mathbf{F}_k(\mathbf{m}) \in R^{N_{w,k}\times N_{h,k}\times N_{d,k}}$ (k = 1, 2, 3, 4), are sequentially fed to the decoding unit. The yellow rectangles represent the "down-sampling" operation. To halve the size of the feature maps, we use convolutional layers with stride of 2, which has been shown in Table 1.

image regression and image segmentation. A two-stage training strategy consisting of alternating minimization of the regression loss and the segmentation loss suggested in the literature is applied to better approximate the discontinuous shapes in the images.

3.3.1. Neural network architecture

In this work we consider the cross-domain image segmentation problem of predicting spatially discontinuous binarized images representing changes in dynamic systems. Some recent studies has investigated the potential of using DNN surrogates to replace high-fidelity model simulations. For example, Jin et al. (2019) proposed a DNN surrogate model with autoregressive structure for approximating time-varying reservoir dynamics [31]. Tang et al. (2019) developed a deep convolutional recurrent neural network architecture, specifically a combination of auto-encoder and a convolutional long-short-term memory recurrent network (convLSTM) [45].

Our proposed hybrid workflow shares some similarities with the one proposed by Tang et al. (2019) where also a deep-learning based history matching problem was pursued. Tang et al. (2019) developed a surrogate for temporal prediction of spatially continuous pressure and saturation snapshots for channelized oil reservoir models. The spatial pressure and saturation predictions were the basis for predictions of well data such as fluid rate and bottom-hole pressure, which were used in a history matching workflow aimed at characterizing the channelized reservoir system. In this paper we will demonstrate how a similar workflow could be used for parameter estimation based on (binary) imaging-type data.

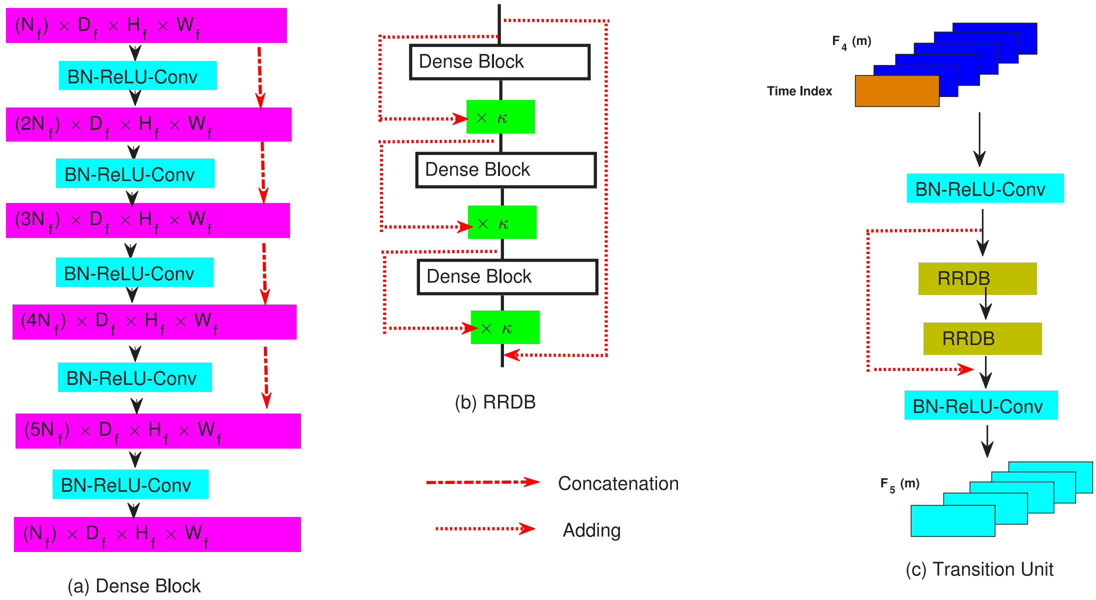


Fig. 4. Schematic illustration of the transition unit for the cRRDB-U-Net network. The time feature as an additional channel is feed to this unit as well to mimic the flow dynamic.

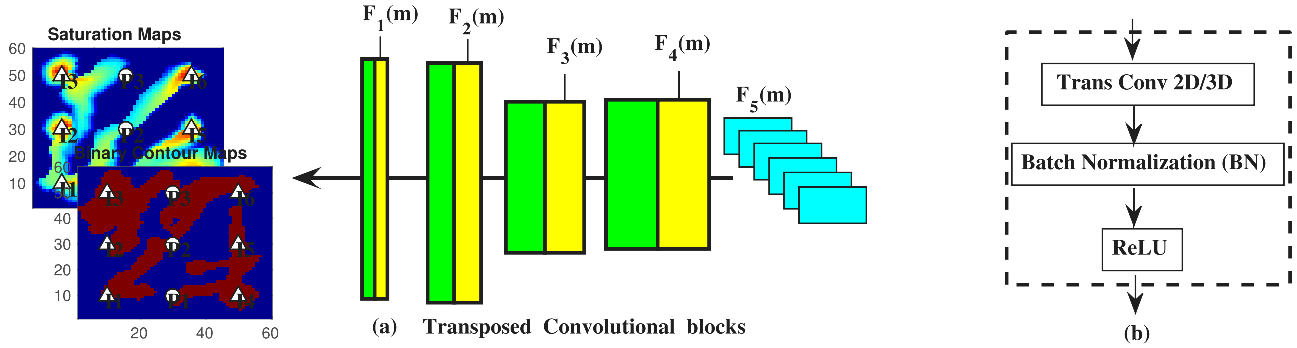


Fig. 5. Schematic illustration of the decoding unit in the cRRDB-U-Net architecture. This unit consists of four transposed convolutional blocks corresponding to the encoding unit. Decoding unit accepts the extracted multi-scale features $\mathbf{F}_k(\mathbf{m})$ $(k = 1, 2, 3, 4)$ to produce the target maps. The green rectangles represent the "up-sampling" operation. The size of the feature maps can be doubled by using transposed convolutional layers, which has been shown in Table 1.

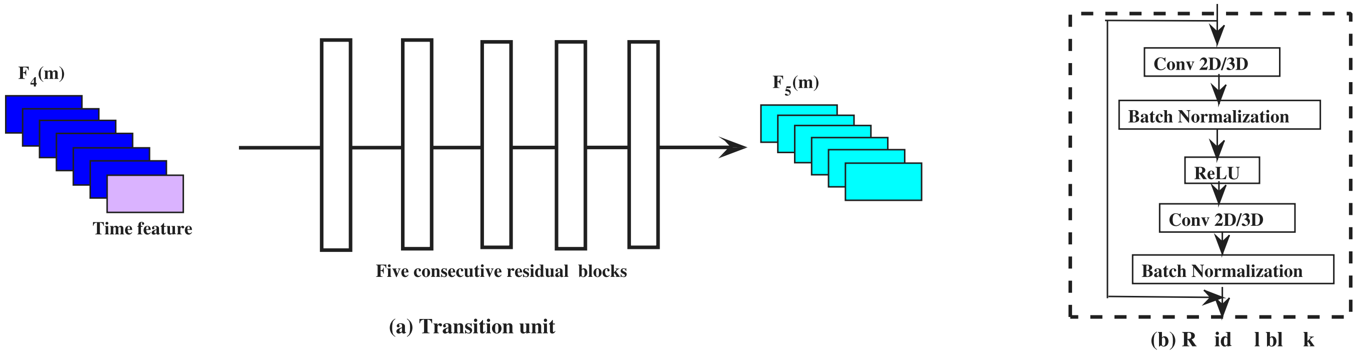


Fig. 6. Schematic illustration of transition unit for the cR-U-Net network. Five residual convolutional *resConv* blocks are used to propagate the feature maps from the encoding unit. In addition, the time feature as an additional channel is fed to this unit as well to capture the time-varying process.

To conduct the image segmentation problem effectively, we construct a deep convolutional neural network architecture, namely conditional residual-in-residual dense U-Net (cRRDB-U-Net) based on the integration of a standard U-Net architecture and a stack of residual-in-residual dense blocks [40]. The U-Net was originally proposed for bio-medical image segmentation [41]. The more recently developed residual-in-residual dense block (RRDB) was originally used for super-resolution image recovery and is adopted here to assist in the accurate image-based reconstruction of high-resolution parameter grids. The cRRDB-U-Net architecture contains an encoding path and a decoding path, which capture the hierarchical spatial encoding and decoding of the input and the output. The U-Net architecture is especially efficient in data utilization due to the design of the concatenating

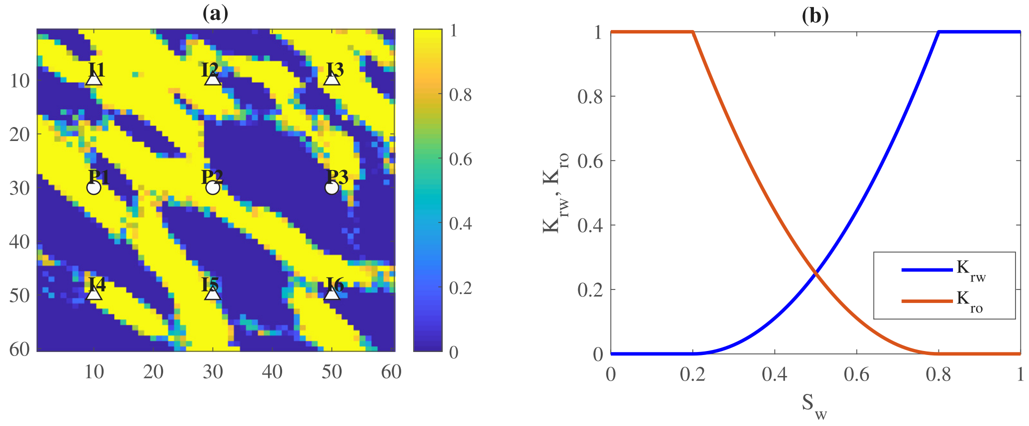


Fig. 7. The illustration of geological realization and well placement for the 2D non-Gaussian facies model. (a) the true model. The high-permeable and low-permeable channels represent two facies, which are indicated by binarized value 0 and 1, respectively; (b) the relative permeability curves of this water–oil two-phase flooding system. The triangles and circles denote the injectors and producers, respectively.

channel "highway". The concatenating channels pass the multi-scale spatial information obtained from the encoding unit to the corresponding decoding unit. The multi-scale encoding can aid the cross-domain predictions of interests, for example, in this paper, from geological parameters to spatially evolving fluid fronts. Between the encoding and the decoding path, a stack of advanced residual-in-residual dense blocks is employed to connect the encoding and decoding units [40].

In our proposed DNN model, it is necessary to include the time index to be able to predict the series of output images at different time instances. Specifically, the time value, as an additional channel, is concatenated to the low-dimensional feature representation produced by the encoding part of the network [25]. This is distinctive to other two approaches in the literature where either an autoregressive structure [24] or a recurrent neural network [45] are used to capture the time-dependent dynamics. Although an autoregressive structure excels at temporal regression tasks, it will definitely cause error accumulation and will also become computationally demanding for modeling of long time-series. The recurrent neural network might become computationally demanding for representing long time series of image outputs, but the time-dependent error accumulation problem will most likely not occur. [24].

Many advanced segmentation models have been reported in the literature [46]. Comparing to the static image segmentation models [47], our studied problem should be very analogy to the spatial–temporal video segmentation problem [48]. It thus deserves to discuss some segmentation methods in the related work. For example, Miao et al. proposes a memory-aggregation network to efficiently address the challenging time serial video segmentation [49]. This type of network model introduced a simple yet effective memory aggregation mechanism to record the informative knowledge from the previous interaction rounds, improving the robustness in discovering challenging objects of interest greatly. Yang et al. explored the functionality of embedding learning to cope with the video object segmentation task [50]. In this model, the background and foreground are equally treated with a novel collaborative video object segmentation approach. The use of feature embedding from both foreground and background can perform the segmentation process from both pixel and instance levels, which will make the proposed methodology quite robust to various object scales. Thus there may also be improvements possible by adopting these more effective network architectures designed especially for addressing our time-serial segmentation tasks.

An illustrative diagram of the cRRDB-U-Net architecture is displayed in Fig. 2. We illustrate individual parts of the architecture in more detail in Figs. 3 to 7. The figures are based on an example application to the prediction of two-phase fluid flow in geological reservoirs. In Fig. 3(a), the encoding unit takes the static geological parameter grid as input. The extracted feature maps $\mathbf{F}_k(\mathbf{m})$ (k = 1, 2, 3, 4, displayed as a set of pink rectangles in Fig. 2) from four consecutive convolutional blocks in the encoding unit will subsequently be delivered to the corresponding decoding unit. From $\mathbf{F}_1(\mathbf{m})$ to $\mathbf{F}_4(\mathbf{m})$, the extracted features become more compressed. Each convolutional block consists of three operations (*BN-ReLU-Conv2D/3D*), including a batch normalization layer (*BN*), a rectified linear activation unit (*ReLU*) and a convolutional layer (*Conv2D/3D*), see Fig. 3(b).

The features $\mathbf{F}_4(\mathbf{m})$ produced by the encoding unit are fed to a transition unit, see Fig. 4(c). The time, as a conditional feature channel, is concatenated to the low-dimensional features $\mathbf{F}_4(\mathbf{m})$ for representing the time-varying process. This transition unit is composed of two adjacent RRDB and convolutional blocks *BN-ReLU-Conv2D/3D*. The purpose of the transition unit is to produce features $\mathbf{F}_5(\mathbf{m})$, which are the most compressed maps and contain both spatial and temporal information. Subsequently, these features $\mathbf{F}_5(\mathbf{m})$ are fed to the decoding unit.

Generally speaking, deeper networks can approximate maps with higher complexity, but at a higher training cost [37,38]. Inspired by the successful applications of image super-resolution problems [39,40], the state of the art residual-in-residual dense block (RRDB) is used to ease the training process of a deeper neural network. The RRDB structure contains a well-designed combination of dense blocks and residual blocks. In order to take full advantage of the multi-scale features from the previous layers [51], a dense block intentionally connects the non-adjacent layers. For example, a dense block with five layers is shown in Fig. 4(a). The structure of the residual convolutional (*resConv*) block bypasses the nonlinear layers through introducing an identity mapping. This special architecture of the *resConv* block usually helps cope with the vanishing/exploding gradient problem when training deep networks [46]. We display the RRDB architecture in Fig. 4(b). It contains a stack of special structures where the dense blocks are embedded between two adjacent residual blocks. More details about RRDB can be referred to [40].

As shown in Fig. 5, the decoding unit takes the feature maps $\mathbf{F}_5(\mathbf{m})$ produced by the transition unit as its inputs. The features $\mathbf{F}_5(\mathbf{m})$ are gradually up-sampled via four consecutive transposed convolutional blocks. The transposed convolutional blocks are applied here for the purpose of increasing the size of feature maps.

Table 1
The network design of cRRDB-U-Net and cR-U-Net architectures used in this paper. This table shows the network structure for the 2D model, which can be easily extended to 3D model by using 3D convolutional operations. Here N_x and N_y denote the width and height of original images.

