



Use of multifidelity training data and transfer learning for efficient construction of subsurface flow surrogate models

Su Jiang^{*}, Louis J. Durlofsky

Department of Energy Science and Engineering, Stanford University, Stanford, CA 94305, USA

ARTICLE INFO

Article history:

Received 23 April 2022

Received in revised form 26 September 2022

Accepted 16 November 2022

Available online 24 November 2022

Dataset link: <https://github.com/sujiang94/surrogate-transfer-learning>

Keywords:

Surrogate model

Transfer learning

Multifidelity data

Reservoir simulation

History matching

Data assimilation

ABSTRACT

In subsurface flow settings, data assimilation/history matching presents computational challenges because many high-fidelity models must be simulated. Various deep-learning-based surrogate modeling techniques have been developed to reduce the simulation costs associated with these applications. However, to construct data-driven surrogate models, several thousand high-fidelity simulation runs may be required to provide training samples, and these computations can make training prohibitively expensive. To address this issue, in this work we present a framework where most of the training simulations are performed on coarsened (low-fidelity) geomodels. These models are constructed using a flow-based upscaling method. The framework entails the use of a transfer-learning procedure, incorporated within an existing recurrent residual U-Net architecture, in which network training is accomplished in three steps. In the first step, where the bulk of the training is performed, only low-fidelity simulation results are used. The second and third steps, in which the output layer is trained and the overall network is fine-tuned, require a relatively small number of high-fidelity simulations. Here we use 2500 low-fidelity runs and 200 high-fidelity runs, which leads to about a 90% reduction in training simulation costs. The method is applied for two-phase subsurface flow in 3D channelized systems, with flow driven by wells. The surrogate model trained with multifidelity data is shown to be nearly as accurate as a reference surrogate trained with only high-fidelity data in predicting dynamic pressure and saturation fields in new geomodels. Importantly, the network provides results that are significantly more accurate than the low-fidelity simulations used for most of the training. The multifidelity surrogate is also applied for history matching using an ensemble-based procedure, where accuracy relative to reference results is again demonstrated.

© 2022 Elsevier Inc. All rights reserved.

1. Introduction

Accurate subsurface flow modeling is required to manage the production of energy resources and the storage of CO₂. The key input in these frameworks is the geological model. Before the geomodel can be used for predictions or optimization, it must be calibrated such that simulation predictions match historical observations. The computations required in this history matching step (also referred to as data assimilation) can be very demanding, however, because thousands to millions of high-fidelity forward simulations must be performed. Surrogate modeling techniques can be very useful in this setting. Data-

^{*} Corresponding author.

E-mail address: sujiang@stanford.edu (S. Jiang).

driven surrogate models represent an effective family of approaches for treating complex geological systems, though existing procedures require several thousand high-fidelity simulations to provide training samples. The computation associated with these training runs is one of the most expensive components of these procedures and may limit their application.

In recent work, a data-driven surrogate model for subsurface flow that employs residual U-Nets within a recurrent neural network (the overall model is referred to as recurrent R-U-Net) was developed and applied for history matching problems in oil production and CO₂ storage applications [1–4]. In the examples considered, which included up to 128,000 finite volume cells (with two unknowns per cell), 2000–3000 high-fidelity simulations were required as training samples to achieve accurate surrogate model predictions. Our goal in this work is to use high and low fidelity (referred to as multifidelity) simulation data in combination with transfer learning to accelerate surrogate model construction. This strategy enables us to perform ~90% of the training simulations using low-fidelity models. The low-fidelity geomodels required for these simulations are constructed from high-fidelity geomodels using a flow-based upscaling procedure. A relatively small number of computationally demanding fine-scale simulations are required to fine-tune the surrogate model.

Many approaches have been developed for the construction of surrogate flow models using deep neural networks. We classify these approaches into two general (though overlapping) categories – physics-informed neural networks (PINNs) and data-driven deep-learning-based surrogate models. The training process for PINNs is conducted by minimizing the (physical) loss from the residual of the governing