Encoding unit	Input	$(N_x, N_y, 1)$
	Conv2D/3D-BN-ReLU, 16 convolutional filters of size (3,3), stride 2	$(N_x/2, N_y/2, 16)$
	Conv2D/3D-BN-ReLU, 32 convolutional filters of size (3,3), stride 1	$(N_x/2, N_y/2, 32)$
	Conv2D/3D-BN-ReLU, 64 convolutional filters of size (3,3), stride 2	$(N_x/4, N_y/4, 64)$
	Conv2D/3D-BN-ReLU, 128 convolutional filters of size (3,3), stride 1	$(N_x/4, N_y/4, 128)$
Transition unit: **cRRDB-U-Net**	Input (outputs of encoder unit + an additional time feature)	$(N_x/4, N_y/4, 129)$
	BN-ReLU-Conv2D/3D, 129 convolutional filters	$(N_x/4, N_y/4, 128)$
	RRDB, 128 convolutional filters	$(N_x/4, N_y/4, 128)$
	RRDB, 128 convolutional filters	$(N_x/4, N_y/4, 128)$
	BN-ReLU-Conv2D/3D, 128 convolutional filters	$(N_x/4, N_y/4, 128)$
Transition unit: **cR-U-Net**	Input (outputs of encoder unit + an additional time feature)	$(N_x/4, N_y/4, 129)$
	resConv, 128 convolutional filters	$(N_x/4, N_y/4, 128)$
	resConv, 128 convolutional filters	$(N_x/4, N_y/4, 128)$
	resConv, 128 convolutional filters	$(N_x/4, N_y/4, 128)$
	resConv, 128 convolutional filters	$(N_x/4, N_y/4, 128)$
	resConv, 128 convolutional filters	$(N_x/4, N_y/4, 128)$
Decoding unit	Input	$(N_x/4, N_y/4, 128)$
	TConv2D/3D-BN-ReLU, 128 convolutional filters of size (3,3), stride 1	$(N_x/4, N_y/4, 128)$
	TConv2D/3D-BN-ReLU, 64 convolutional filters of size (3,3), stride 2	$(N_x/2, N_y/2, 64)$
	TConv2D/3D-BN-ReLU, 32 convolutional filters of size (3,3), stride 1	$(N_x/2, N_y/2, 32)$
	TConv2D/3D-BN-ReLU, 16 convolutional filters of size (3,3), stride 2	$(N_x, N_y, 16)$
	Conv2D/3D, 1 convolutional filter of size (3,3), stride 1	$(N_x, N_y, 1)$

Each transposed convolutional block consists of three operations (*BN-ReLU-TConv2D/3D*), including a batch normalization layer (*BN*), a rectified linear activation unit (*ReLU*) and a transposed convolutional layer (*TConv2D/3D*), see Fig. 5(b).

The up-sampled features are then combined with the previously extracted feature maps $\mathbf{F}_k(\mathbf{m})$ (k = 1, 2, 3, 4) from the encoding unit. Finally, the decoding unit generates the final outputs, e.g., a time series of binary fluid facies maps. To summarize, after the encoding unit, the inputs are compressed into 128 feature maps with the size of 15 × 15 (2D model) and 10 × 30 × 1 (3D model), and then concatenated with an additional time feature. These 129 features maps are fed to the transition unit for producing 128 constant size of feature maps, finally, these 128 feature maps are provided to the decoding unit for producing the output, e.g., binarized fluid-front images in this paper.

To verify the effectiveness of RRDB structure, a comparative study of the standard U-Net with and without this block is conducted. Specifically, we integrate a stack of residual convolutional *resConv* blocks [31] with the standard U-Net architecture, which can be referred to as cR-U-Net hereinafter. In addition to the transition unit, the overall neural network architecture of the cR-U-Net is very similar to the cRRDB-U-Net. In the cR-U-Net architecture, the feature maps produced from the encoding unit are concatenated with the time value and then are fed to five residual-blocks, see Fig. 6. The architectures of both cRRDB-U-Net and cR-U-Net are summarized in Table 1.

We should note that the choice of a neural network architecture is flexible and still relatively subjective. It is true that one can design a new DNN model by adapting parts of components of some existing DNN models, and satisfactory results can possibly be produced by different DNN models. In most cases we cannot provide standard guidelines to determine the optimal neural network architecture for a specific problem. In this paper, we have configured our cRRDB-U-Net architecture based on similar networks and applications found in the literature [31].

3.3.2. Dataset pre-processing and preparation

In order to train the cRRDB-U-Net surrogate model, we generate a set of training samples consisting of parameter grid as inputs and time series of binary fluid facies grid as outputs. The deep-learning based surrogate model represents the time-varying process

$$\hat{\mathbf{y}}^{i,n} = \hat{\mathbf{h}}^{n}(\mathbf{m}^{i}, t^{n}, \boldsymbol{\theta}), \quad n = 1, \ldots, N_t; \quad i = 1, \ldots, N_s. \tag{6}$$

where $\hat{\mathbf{y}}^{i,n} \in R^{N_x \times N_y}$ is the network prediction (image) at time $\mathbf{t}^n$ given the input $\mathbf{m}^i \in R^{N_x \times N_y}$, $\boldsymbol{\theta}$ denotes a vector containing all trainable parameters of the cRRDB-U-Net surrogate model, and i denotes the index of the training sample. N_s represents the total number of training samples.

Training data $\mathbf{y}^{i,n}$ is generated by simulating a high-fidelity forward simulation model (HFM) and selecting snapshots of its output at times $\mathbf{t}^n$. We will assume that the simulations produce maps of continuous state variables, which can be used for image regression tasks. We employ a post-processing step to convert the continuous maps $\mathbf{y}^{i,n}$ to binary maps $\mathbf{y}^{i,n}$ to address the image segmentation problem addressed in the paper. The binary output is obtained by applying a pre-defined threshold value S_{con} and the grid blocks (pixels) are assigned a value 0 or 1. We define a pixel-wise indicators $\mathbf{F}$ as follows.

$$\begin{aligned} &\mathbf{F}^{i,n} = 0, \mathbf{y}^{i,n} >= S_{con} \quad or \quad \mathbf{F}^{i,n} = 1, \mathbf{y}^{i,n} < S_{con} \\ &\hat{\mathbf{F}}^{i,n} = 0, \hat{\mathbf{y}}^{i,n} >= S_{con} \quad or \quad \hat{\mathbf{F}}^{i,n} = 1, \hat{\mathbf{y}}^{i,n} < S_{con}. \end{aligned} \tag{7}$$

Fig. 1 shows an example of continuous simulated HFM output (saturation maps) and the corresponding binarized images (binary contour maps) by applying Eq. (7).

The output $\mathbf{y}^{i,n}$ depends solely on the input image $\mathbf{m}^i$ and time index t^n. We rearrange the data structure predicted from one high-fidelity model simulation, i.e., saturation map ($\mathbf{m}^i$; $\mathbf{y}^{i,1}, \cdots$, $\mathbf{y}^{i,N_t}$), as N_t consecutive training samples

$$\{(\mathbf{m}^{i}; \mathbf{y}^{i,1}, \cdots, \mathbf{y}^{i,N_t})\}_{i=1}^{N_s} \Rightarrow \{(\mathbf{m}^{i}, t^{n}; \mathbf{y}^{i,n})\}_{i=1,n=1}^{N_s,N_t}. \tag{8}$$

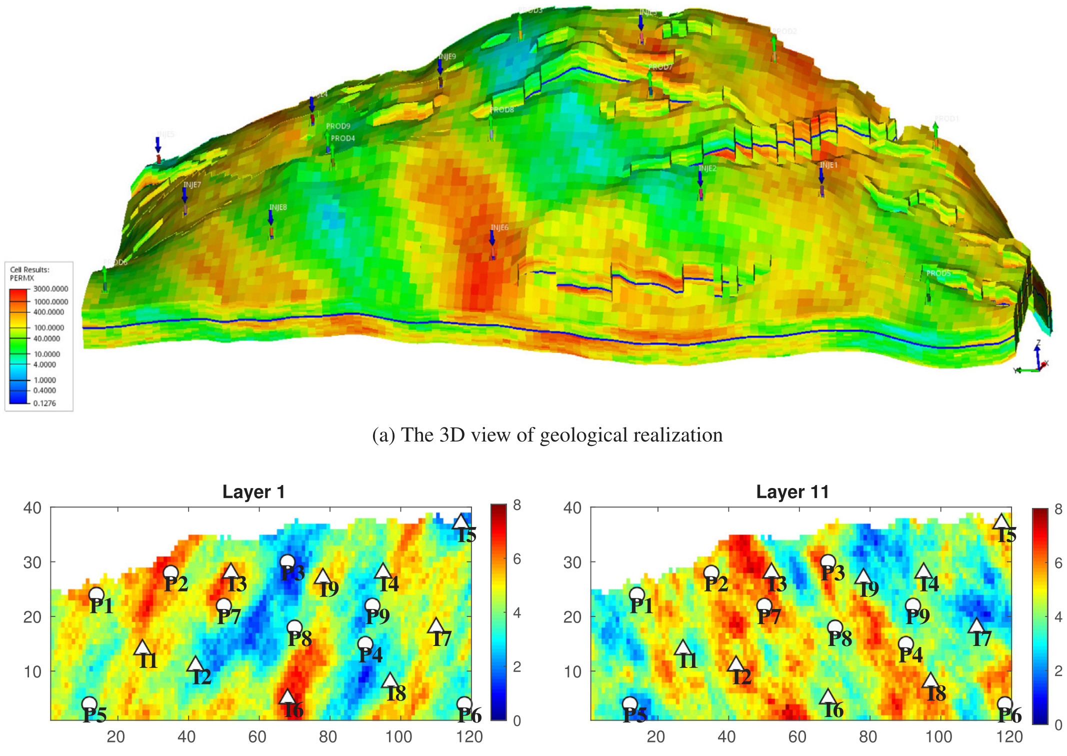
and the corresponding binarized maps

$$\{(\mathbf{m}^{i}; \mathbf{F}^{i,1}, \cdots, \mathbf{F}^{i,N_t})\}_{i=1}^{N_s} \Rightarrow \{(\mathbf{m}^{i}, t^{n}; \mathbf{F}^{i,n})\}_{i=1,n=1}^{N_s,N_t}. \tag{9}$$

in this way, the temporal relationship between the inputs and time-varying outputs is clearly captured in the time-conditional network structure. The total number of training samples fed to this cRRDB-U-Net becomes $N_s \times N_t$.

3.3.3. Training procedures

The choice of loss function for training neural networks is generally case-dependent. The choice of an appropriate loss function for the task at hand may strongly improve the performance of the network. The binary cross entropy (BCE) loss function is generally used for image segmentation tasks, while the mean square error



(a) The 3D view of geological realization

(b) The 2D cross-section of the 1st vertical layer

(c) The 2D cross-section of the 11th vertical layer

Fig. 8. The illustration of the spatial logarithmic permeability for the 1st and 11th horizontal layer. The triangles and circles denote the injectors and producers, respectively.

(MSE) loss function is more commonly used for image regression tasks. These two loss functions can be defined as follows

$$L_{MSE}(\boldsymbol{\theta}) = \frac{1}{N_s N_t} \sum_{i=1}^{N_s} \sum_{n=1}^{N_t} \| \hat{\mathbf{y}}^{i,n} - \mathbf{y}^{i,n} \|_2^2. \tag{10}$$

and

$$L_{BCE}(\boldsymbol{\theta}) = \frac{1}{N_s N_t N_{\mathbf{m}}} \sum_{i=1}^{N_s} \sum_{n=1}^{N_t} \sum_{j=1}^{N_{\mathbf{m}}} \mathbf{F}^{i,n,j} \log \hat{\mathbf{F}}^{i,n,j} + (1 - \mathbf{F}^{i,n,j}) \log(1 - \hat{\mathbf{F}}^{i,n,j}). \tag{11}$$

Algorithm 1 summarizes a conventional one-stage (OS) strategy for training the proposed cRRDB-U-Net model for image segmentation based on the BCE lose function. A variant of stochastic gradient descent optimizers, e.g., *Adam*, is used to train cRRDB-U-Net surrogate model. *Adam* computes adaptive learning rates for different parameters using estimates of the first and second order moments of the gradients. The learning rate controlling the magnitude of updates of model parameters at each iteration is 5×10^{-3}. In addition, a learning rate scheduler which drops ten times on plateau training is applied to guarantee a good convergence performance. This network is built and trained using the deep learning package PyTorch [30].

Although DNNs have shown promising and impressive performance in approximating the models with high-dimensionality and non-linearity, it is still very challenging to accurately predict the position of spatially discontinuous outputs, such as the shapes captured in binary images, as considered in this paper. It has been indicated in the literature that, in such cases, a two-stage (TS) training strategy through alternatively minimizing regression and segmentation loss functions is likely to improve the performance [39]. Taking into account that the aim of our proposed cRRDB-U-Net surrogate model is to accurately predict binary image data, we construct a combined loss function where a small weight is used to regularize the MSE loss rather than the BCE loss as suggested in [39] where an accurate prediction of spatially continuous grid-based fluid saturations was the final target. In our network training process, we adopt a similar TS strategy. A hybridization of the MSE and BCE loss functions with a predefined weighting coefficient ω, i.e., can be defined as follows

Algorithm 1: Procedure of optimizing neural network parameters $\boldsymbol{\theta}$ of cRRDB-U-Net using conventional one-stage (OS) training strategy.

```
Set an initial network trainable parameters $\boldsymbol{\theta}^0$ ;
while $epoch < n_{epoch}$ do
    while $minibatch < N_s \times N_t$ do
        Calculate the gradient $\bigtriangledown L_{BCE}(\boldsymbol{\theta})$ using
          auto-differentiation (AD) tool;
        Update the parameters $\boldsymbol{\theta}$ using $\textbf{Adam}(\boldsymbol{\theta}) \rightarrow \boldsymbol{\theta}$ ;
        Evaluate the loss function $L_{BCE}$ (i.e., Eq. (11));
        Check convergence;
    end
end
Return the optimal parameters $\boldsymbol{\theta}$
```

$$L(\boldsymbol{\theta}) = L_{BCE}(\boldsymbol{\theta}) + \omega L_{MSE}(\boldsymbol{\theta}) \tag{12}$$

The procedure of iteratively updating the neural network parameters using the TS training strategy is summarized in Algorithm 2. In each iteration, a subset of the training samples is randomly chosen from the full dataset, and then the tunable network parameters $\boldsymbol{\theta}$ of cRRDB-U-Net model are adaptively adjusted twice in a consecutive manner. In the first stage, the gradient of regression loss $L_{MSE}(\boldsymbol{\theta})$ is used to compute a preliminary update of the parameters. Then, in the second stage, the gradient of combined loss $L(\boldsymbol{\theta})$ (i.e., Eq. (12)) is used to further adapt the network parameters. These two training stages will be consecutively implemented for each data minibatch to update the parameters until the training process reaches convergence. In numerical experiments presented later, a comparative study of the trained cRRDB-U-Net using the TS (i.e., Algorithm 2) and OS (i.e., Algorithm 1) training strategies will be conducted.

Advantages of TS training strategy can be expected from two contributions: (1) Through ingesting the spatial continuities of state variables in the first training stage, the intrinsic physical principles are partially considered, which can help facilitate the network training by incorporating physical constrains. (2) Incorporating continuous state variable data acts as an effective means to augment the training sample size, which in turn enables partial mitigation of the overfitting problem and helps generalizing the CRRDB-U-Net surrogate model to generic models.

Algorithm 2: Procedure of iteratively optimizing the neural network parameters $\boldsymbol{\theta}$ using the two-stage (TS) training strategy. The MSE loss weighting coefficient ω is set to 0.01.

```
Set an initial trainable network parameters $\boldsymbol{\theta}^0$ ;
while $epoch < n_{epoch}$ do
    while $minibatch < N_s \times N_t$ do
        Stage (1) :    Calculate the gradient of $\bigtriangledown L_{MSE}(\boldsymbol{\theta})$ using AD tool;
        Update the parameters $\boldsymbol{\theta}$ using Adam($\boldsymbol{\theta}$) $\rightarrow \boldsymbol{\theta}$ ;
        Stage (2) :    Calculate the gradient $\bigtriangledown J(\boldsymbol{\theta})(\boldsymbol{\theta})$ using AD tool;
        Update the parameters $\boldsymbol{\theta}$ using Adam($\boldsymbol{\theta}$) $\rightarrow \boldsymbol{\theta}$ ;
        Check convergence;
    end
end
Return the optimal parameters $\boldsymbol{\theta}$
```

Once the neural network is trained using either the one-stage or two-stage training strategy, online prediction is straightforward. Given an arbitrary input $\mathbf{m}$, iterative implementation of Eq. (6) is then used to predict outputs for all N_t time instances. Specifically, each saturation output $\hat{\mathbf{y}}^n$ or binarized output $\hat{\mathbf{S}}^n$ at the nth time instance is sequentially predicted by providing the geological parameters $\mathbf{m}$ and the time index t^n as inputs. This procedure is computationally efficient as it does not involve any high-fidelity model simulations. After training the cRRDB-U-Net surrogate model successfully, we can apply the cRRDB-U-Net surrogate model within other workflows. Here we will consider its use for estimation of the uncertain parameters $\mathbf{m}$ given binary image data using the ES-MDA workflow discussed in Section 3.1.

4. Experiments and results

In this section, the proposed deep-learning hybrid framework will be applied to two example cases representing subsurface flow model parameter estimation problems in which 2D and 3D seismic images are available as measured data respectively. Both cases consider spatially heterogeneous reservoirs with immiscible two-phase (oil and water) flow dynamics.

4.1. Description of the example cases

The 2D reservoir model of Case 1 was created by [52] and consists of a single rock layer representing a fluvial depositional setting containing high-permeable channels (river deposits) and a low-permeable background (clay or fine sand deposits). The high-permeable and low-permeable channels represent two faices, which are indicated by binarized value 0 and 1, respectively. The value of log-permeability for these two facies has a large contrast. The permeability of clay facies and sand facies is 20 mD and 2000 mD, respectively. Given the facies indicators $\mathbf{m}$, we compute the permeability value for each grid using a transformation function $(20e^{log\,100\mathbf{m}})$. Fig. 7 illustrates the distribution of facies indicators in the high-fidelity model realization used to generate synthetic measurements, as well as the locations of 3 vertical liquid producer well and 6 vertical water injection wells labeled as P_1 to P_3, and I_1 to I_6 that are drilled into the reservoir. Note that the permeability values follow a non-Gaussian distribution containing two modes with nearly constant values, which is generally considered very challenging for parameter estimation methods.

Table 2
Experiment settings using OPM for these two numerical models.

Basic settings	Case 1	Case 2
Dimension, $N_x \times N_y \times N_z$	$60 \times 60 \times 1$	$40 \times 120 \times 20$
Number of injectors and producers	6, 3	9, 9
Water/oil density	1014 kg/m^3, 859 kg/m^3	
Water/oil viscosity	0.4 mP·s, 2 mP·s	
Initial oil/water saturation	$S_o = 0.80$, $S_w = 0.20$	
Bottom-hole pressure for producers	25 MPa	15 MPa
Bottom-hole pressure for injectors	40 MPa	30 MPa
Historical production time	1800 days	5400 days
Pre-defined threshold value S_{con}	0.35	

The second example case (Case 2) is a frequently used 3D benchmark model used in the SAIGUP project [42] with a realistic structure based on existing North Sea oil fields. The 3D SAIGUP benchmark model contains nine producers and nine injectors, which are labeled from P_1 to P_9, and I_1 to I_9, see Fig. 8(a). The triangles and circles denote the injectors and producers, respectively. Fig. 8(b)–(c) separately show the log-permeability fields of the 1st layer and 11th layer for the 3D SAIGUP model realization that will be used a synthetic truth, which are the Gaussian-distributed realizations with high spatial variability in both the horizontal and vertical directions.

In our numerical experiments, the open-source simulator *Flow* from the Open Porous Media (OPM) project for reservoir modeling and simulation [53], is used to run the high-fidelity (HF) model simulations and generate the training samples. The result of the simulations are pressure and saturation grids at the times of seismic repeat surveys and time series of bottom-hole pressure (BHP) and flow rates of both oil and water in all wells. In this study we will use the saturation grids (2D and 3D images for the two examples cases respectively) simulated with the synthetic truth models as measurements. Some details about reservoir geometry, rock properties, fluid properties, and well-control settings for the two example cases are shown in Table 2.

4.2. Training data generation

The prior uncertainty in the gridblock values of permeability is captured by an ensemble of random realizations of the permeability field. For the 2D non-Gaussian facies model, we use the 2000 facies realizations made available by [52]. For the 3D SAIGUP benchmark model we generate Gaussian-distributed realizations of log-permeability using the Stanford geo-statistical modeling software (*SGeMS*) [54]. An optimization-based principle component analysis (O-PCA) proposed in [52] and conventional PCA are applied to re-parameterize the parameter fields for these two models, respectively. 70 and 304 PCA coefficients are preserved to represent the original parameter fields in the two cases respectively and then used to generate the training and validation

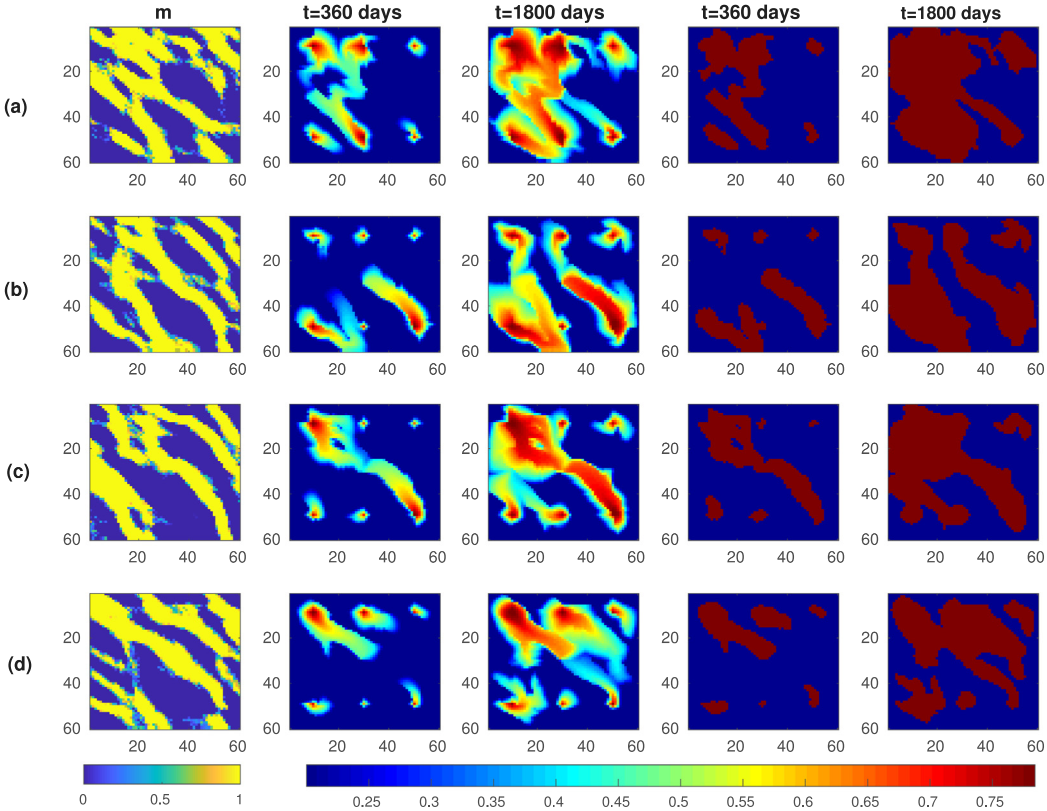


Fig. 9. Predictions of the time-varying saturation maps and binarized contour maps corresponding to four non-Gaussian realizations for the 2D synthetic non-Gaussian facies model. The contour maps are obtained through applying a front threshold value of 0.35. Subfigures (a)–(d) represent four different model realizations and their predictions at day 360 and day 1800, respectively.

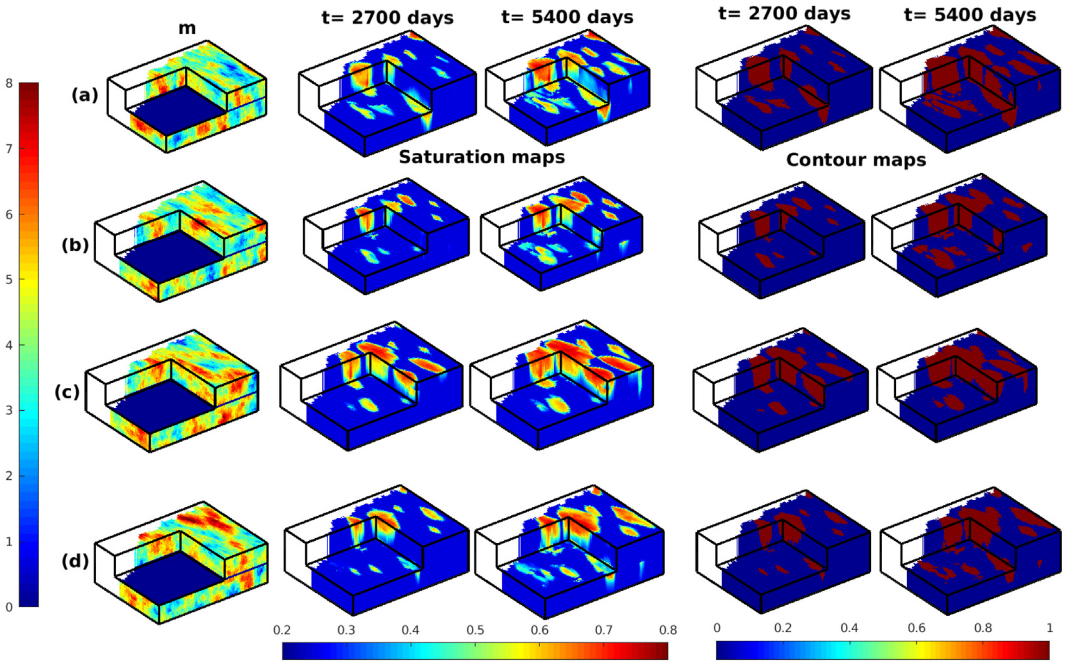


Fig. 10. Predictions of the time-varying saturation maps and binary contour maps corresponding to four Gaussian realizations of logarithmic permeability for the 3D SAIGUP benchmark model. The contour maps are obtained through applying a front threshold value of 0.35. Subfigures (a)–(d) represent four different model realizations and their predictions at day 2700 and day 5400.

samples. We should note that O-PCA is particularly useful to preserve the non-Gaussian properties. It reformulated the PCA as a bound-constrained optimization problem and introduced a regularization term to generate binary or bi-modal parameter fields.

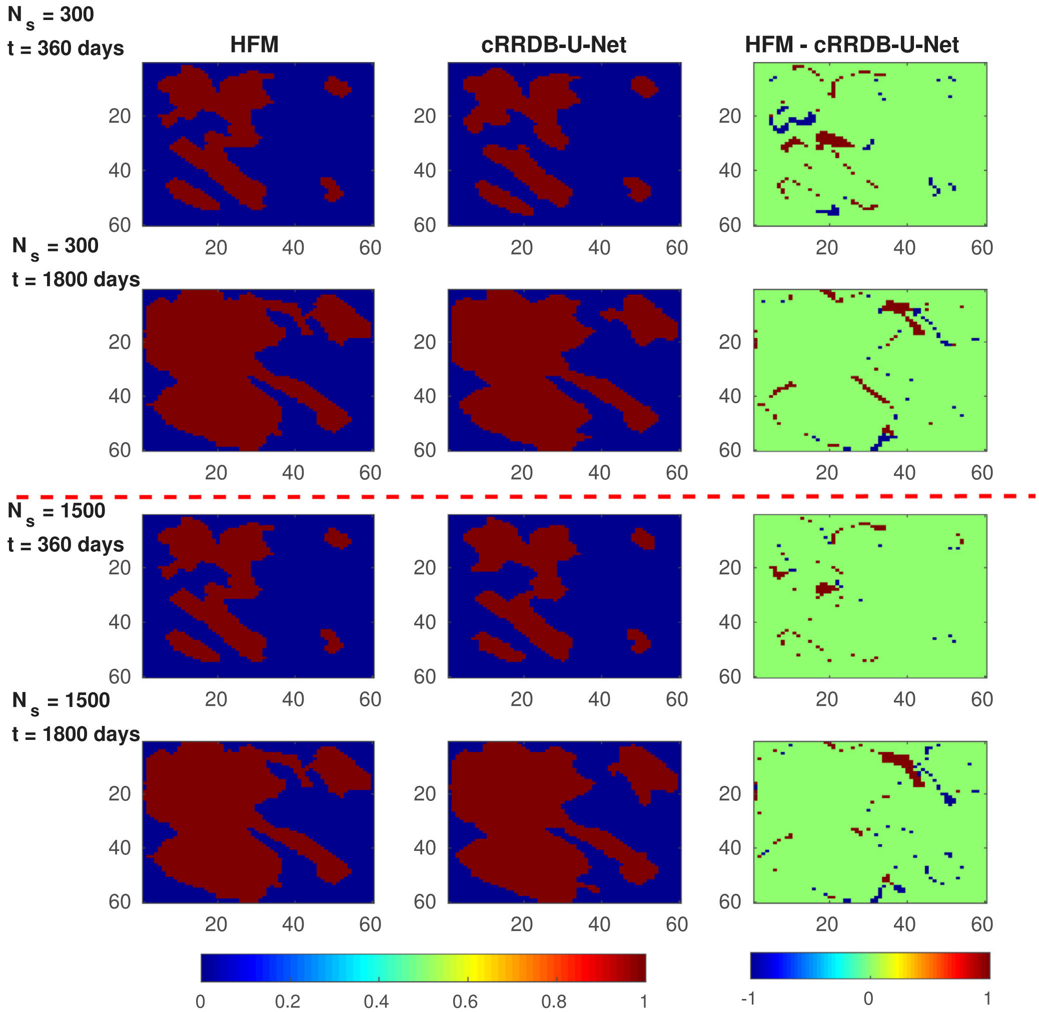


Fig. 11. Comparisons of time-varying fluid facies maps predicted from HFM, cRRDB-U-Net and their absolute errors at day 360 and day 1800 for the 2D synthetic non-Gaussian facies model. The cRRDB-U-Net models are trained using $N_s = 300$ and 1500 samples.

More information can be referred to [52]. We randomly generate an ensemble of the Gaussian-distributed PCA coefficients and then get the facies models using O-PCA. Although we suggest that the facies indicators should be binary 0,1, the generated realizations by O-PCA contain multiple colors (around the channel edges) and not all grid cells are classified as either 0 or 1, e.g., see Figs. 7–8.

Figs. 9 and 10 show the log-permeability (left) and simulation results at 2 times for four realizations of the Case 1 and Case 2 models respectively. The simulations results are the saturation images from which binarized images are derived based on a saturation threshold value of S_{con}=0.35. The model realizations shown in Fig. 9(a) and Fig. 10(a) are chosen to be the reference (synthetic truth) models for these two cases. It can be seen that, in both cases, the evolution of the fluid facies varies strongly among the different geological model realizations, resulting in high variability in the training dataset. We should note that while we use the LHDC as innovations in the history matching step, we will be showing the binarized fluid facies maps in subsequent figures, because they can be more easily interpreted.

The simulation period in the cases are 1800 days and 5400 days respectively, and the training sample data are collected at N_t=10 intervals of 180 days for Case 1, and N_t=10 intervals of 540 days for Case 2. After reorganizing the dataset, 3000, 5000, 8000, 10000 and 15000 training images are created corresponding to 300, 500, 800, 1000, and 1500 simulation runs respectively for the 2D non-Gaussian facies model.

4.3. Performance metrics

To evaluate the quality of cRRDB-U-Net surrogate model with respect to the number of training samples, N_{test} independent model simulations based on the HF and surrogate models are performed. We define an evaluation metric γ_s^n to represent the pixel-wise mismatch between two binarized images at timestep n evaluated over N_{test} validation samples is defined as

$$\gamma_s^n = \frac{1}{N_{test} N_{\mathbf{m}}} \sum_{i=1}^{N_{test}} \sum_{j=1}^{N_{\mathbf{m}}} \| \hat{F}_j^{i,n} - F_j^{i,n} \|, \quad n = 1, \ldots, N_t. \tag{13}$$

and can be interpreted as the fraction of incorrectly labeled grid cells (pixels) in the grid (image). $\hat{\mathbf{F}}_j^{i,n}$ and $\mathbf{F}_j^{i,n}$ denote the binarized fluid facies maps predicted from the high-fidelity model (HFM) and cRRDB-U-Net surrogate model respectively for validation sample i, gridblock j and timestep n. If the two images are equal, γ_s^n will attain it minimum value of 0, if no two values in the images are identical, γ_s^n=0. Note that differences between the binary images can be related to errors in the location of the fluid front that separates the two fluid phases in the reservoir. The overall field-average values over all N_t time instances, denoted as γ_s, is given by

$$\gamma_s = \frac{1}{N_t} \sum_{n=1}^{N_t} \gamma_s^n. \tag{14}$$

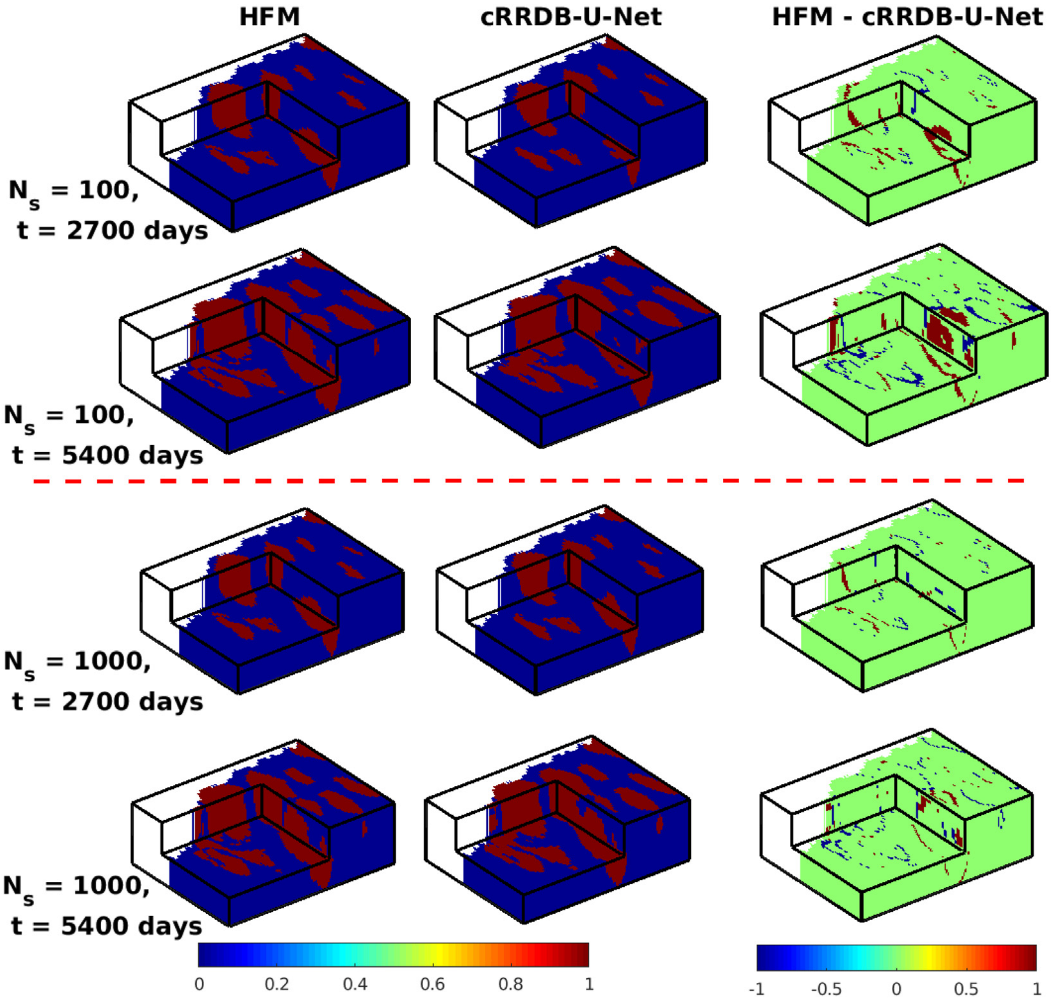


Fig. 12. Comparisons of time-varying fluid facies maps predicted from HFM, cRRDB-U-Net and their absolute errors at day 2700 and day 5400 for the 3D SAIGUP model. The cRRDB-U-Net models are trained using $N_s = 100$ and 1000 samples.

The binary facies indicators or log-permeability values are the only uncertain parameters and they are calibrated using the proposed ES-MDA framework using cRRDB-U-Net surrogate model. With the aim of analyzing the history matching results, we introduce two error metrics measured on data misfits e_{obs} and parameter misfits $e_{\mathbf{m}}$ as follows,

$$e_{obs} = \sqrt{\frac{\sum_{i=1}^{N_d}\sum_{j=1}^{N_t}(\mathbf{d}_{obs}^{i,j} - \mathbf{d}_{upt}^{i,j})^2}{N_d N_t}},$$
$$e_{\mathbf{m}} = \sqrt{\frac{\sum_{i=1}^{N_m}(\mathbf{m}_{true}^{i} - \mathbf{m}_{upt}^{i})^2}{N_m}}. \qquad (15)$$

where, $\mathbf{d}_{obs}^{i,j}$ and $\mathbf{d}_{upt}^{i,j}$ represent the measurements and simulated data using the updated model, respectively. $\mathbf{m}_{true}^{i}$ and $\mathbf{m}_{upt}^{i}$ denote the binary facies indicators or logarithmic permeability values from the 'true' model and updated model, respectively.

4.4. Training and validation of the surrogate

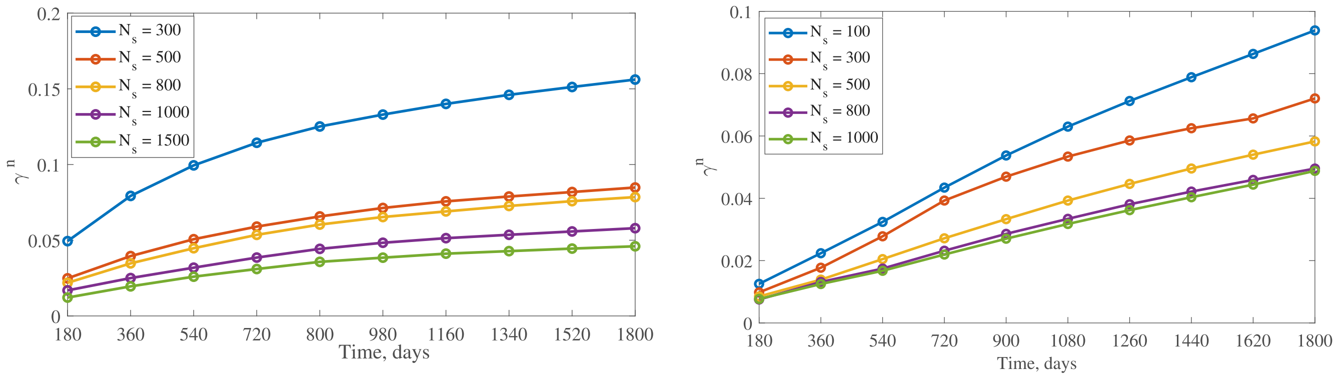
The parameter settings for training the cRRDB-U-Net surrogate are summarized in Table 3. These parameters were used to train the surrogate models for all listed training set and batch sizes. Taking the 2D synthetic model as an example, during the training process, 100 training samples are randomly selected from the entire, e.g., 20000, training dataset to optimize the neural network parameters in each iteration. In the following we will discuss the quality of predictions (i.e. the fluid facies maps or images) generated with the trained surrogate.

Figs. 11 and 12 show the fluid facies maps predicted by the HF and cRRDB-U-Net surrogate models for the reference realizations of Case 1 and Case 2 respectively. It can be seen in Fig. 11 for Case 1 that the trained surrogate model is capable of predicting accurate fluid distributions. For instance, the presence of single or multiple fluid fronts at different times is correctly captured as seen in Fig. 11. Small errors are noticeable, however, which are associated with small errors in predicted front locations. These errors are seen to decrease with increasing number of training samples N_s. The impact of the number of training samples is particularly clear at early times where multiple isolated fluid fronts are developing. Fig. 12 displays analogous results at day 2700 and day 5400 for Case 2. in this case the possibility of vertical flow results in somewhat larger regions of error for small training set size.

We further assess the quality of proposed surrogate model based on the γ_s^n and γ_s metrics which quantify the quality of image reconstruction, particularly taking into account the structural similarity of edges inside the images. In order to verify the robustness of the surrogate model, the values of γ_s^n corresponding to $N_{test} = 200$ independent validation samples are plotted in Fig. 13. Increasing the number of training samples progressively improves the values of γ_s^n metric. Values are relatively high at

Table 3
Training parameters settings for training cRRDB-U-Net surrogate model.

Training size (N_s)	300,500,800,1000,1500		100,300,500,800,1000
Re-organized training size ($N_s \times N_t$)	2000,6000,10000,16000,20000		1000,3000,5000,8000,10000
Testing size (N_{test})		200	
Initial learning rate		0.005	
Optimizer		*Adam*	
Batch size	100		10
Number of epochs		100	



(a) 2D non-Gaussian facies model

(b) 3D Gaussian SAIGUP model

Fig. 13. The plots of γ_s^n values of cRDDB-U-Net with respect to the number of training samples N_s = 300, 500, 800, 1000, and 1500 samples for the (a) 2D non-Gaussian facies model; (b) 3D Gaussian SAIGUP model.

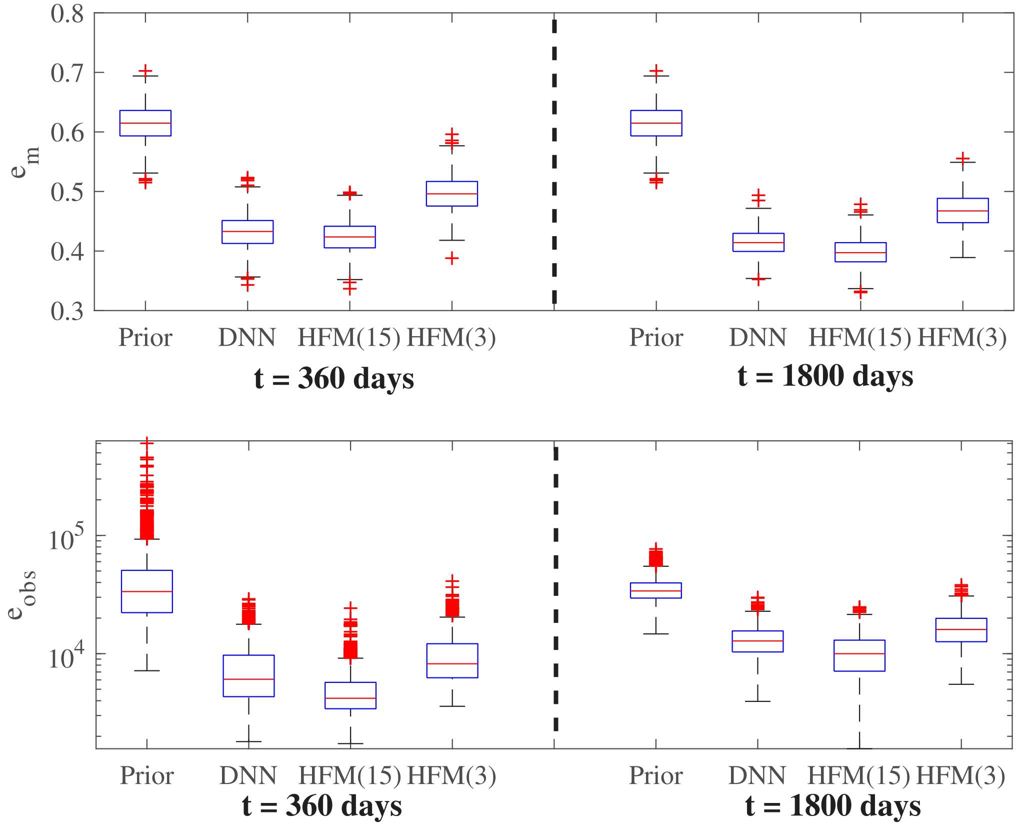


Fig. 14. Boxplots of data misfits and parameter errors using cRRDB-U-Net surrogate model and high-fidelity models for the 2D non-Gaussian facies model. The cRRDB-U-Net surrogate model is trained by 1500 training samples. HFM(3) and HFM(15) are the abbreviations of ES-MDA-HFM with 3 and 15 iterations, respectively.

early times because the fluid fronts have not yet expanded very strongly such that most of the domain will still contain only the initial fluid phase (oil in this case) in all test cases. The values are gradually decreasing with time as the injected fluid phase is replacing the initially present phase, leading to expanding fluid fronts, and correspondingly higher chances of differences between the true and predicted front locations. The surrogate is obtaining relatively lower γ_s^n for Case 2 than for Case 1. The overall field-average error γ_s is 4.24% for 1000 training samples, suggesting a relatively high degree of accuracy.

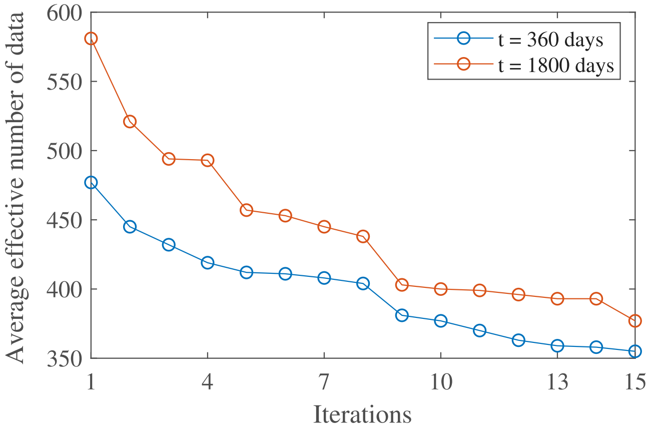


Fig. 15. The over-all effective number of data during the iteration process.

4.5. History matching results — Case 1

In the previous section we verified the applicability of our proposed surrogate model for predicting dynamic fronts. We now apply the surrogate model within a history matching (i.e. data assimilation) workflow, where the image-oriented distance parameterization is used to characterize differences between observed and simulated images, and the ES-MDA method is used to minimize these differences by updating the underlying model parameters. In the examples these parameters are the properties of the grid-cells (specifically, the permeability) of the HF model. While the total number of active grid cells are 3600 for Case 1 and 78720 for Case 2, the PCA parameterization has reduced this to 70 and 304 coefficients respectively (See Section 4.1). The observations correspond to the LHDC metric derived from applying a threshold value of S_{con} = 0.35 to the saturation maps at day 360 or at day 1800 resulting from simulation of the HF reference (synthetic truth) model, and the corresponding maps from a surrogate model simulation. The standard deviation of the uncorrelated measurement errors is assumed to be 30 m, which is close to the length of one grid block. Results from the hybrid workflow will be compared to results obtained by using an ensemble of HF model realizations instead of the trained surrogate. We choose $N_a = 15$ iterations and an ensemble size of $N_e = 500$ as standard values for the ES-MDA workflow and we compare results obtained with the hybrid workflow for a range of training set sizes.

4.5.1. Fixed computational budget

We first compare results from the hybrid workflow against results from the HF model workflow for (a) a fixed number of HF model simulations (1500), and (b) a fixed number of iterations (15). Fig. 14 compares the data misfits for the prior and posterior ensemble of realizations obtained with the DNN trained with $N_s = 1500$, and the posterior HF mode realizations resulting from $N_a = 15$ iteration (HFM(15)) and $N_a = 3$ iterations (HFM(3)). Note that HFM(3) requires $3 \times 500 = 1500$ HF model simulations, which is the equal to the 1500 simulations used to train the DNN, while HFM(15) requires $15 \times 500 = 7500$ HF model simulations. Results are presented for images obtained at 360 days and 1800 days separately. The DNN-based data misfits are larger than those obtained with 15 ES-MDA iterations with the HF model, but smaller than those obtained with 3 iterations. These results indicate that a significant reduction of computational cost should be feasible for a given desired quality of the result. In

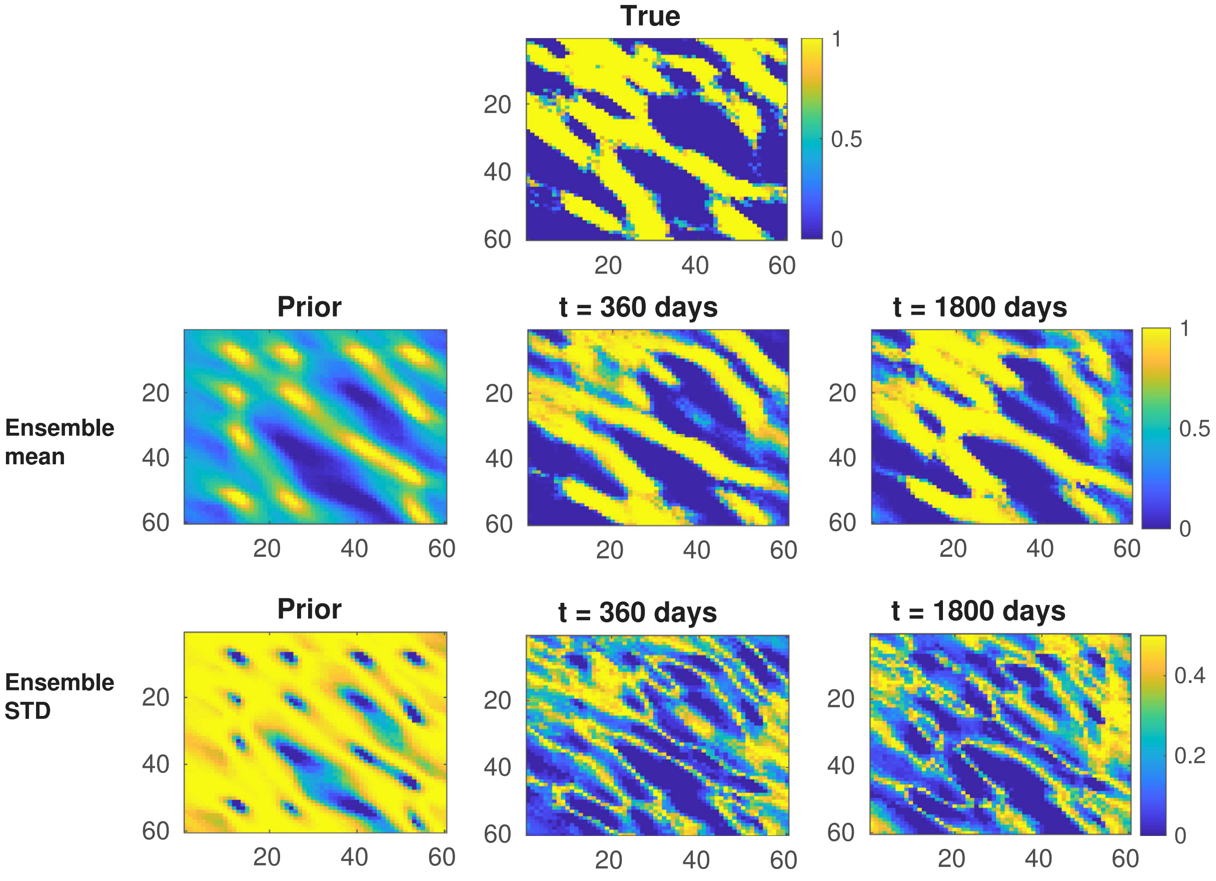
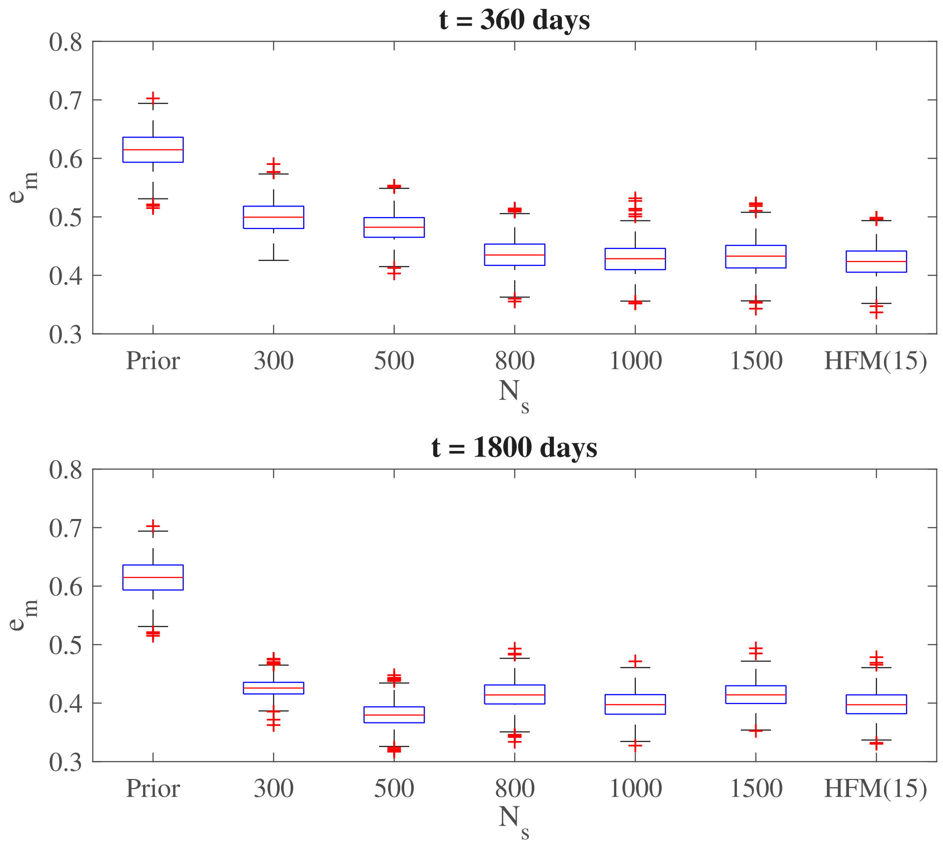


Fig. 16. Ensemble mean and standard deviation (STD) of binary facies indicators for the prior models and posterior models at day 360 and day 1800.


t = 360 days

t = 1800 days

Fig. 17. Boxplots of parameter misfits $e_\mathbf{m}$ using cRRDB-U-Net surrogate model as a function of the number of training sample N_s. The two rows are corresponding to day 360 and day 1800, respectively.

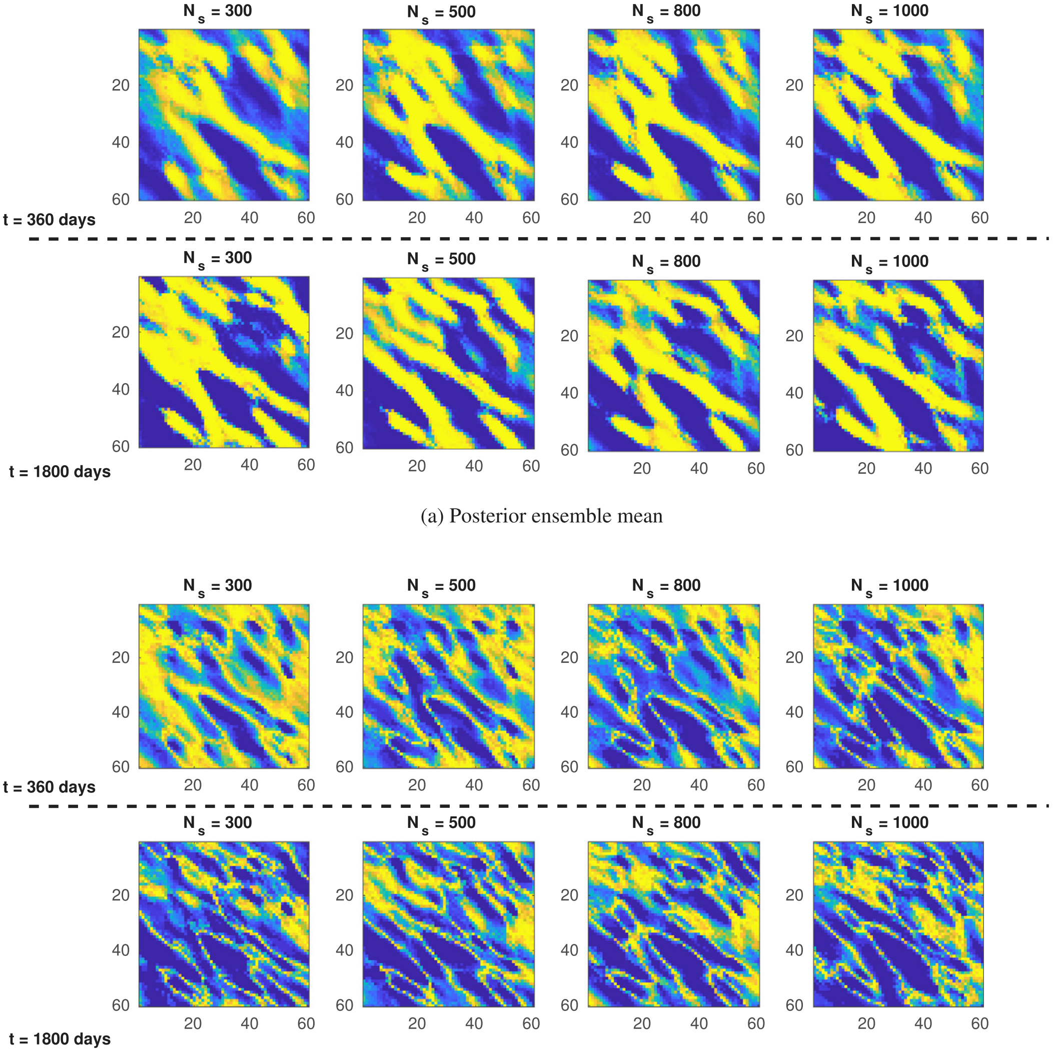
this study, the total number of HFM simulations is taken as an indicator of the computational cost, since the GPU time for running the DNN surrogate model is negligible compared to the CPU time for running the HF model. The runtime for training the DNN with data from $N_s = 1000$ samples is about 25 min, which is equivalent to running 300 HF model simulations. In other word, the computational cost of the off-line training stage is equivalent to about 1300 HF model runs, while the cost of HFM(15) is equal to 7500 HFM simulations. The use of cRRDB-U-Net as a surrogate model reduces the computational time by a factor 5.8 for the 2D synthetic model of Case 1. Note that the computational saving of our proposed deep-learning method will increase linearly with the ensemble size.

Fig. 15 displays the average effective number of data as a function of iterations. Pre-processing the imaging-type data using distance-based parameterization can drastically decrease the number of data, for example from original 3600 to 580 at day 1800. A small effective number of data indicates a high degree of similarity of two images. The effective number of data gradually decreases as the iterations, revealing that the predicted water fronts from the posterior models become closer to the observed water fronts and the model uncertainty is hence reduced.

The ensemble mean and standard deviations of the updated ensemble of permeability before and after history matching at 360 days and 1800 days are displayed in Fig. 16. These statistics are calculated as the per-grid-block averages and standard deviations over the 500 realizations in the ensemble $\mathbf{m}_1^{N_a}, \ldots, \mathbf{m}_{N_e}^{N_a}$, where $N_e = 500$ and $N_a = 15$. Values are compared to the corresponding statistics calculated for the prior ensemble (iteration 0 instead of N_a). The channel structures can be reconstructed almost perfectly from both the 360 and 1800-day images. While the large standard deviations in the prior standard deviation suggest that in the initial ensemble the locations of channel boundary positions are strongly varying, a comparison of the posterior standard deviation maps and the true parameter map indicates that the channel boundaries are consistently aligned with the truth. Larger variability is mostly found in this bands a few grid blocks wide along the true channel boundaries. The field-average posterior ensemble standard deviation has approximately decreased from 0.45 to 0.16, which indicates a significant reduction of model uncertainty. We also should note that since the original realizations conditioned to permeability values at the well locations, the majority of realizations, e.g., the prior ensemble mean, already seem to have channels at approximately the right locations.

4.5.2. *Effects of training sample size*

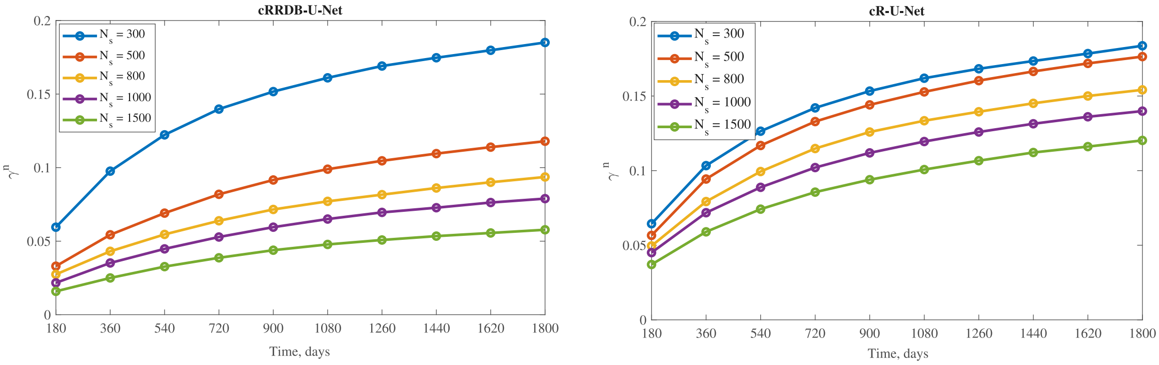
We repeat the entire workflow for a series of increasing training set sizes $N_s = 300, 500, 800, 1000, 1500$. Fig. 17 shows the posterior parameter misfits as a function of the number of training samples. It clearly can be seen that the accuracy of the hybrid workflow results gradually improves as the number of training samples increases, especially for data gathered at day 360. Results do not improve much for $N_s > 800$. For data gathered at 1800 days, the best results are obtained, somewhat surprisingly, for $N_s = 500$. Since the different training scenarios rely on different random parameter realizations, the results of each training stage could be impacted by the samples included in the training set. ES-MDA is a statistical method and results could also vary slightly if the ensemble statistics are affected by these results as well. One could, in principle, repeat each experiment with training sets containing different random samples, to quantify the impact of the random sample selection on the results, but we have not done that here. The overall trends in the results, however,



(a) Posterior ensemble mean

(b) Posterior ensemble standard deviation (STD)

Fig. 18. Posterior ensemble mean and standard deviation (STD) of parameter estimates obtained for 300,500,800,1000 and 1500 training samples. (a) Posterior ensemble mean; (b) posterior ensemble STD.



(a) cRRDU-U-Net

(b) cR-U-Net

Fig. 19. The γ_s^n values of cRDDB-U-Net and cR-U-Net architecture with respect to the number of simulation models for training network, i.e., N_s = 300, 500, 800, 1000, and 1500, respectively. The results correspond to (a) cRRDB-U-Net surrogate model using **OS** training strategy; (b) cR-U-Net model.

clearly indicate that the accuracy of the surrogate model and its corresponding history matching results will improve as the number of training samples increases.

Fig. 18 shows the posterior ensemble mean and standard deviation (STD) for the different N_S scenarios. It is evident that the main structure of binary channels can be successfully reconstructed in all cases, also for the smaller values of N_s, and that the posterior ensemble mean gradually becomes much closer to the true model as the number of training samples increases. Based on Fig. 17 Fig. 18 it can be concluded that results with similar

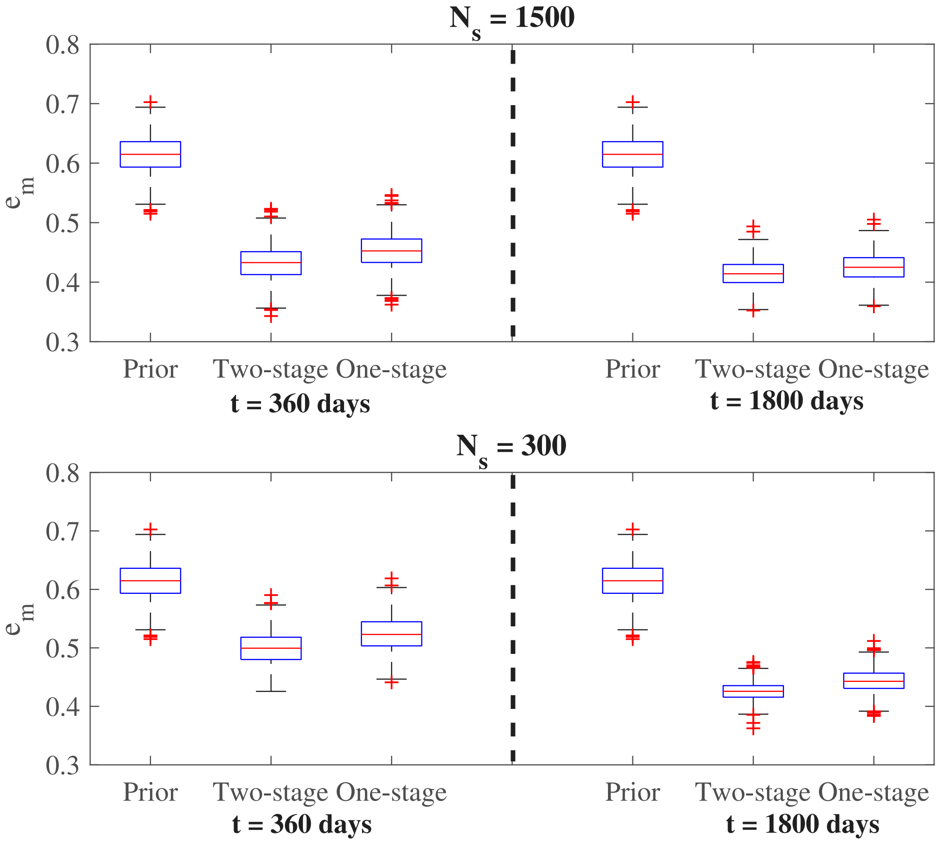


Fig. 20. Boxplots of parameter misfits $e_\mathbf{m}$ for the **TS** and **OS** training strategy. The two rows are corresponding to $N_s = 1500$ and $N_s = 300$ training samples, respectively.

quality as the HF model workflow requiring 7500 simulations can be obtained using the hybrid surrogate-supported workflow with 500–800 simulations. Overall, we can collect more informative measurements at day 1800, and therefore obtain relatively better results than that of day 360.

We summarize the posterior parameter misfits $e_\mathbf{m}$ and computational cost indicated by the number of HFM simulations. It can be obviously seen that the proposed cRRDB-U-Net surrogate model maintains a satisfactory accuracy even using a small number of training samples. For example, the cRRDB-U-Net model using N_s = 500 training samples is still capable of generating satisfactory posterior models which are very similar to the true model. In terms of computational efficiency, it clearly reveals that our proposed history matching framework only needs a relatively small number of high-fidelity model simulations, e.g., 500, in contrast to 7500 HFM simulations required by the conventional ES-MDA method.

4.5.3. Comparative study of one-stage and two-stage training strategies

The results presented so far were obtained using the two-stage training strategy detailed in Section 3.3.3. We repeated the training procedure for $N_s = 300, 500, 800, 1000, 1500$ in order to compare two-stage strategy with the one-stage strategy, which only considers the BCE loss function which is commonly used for image segmentation tasks. Results from the one-stage training procedure are illustrated in Fig. 19(a). When comparing with the results shown in Fig. 13(a), it is clear that the two-stage strategy achieves lower validation errors than the one-stage strategy. Fig. 20 shows a comparison of the posterior parameter misfits for these two training strategies for $Ns = 300$ and $N_s = 1500$. When we train the surrogate models with N_s=300 training samples, the one-stage strategy achieves γ values of 9.85% and 18.82% at day 360 and day 1800, which are larger than the 7.12% and 15.12% errors from two-stage strategy. However, the differences between these two training strategies decrease as the number of training samples increases. For example, when we train the surrogate model using 1500 samples, the one-stage strategy achieves γ_s values of 3.01% and 5.18% at day 360 and day 1800, respectively, which are only slightly larger than the values of 2.25% and 4.89% from the two-stage strategy. Overall, two-stage training is found to improve the predictions, particularly when small numbers of training samples are available.

We remind the reader that the first training aims to minimize the MSE loss expressed in terms of continuous saturation values. It has been indicated in the literature that a two-stage (TS) training strategy through alternately minimizing regression and segmentation loss functions is likely to improve the performance. For example, the two-stage procedure was also used by Mo et al. (2018) who also use a weight 0.01 and was the basis for our choice. A larger weight ω generally will lead to a better approximation of the spatially continuous saturation. However, the aim of our proposed surrogate modeling and data assimilation workflow is to accurately predict binary-type image data. Some limited experimentation with alternative values of *omega* (not included) suggested that larger values will lead to less accurate reproduction of the features in the binary images. Based on visual inspection of the results for different values of ω, a value of 0.01 appeared to be a good choice, but the issue of finding the optimal value is subject to further investigation.

4.5.4. Comparison with conventional residual U-Net

To verify the effectiveness of the RRDB structure of the deep CNN, a comparative study of cR-U-Net and cRRDB-U-Net was

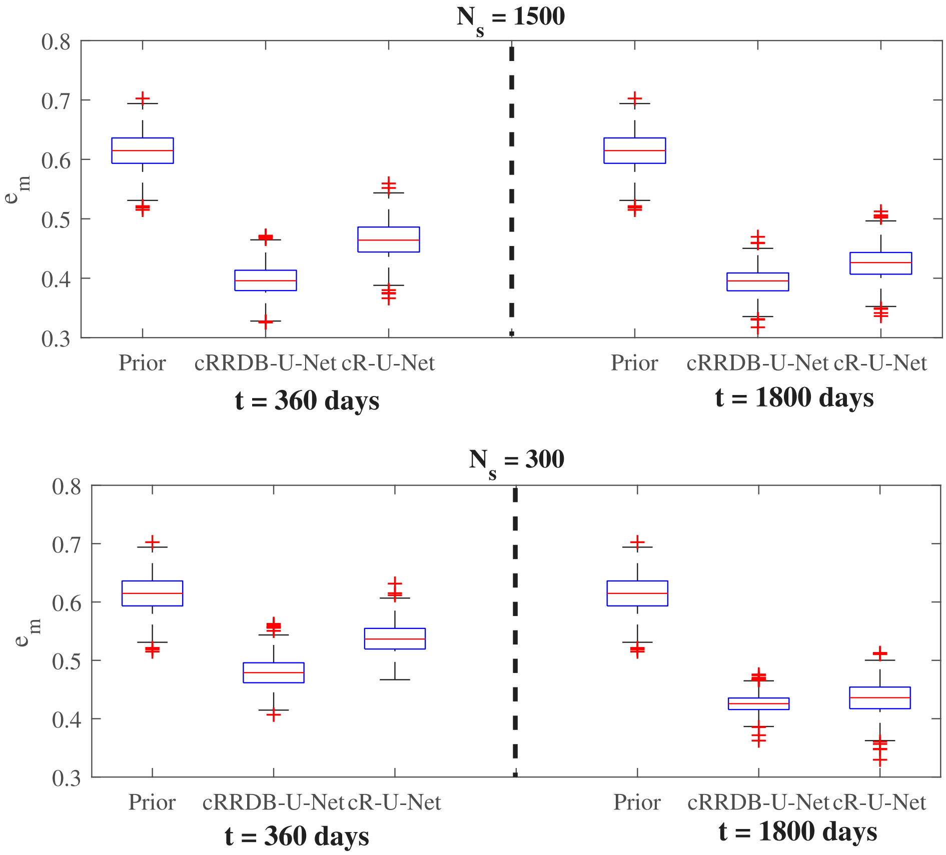


Fig. 21. Boxplots of parameter misfits $e_\mathbf{m}$ using cR-U-Net and cRRDB-U-Net surrogate models. The two rows are corresponding to $N_s = 1500$ and $N_s = 300$ training samples, respectively.

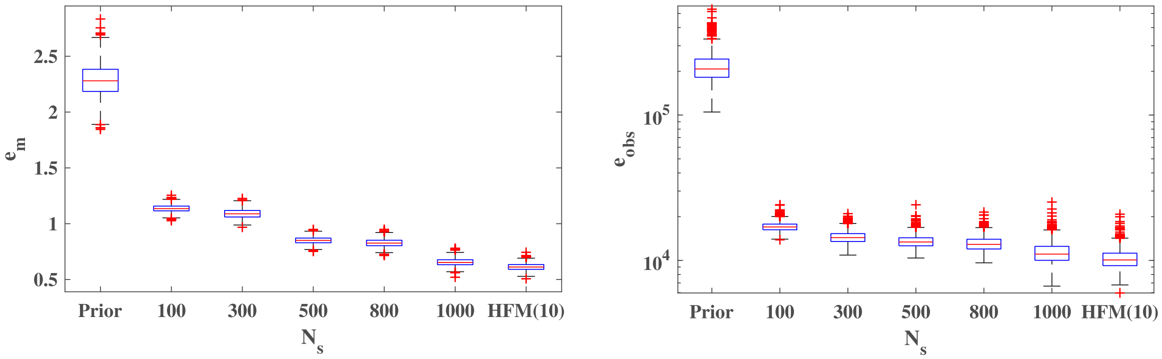


Fig. 22. Boxplots of data misfits and parameter misfits using cRRDB-U-Net surrogate model and high-fidelity model for the 3D benchmark SAIGUP model. The cRRDB-U-Net models are trained with $N_s = 100$, 300, 500, 8000 and 1000 samples, respectively.

performed. Fig. 19(b) shows the parameter error metric γ_s for the cR-U-Net surrogate model. Comparing to the results in Fig. 13(a), the improvements from the RRDB structure are clearly indicated lower field-average relative error γ_s values, especially for the larger training set sizes. When we train the deep-learning surrogate models using 300 samples, the cR-U-Net and cRRDB-U-Net obtain comparable results with γ_s values around 18%. Although RRDB enables us to train deeper neural networks for better approximations of spatially discontinuous fluid-fronts, more parameters are introduced by the RRDB structure that need to be trained as well. Training the cRRDB-U-Net surrogate model with a small number of samples might cause overfitting problems, and hence may not achieve desirable improvements in comparison to cR-U-Net. As the number of training samples increases, however, the improvements in quality of the cRRDB-U-Net surrogate model are much larger than that of cR-U-Net. For example, when we train these two surrogate models using 1500 samples, cR-U-Net and cRRDB-U-Net obtain γ_s values of 12.5% and 4.89% at day 1800, respectively. Overall, the RRDB structure significantly improves the surrogate model's ability to predict the positions of fluid fronts, especially for large training set sizes. Fig. 21 shows a comparison of the posterior parameter misfits for the cRRDB-U-Net and cR-U-Net surrogate models. It clearly can be seen that the RRDB structure outperforms standard U-Net in terms of the posterior parameters errors.

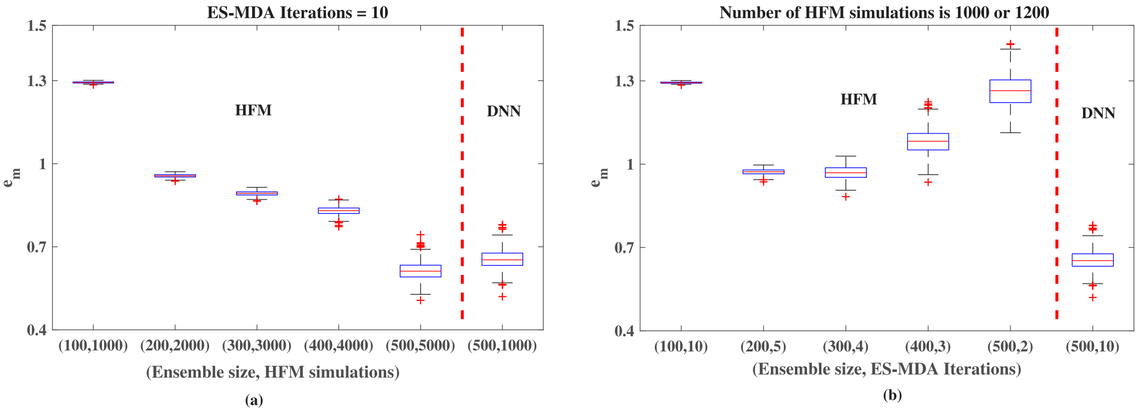


Fig. 23. Boxplots of posterior ensemble parameter errors $e_{\mathbf{m}}$ as a function of ensemble size N_s and ES-MDA iterations. (a) The ES-MDA iteration number is 10 and the ensemble size ranges from 100 to 500; (b) The number of HFM simulations is about 1000 or 1200.

4.6. History matching results — Case 2

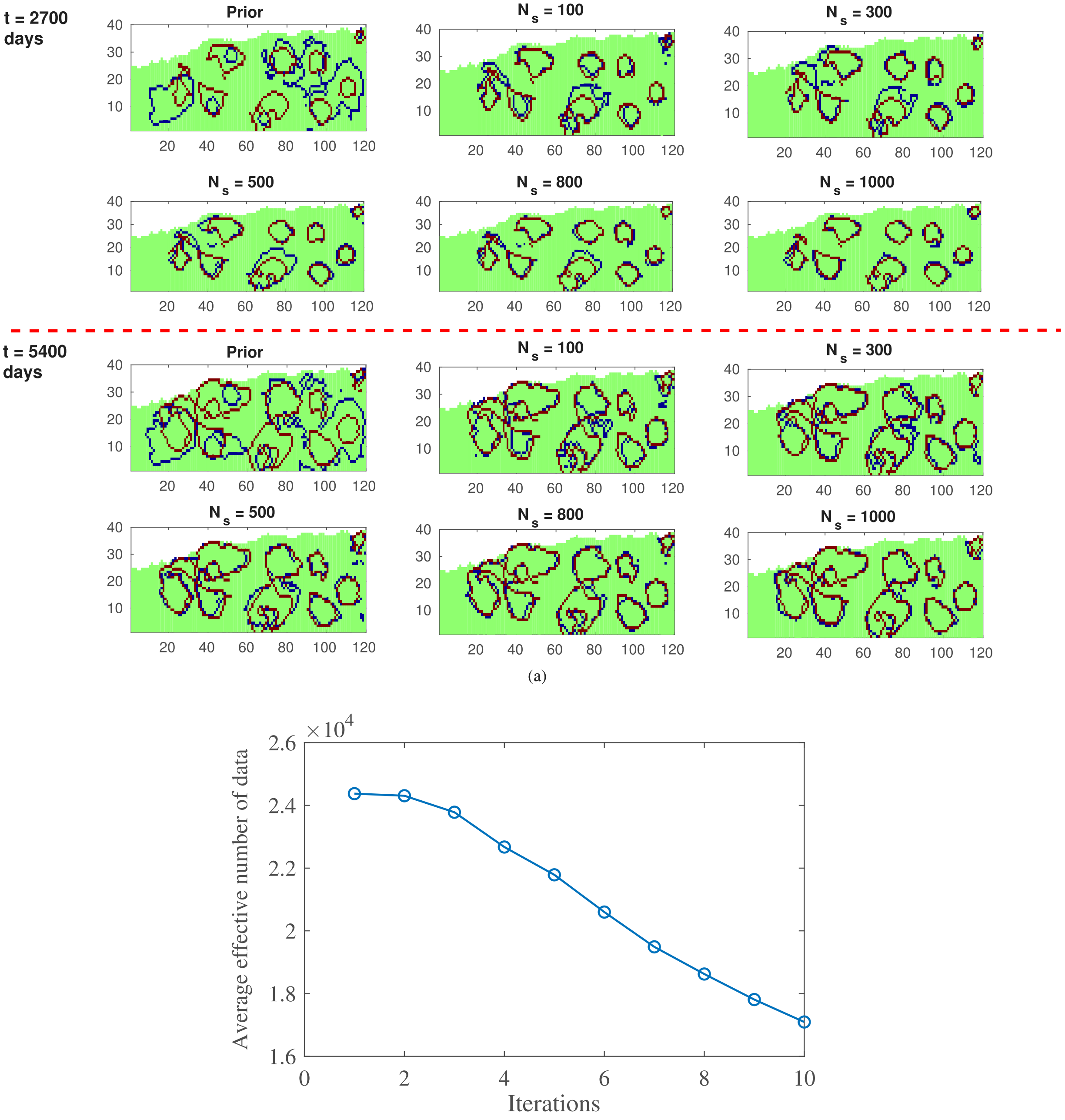
In this section we present a more realistic application of the proposed surrogate model that involves a more complex 3D reservoir model with uncertain permeability in all 78720 active grid cells. The dimension of the uncertainty space is reduced by capturing the spatial relationships between individual grid block values in a total of 304 coefficients for corresponding 3D patterns obtained by Principle Orthogonal Analysis of a large number of plausible log-permeability realizations. The PCA coefficients are the only uncertain parameters, and are updated using the proposed hybrid workflow. We assume that saturation images are available at either 2700 days or 5400 days, after inversion of time-lapse seismic data. We furthermore assume that the data has sufficiently high resolution to enable estimation of saturation in all 22 layers of the reservoir. We use an ensemble size of $N_e = 500$ and a fixed number of $N_s = 10$ iterations for the ES-MDA algorithm.

Fig. 22 shows the parameter misfits $e_{\mathbf{m}}$ before and after history matching for training set sizes N_s= 100, 300, 500, 800 and 1000. Fairly consistent improvements in the accuracy of the results can be observed with increasing training set size. In this case study, the surrogate models trained with 100 and 300 samples obtain similar history matching results. Similar results are also observed for the surrogate models trained with 500 and 800 samples. A possible explanation is that a random sampling strategy might generate some ineffective training data, which have no significant contribution to the improvement of the history matching results. One could imagine that procedures in which additional training samples are generated guided by intermediate history matching results could be more effective, but we have not investigated this idea further.

Our proposed ES-MDA-DNN is able to generate history matching results comparable to ES-MDA-HFM. However, to achieve the same accuracy, ES-MDA-DNN and ES-MDA-HFM require respectively 1000 and 5000 high-fidelity model simulations. The use of cRRDB-U-Net as a surrogate model reduces the computational time by a factor of 5 for the problem defined in this case. We should emphasize that the computational saving will become more substantial when a larger ensemble sizes are used. The ES-MDA-DNN hybrid workflow requires a relatively small number of high-fidelity model runs at the offline training stage and the computational cost does not grow with the ensemble size. Compared to the HFM(10) results, the hybrid DNN workflow is generally performing a little bit worse but at a lower cost. The use of 500 initial realizations might be much larger than that would be used in applications to complex fields, where an ensemble size of 200 is often considered as the realistic maximum. Fig. 23 shows the posterior ensemble parameter errors $e_{\mathbf{m}}$ as a function of ensemble size and number of ES-MDA iterations. It can be clearly seen in Fig. 23(a) that our proposed ES-MDA-DNN obtains a better solution that is at present practically feasible at lower computational cost (e.g., 1000 simulations). By contrast, ES-MDA-HFM would lead to a total number of 2000 HFM simulations over 10 iterations. Furthermore, the obtained results with an HF model ensemble size of 100 suggest ensemble collapse to some degree. As illustrated in Fig. 23(b), for a fixed computational budget (1000 simulations in this example), the hybrid surrogate-assisted workflow delivers more accurate results than ES-MDA-HFM.

Fig. 24(a) shows the fluid-front positions before and after history matching at day 2700 and day 5400. The fluid-fronts for the prior models are significantly different from the observed front positions. There are nine completely discrete front contours around the nine injectors at day 2700. After the history matching, the fluid-fronts almost match the observed ones. Although the front positions are relatively more complex at day 5400 than at day 2700, a very good result still can be obtained. When the cRRDB-U-Net surrogate models are trained with a small sample size, e.g., $N_s = 300$ or $N_s = 500$, several fluid-front positions near injectors I1, I2 and I6 (the lower left part of the reservoir model) are not well matched, and the permeability around these positions cannot be significantly improved. In contrast, the fluid-front positions near the injector I9, I4 and I7 (the upper right part of the reservoir) are matched very well, which leads to good calibrations of the reservoir models around this area. Fig. 24(b) shows the average effective number of data as a function of iterations. Comparing to the above 2D synthetic model, the use of distance-based parameterization obtains a much larger reduction of effective data, e.g., from original 157,440 to 17,095 after history matching, which quantifies the reduction of model uncertainty. This result also demonstrates a high scalability of our proposed hybrid workflow to practical applications.

Fig. 25 shows the posterior ensemble mean and standard deviation corresponding to different training sample sizes N_s=100, 300, 500, 800 and 1000, respectively. It clearly can be seen that the posterior ensemble means are very close to the true model. The geological parameters, e.g. highly permeable zones, are successfully reconstructed using even a small number of training samples, e.g. N_s = 300. The true model is gradually reproduced as the number of training samples increases. The significant reduction in the standard deviation of the spread of parameter estimates, e.g., from approximately 2.5 in the prior to 0.5 in the posterior ensemble, further indicates a high accuracy of the history matching result. In order to further verify the reliability of the posterior models from our proposed surrogate-based history matching approach, we compare predictions of the well



(a)

(b)

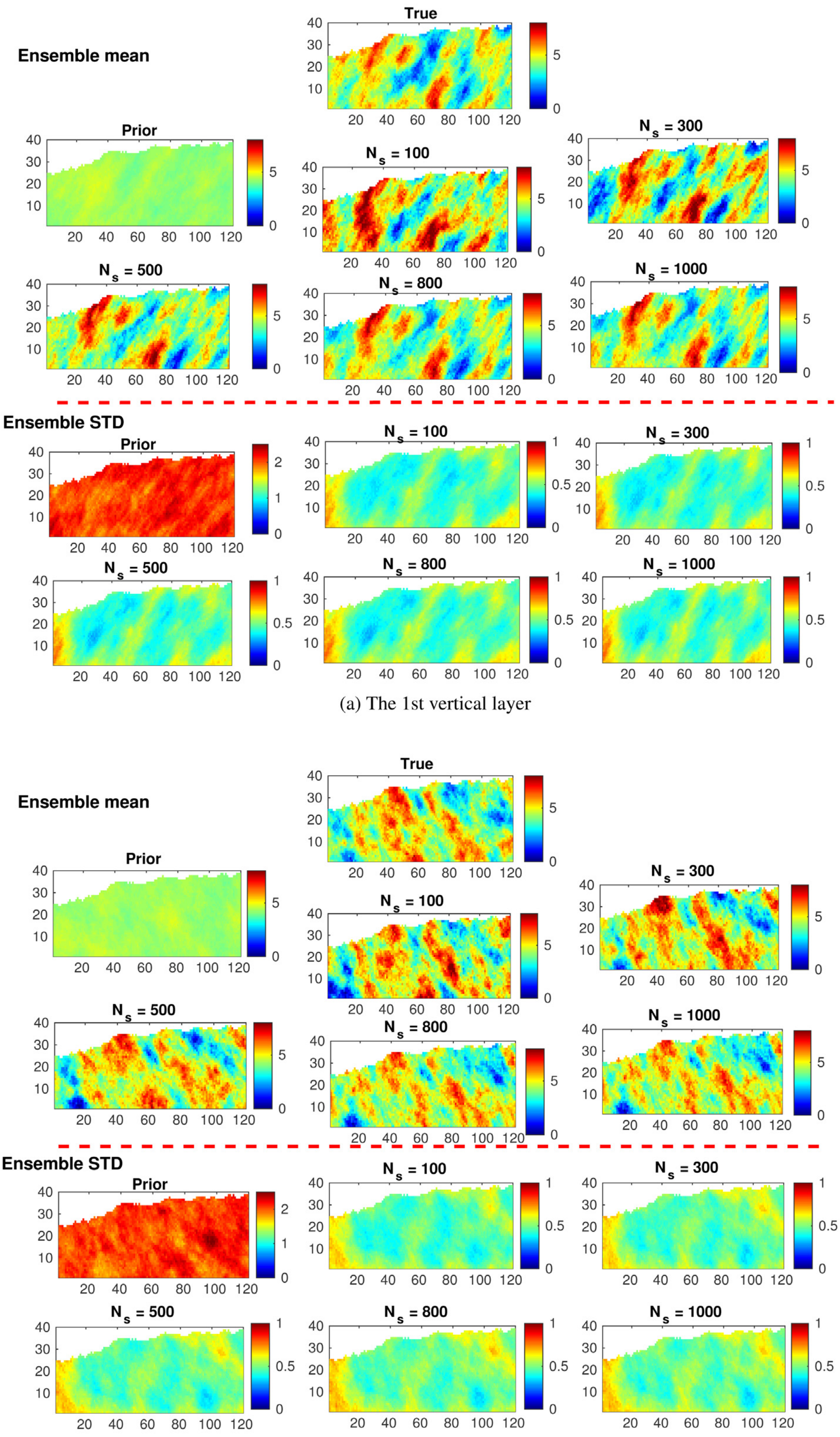
Fig. 24. Characteristics of the ensemble of innovations used by LHDC for 3D model. (a) Simulated water-fronts after 2700 days and 5400 days in one posterior model realization before and after history matching using the cRRDB-U-Net surrogate model. The observed and simulated fronts are denoted in blue and red lines, respectively. This figure displays the water-fronts of the 11th vertical layer. (b) The over-all effective number of data during the iteration process.

water injection rate (WWIR) and well water-cut (WWCT) for nine injectors and nine producers, see Fig. 26. These quantities were not used in the history matching procedure and can therefore be viewed as independent validation data. In order to generate the predictions, the permeability estimates obtained with the surrogate-assisted hybrid workflow are used as input for simulations with the HF model simulator. Although the predictions of the initial models are significantly different from the true model, after history matching to the binary image data, the mean and spread in rate predictions from the updated models are much more consistent with the rates generated with the HF truth model.

4.7. Computational cost

In this section, we will briefly discuss the aspects of computation cost of the proposed workflow for parameter estimation using cRRDB-U-Net surrogate modeling. We denote the cost of a high-fidelity model simulation and a surrogate model simulation as C_{HFM} and C_S respectively, the number of training data N_s, the ensemble size N_e and the number of iterations N_a, the total computational cost of the overall workflow based on HF model simulations only is

$$T_{HFM} = N_a \cdot N_e \cdot C_{HFM}. \tag{16}$$



(a) The 1st vertical layer

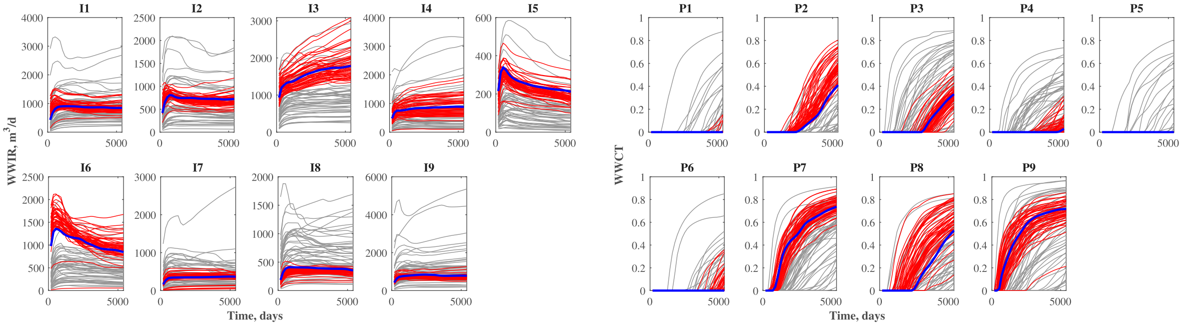
(b) The 11th vertical layer

Fig. 25. Comparison of the updated ensemble posterior realizations using cRDDB-U-Net surrogate model with respect to the number of training samples $N_s = 100$, 300, 500, 800 and 1000, respectively. (a) The 1st horizontal layer; (b) The 11th horizontal layer.

For the surrogate-assisted hybrid workflow, the total cost is

$$T_S = N_s \cdot C_{HFM} + C_{train} + N_a \cdot N_e \cdot C_S. \tag{17}$$

The runtimes of a single HF model simulation for the 2D model of Case 1 and the 3D model of Case 2 are about 5.0 s and 250.0 s respectively on a machine with i5-4690 Intel CPUs (4 cores, 3.5 GHz) and 24 GB memory. In comparison, the runtime of the cRRDB-U-Net surrogate model is about 0.1 s for both cases, so it is a factor of $10^2 - -10^3$ smaller than the runtime for the HF model. The cRRDB-U-Net surrogate model is trained on a NVIDIA



(a) Well water injection rate (WWIR)

(b) Well watercut (WWCT)

Fig. 26. Predictions of water injection rate and watercut for the nine injectors and nine producers. The gray lines — initial models, blue lines — reference model, red lines — updated models from the cRRDB-U-Net surrogate trained with $N_s = 1000$ samples.

GeForce GTX 745 GPU and separately takes about 25 min and 31.4 h using 1000 training samples for the 2-D and 3-D case respectively, which are equivalent to run approximately 300 and 450 HFM simulations, respectively. For the comparison, we set $C_S \approx 10^{-3} C_{HFM}$ and $C_{train} \approx 10^{-1} \cdot N_s \cdot C_{HFM}$ and determine cost reduction T_S/T_{HFM} expressed in terms of HF model simulations as a function of the parameters N_s, N_e and N_a,

$$\frac{T_S}{T_{HFM}} = \frac{N_s + 10^{-1} N_s + 10^{-3} N_a N_e}{N_a N_e} \approx \frac{Ns}{N_a N_e}. \tag{18}$$

For typical values $N_s = 1000$, $N_e = 200$ and $N_a = 10$, the cost ratio is about 0.5. For an ensemble size of 500 this is about 0.2, and if the training size can be reduced to 500 as well, the ratio becomes 0.1. We note that in the surrogate-assisted workflow the number of iterations could be increased significantly without incurring higher computational cost since the cost of simulating the proxy is almost negligible relative to simulating the HF model. We may expect that a higher number of iterations can contribute to improved performance of the ES-MDA algorithm to a certain extend.

5. Summary and conclusions

In this study, a hybrid model parameter estimation workflow is presented for use with image data. We assume that the relevant information in the images, or in differences between images, can be represented by the result of an image segmentation process that classifies features in the images into binary categories. We propose a modification of the well-known Residual U-Net (R-U-Net), referred to conditional residual-in-residual dense block U-Net (cRRDB-U-Net), comprised of encoding, transition and decoding units. In order to capture the time-varying process, we concatenate the time values to the highly compressed features produced by the encoding unit. This deep neural network is trained with data from a set of high-fidelity model simulations where random realizations of parameter grids are provided as input. The simulated continuous high-fidelity model output is processed into binary images and input to a binary cross entropy loss function. We augment the loss function with a mean square error contribution based on the continuous simulation output, resulting in a 2-stage minimization procedure. The trained network is subsequently used as a surrogate model to replace the high-fidelity model simulations in a workflow to estimate the values of uncertain gridblock parameters. An image-oriented distance parameterization is used to quantify the differences between the binary images predicted by the network surrogate and the observed binary images. The differences are minimized using an iterative Ensemble Smoother algorithm.

The workflow is demonstrated on 2D and 3D examples representing typical problems encountered when modeling subsurface flow dynamics, especially two-phase fluid displacement problems such as CO_2 storage in aquifers of water injection in oil reservoirs. We note that while the 2D example is smaller, the parameter field in this example, permeability in a geological reservoir, has itself a binary character, representing the high-permeability associated with coarse-sand channel deposits in a low-permeability background of clay deposits.

From the results of the experiments we conclude that in both examples the workflow is able to reconstruct the true permeability field with an increasing accuracy when the number of training samples is increased. The quality of the estimated parameter fields is verified independently by providing them as input to high-fidelity model simulations to predict phase rates at well locations, showing a significant improvement in the match to the time series simulated with the true permeability field. The use of cRRDB-U-Net as a surrogate model reduces the overall computational time by a factor 5 for the problems investigate in this study. Cost reductions up to 90% relative to full-model workflows can be expected in realistic applications where large numbers of parameter realizations are estimated. The computational cost associated with simulating the surrogate model only increases slowly with the model size.

In this paper we have kept the surrogate model fixed after initial training, which requires an a prior choice for the dimension of the training set. The incorporation of additional data in the data assimilation workflow has the potential to inform further improvements to the surrogate model and thereby generate more accurate solutions. Computational savings could potentially be obtained by initial training based on a small training set, and adaptive enrichment of the training set and refinement of the surrogate model during the data assimilation workflow.

The integration of an image pre-processing procedure and deep-learning surrogate modeling techniques could motivate many interesting applications in a variety of engineering disciplines, especially for imaging-type monitoring problems where large-scale physical models are highly complex and computationally intensive. The proposed deep-learning hybrid workflow offers an alternative avenue to make reliable predictions from a reduced set of physics-based models. We note that similar problems are encountered in for example medical imaging, satellite imagery (remote sensing) and other geoscience domains. Overall, an extension of our proposed hybrid workflow to these areas should be straightforward with only several domain-specific adaptations.

CRediT authorship contribution statement

Cong Xiao: Conceptualization, Methodology, Software, Writing - original draft . **Olwijn Leeuwenburgh:** Geological model generation, Model design, Writing - original draft, Writing - reviewing and editing. **Hai-Xiang Lin:** Investigation, Supervision, Writing - original draft, Writing - reviewing and editing. **Arnold Heemink:** Investigation, Supervision, Writing - original draft, Writing - reviewing and editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The first author would like to thank the China Scholarship Council (CSC) for providing research funding. Additional Computing resources were provided by the Mathematics Physics Group, Department of Applied Mathematics at Delft University of Technology. The use of open source codes OPM-*flow* (https://opm-project.org/) is gratefully acknowledged. We are grateful to Yanhui Zhang from KAUST, Saudi Arabia, for useful discussions on image-oriented distance parameterization approach. The code and data can be provided upon the request.

References

[1] Paul Suetens, Fundamentals of Medical Imaging, Cambridge university press, 2017.

[2] T. Lopez, A. Al Bitar, Sylvain Biancamaria, A. Güntner, A. Jäggi, On the use of satellite remote sensing to detect floods and droughts at large scales, Surv. Geophys. (2020) 1–27.

[3] Parth Sarathi Mahapatra, Siva Praveen Puppala, Bhupesh Adhikary, Kundan L. Shrestha, Durga Prasad Dawadi, Shankar Prasad Paudel, Arnico K. Panday, Air quality trends of the Kathmandu Valley: A satellite, observation and modeling perspective, Atmos. Environ. 201 (2019) 334–347.

[4] Jin-Feng Ma, Lin Li, Hao-Fan Wang, Ming-You Tan, Shi-Ling Cui, Yun-Yin Zhang, Zhi-Peng Qu, Ling-Yun Jia, Shu-Hai Zhang, Geophysical monitoring technology for CO 2 sequestration, Appl. Geophys. 13 (2) (2016) 288–306.

[5] Guangliang Fu, Hai-Xiang Lin, Arnold Heemink, Arjo Segers, Fred Prata, Sha Lu, Satellite data assimilation to improve forecasts of volcanic ash concentrations, Atmos. Chem. Phys. Discuss. (2016) 1–22.

[6] Jianbing Jin, Arjo Segers, Arnold Heemink, Mayumi Yoshida, Wei Han, Hai-Xiang Lin, Dust emission inversion using himawari-8 AODs over east Asia: An extreme dust event in may 2017, J. Adv. Modelling Earth Syst. 11 (2) (2019) 446–467.

[7] Mario Rüttgers, Sangseung Lee, Soohwan Jeon, Donghyun You, Prediction of a typhoon track using a generative adversarial network and satellite images, Sci. Rep. 9 (1) (2019) 1–15.

[8] Robert Trautz, Thomas Daley, Douglas Miller, Michele Robertson, George Koperna Jr, David Riestenberg, Geophysical monitoring using active seismic techniques at the Citronelle Alabama CO2 storage demonstration site, Int. J. Greenh. Gas Control 99 (2020) 103084.

[9] Ludmila Adam, Kasper van Wijk, Thomas Otheim, Michael Batzle, Changes in elastic wave velocity and rock microstructure due to basalt-CO2-water reactions, J. Geophys. Res. 118 (8) (2013) 4039–4047.

[10] A.A. Emerick, A.C. Reynolds, History matching time-lapse seismic data using the ensemble Kalman filter with multiple data assimilations, Comput. Geosci. 16 (3) (2012) 639–659.

[11] T. Mannseth, K. Fossum, Assimilating spatially dense data for subsurface applications—balancing information and degrees of freedom, Comput. Geosci. 22 (5) (2018) 1323–1349.

[12] T. Bhakta, X. Luo, G. Naevdal, Correlation-based adaptive localization with applications to ensemble-based 4D-seismic history matching, SPE J. 23 (02) (2018) 396–427.

[13] M. Liu, D. Grana, Ensemble-based seismic history matching with data reparameterization using convolutional autoencoder, in: SEG Technical Program Expanded Abstracts 2018, Society of Exploration Geophysicists, 2018, pp. 3156–3160.

[14] O. Leeuwenburgh, R. Arts, Distance parameterization for efficient seismic history matching with the ensemble Kalman Filter, Comput. Geosci. 18 (3–4) (2014) 535–548.

[15] Y. Zhang, O. Leeuwenburgh, Image-oriented distance parameterization for ensemble-based seismic history matching, Comput. Geosci. 21 (4) (2017) 713–731.

[16] Mario Trani, Rob Arts, Olwijn Leeuwenburgh, et al., Seismic history matching of fluid fronts using the ensemble Kalman filter, SPE J. 18 (01) (2012) 159–171.

[17] P. Bergey, A. Abadpour, R. Piasecki, 4D seismic history matching with ensemble kalman filter-assimilation on hausdorff distance to saturation front, in: SPE Reservoir Simulation Symposium, Society of Petroleum Engineers, 2013.

[18] S. Da Veiga, E. Tillier, R. Derfoul, Appropriate formulation of the objective function for the history matching of seismic attributes, Comput. Geosci. 51 (2013) 64–73.

[19] O. Gosselin, S.I. Aanonsen, I. Aavatsmark, A. Cominelli, R. Gonard, M. Kolasinski, F. Ferdinandi, L. Kovacic, K. Neylon, et al., History matching using time-lapse seismic (HUTS), in: SPE Annual Technical Conference and Exhibition, Society of Petroleum Engineers, 2003.

[20] Pinggang Zhang, Gillian E. Pickup, Michael A. Christie, et al., A new practical method for upscaling in highly heterogeneous reservoir models, SPE J. 13 (01) (2008) 68–76.

[21] Cong Xiao, Olwijn Leeuwenburgh, Hai Xiang Lin, Arnold Heemink, Non-intrusive subdomain POD-TPWL for reservoir history matching, Comput. Geosci. 23 (3) (2019) 537–565.

[22] R.K. Tripathy, I. Bilionis, Deep UQ: Learning deep neural network surrogate models for high dimensional uncertainty quantification, J. Comput. Phys. 375 (2018) S0021999118305655.

[23] J. Nagoor Kani, Ahmed H. Elsheikh, Reduced order modeling of subsurface multiphase flow models using deep residual recurrent neural networks, Transp. Porous Media (2018).

[24] N. Zabaras, X. Shi, S. Mo, J. Wu, Deep autoregressive neural networks for high-dimensional inverse problems in groundwater contaminant source identification, 2018.

[25] Y. Zhu, N. Zabaras, X. Shi, S. Mo, J. Wu, Deep convolutional encoder-decoder networks for uncertainty quantification of dynamic multiphase flow in heterogeneous media, Water Resour. Res. 55 (1) (2019).

[26] A.Y. Sun, Z. Zhong, H. Jeong, Predicting CO2 plume migration in heterogeneous formations using conditional deep convolutional generative adversarial network, Water Resour. Res. (2019).

[27] Y. Zhu, N. Zabaras, Bayesian deep convolutional encoder-decoder networks for surrogate modeling and uncertainty quantification, J. Comput. Phys. 366 (2018).

[28] S.J. Kim, D. Kim, K. Lee, Y.J. Heo, W.K. Chung, Super-high-purity seed sorter using low-latency image-recognition based on deep learning, IEEE Robotics Autom. Lett. 3 (4) (2018) 1.

[29] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Gregory S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian J. Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Józefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Gordon Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul A. Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda B. Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, Xiaoqiang Zheng, Tensorflow: Large-scale machine learning on heterogeneous distributed systems, 2016, CoRR abs/1603.04467.

[30] S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Paszke, A. Desmaison, Pytorch: An imperative style, high-performance deep learning library, in: Advances in Neural Information Processing Systems, 2019, pp. 8024–8035.

[31] Y. Liu, Z.L. Jin, L.J. Durlofsky, Deep-learning-based reduced-order modeling for subsurface flow simulation, 2019, arXiv preprint arXiv:1906.03729.

[32] Dawid Połap, Gautam Srivastava, Neural image reconstruction using a heuristic validation mechanism, Neural Comput. Appl. (2020) 1–11.

[33] Dawid Polap, Marta Wlodarczyk-Sielicka, Classification of non-conventional ships using a neural bag-of-words mechanism, Sensors 20 (6) (2020) 1608.

[34] N.R. Pal, S.K. Pal, A review on image segmentation techniques, Pattern Recognit. 26 (9) (1993) 1277–1294.

[35] C. Xu, D.L. Pham, J.L. Prince, Current methods in medical image segmentation, Ann. Rev. Biomed. Eng. 2 (1) (2000) 315–337.

[36] Y.U. Haichao, D. Gao, Y. Wang, J. Wang, Image segmentation and object detection using fully convolutional neural network, 2019, US Patent App. 10/304, 193.

[37] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, Kyoung Mu Lee, Enhanced deep residual networks for single image super-resolution, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2017.

[38] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, Yun Fu, Residual dense network for image restoration, 2018, ArXiv.

[39] Shaoxing Mo, Nicholas Zabaras, Xiaoqing Shi, Jichun Wu, Integration of adversarial autoencoders with residual dense convolutional networks for estimation of non-gaussian hydraulic conductivities, Water Resour. Res. n/a(n/a) e2019WR026082.

[40] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, Yun Fu, Residual dense network for image super-resolution, 2018, http://arxiv.org/abs/1802.08797.

[41] P. Fischer, O. Ronneberger, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2015, pp. 234–241.

[42] J.N. Carter, K.D. Stephen, R.W. Zimmerman, A. Skorstad, T. Manzocchi, J.D. Matthews, J.A. Howell, Assessing the effect of geological uncertainty on recovery estimates in shallow-marine reservoirs: the application of reservoir engineering to the SAIGUP project, Petrol. Geosci. 14 (1) (2008) 35–44.

[43] A.C. Reynolds, D.S. Oliver, N. Liu, Inverse Theory for Petroleum Reservoir Characterization and History Matching, Cambridge University Press, 2008.

[44] R. Chassagne, D. Obidegwu, C. MacBeth, Seismic assisted history matching using binary image matching, in: EUROPEC 2015, Society of Petroleum Engineers, 2015.

[45] Y. Liu, M. Tang, L.J. Durlofsky, A deep-learning-based surrogate model for data assimilation in dynamic subsurface flow problems, 2019, arXiv preprint arXiv:1908.05823.

[46] X. Zhang, S. Ren, K. He, J. Sun, Identity mappings in deep residual networks, in: European Conference on Computer Vision, 2016.

[47] Rajeshwar Dass, Swapna Devi, Image segmentation techniques 1, Citeseer, 2012.

[48] Sergii Mashtalir, Volodymyr Mashtalir, Spatio-temporal video segmentation, in: Advances in Spatio-Temporal Segmentation of Visual Data, Springer, 2020, pp. 161–210.

[49] Jiaxu Miao, Yunchao Wei, Yi Yang, Memory aggregation networks for efficient interactive video object segmentation, 2020, http://arxiv.org/abs/2003.13246.

[50] Zongxin Yang, Yunchao Wei, Yi Yang, Collaborative video object segmentation by foreground-background integration, 2020, http://arxiv.org/abs/2003.08333.

[51] Z. Liu, L. Van Der Maaten, G. Huang, K.Q. Weinberger, Densely connected convolutional networks, 2016.

[52] Hai X. Vo, Louis J. Durlofsky, A new differentiable parameterization based on principal component analysis for the low-dimensional representation of complex geological models, Math. Geosci. 46 (7) (2014) 775–813.

[53] Atgeirr Flø Rasmussen, Tor Harald Sandve, Kai Bao, Andreas Lauser, Joakim Hove, Bård Skaflestad, Robert Klöfkorn, Markus Blatt, Alf Birger Rustad, Ove Sævareid, et al., The open porous media flow reservoir simulator, 2019, arXiv preprint arXiv:1910.06059.

[54] GSLIB: Geostatistical software library and user's guide, Technometrics (1995).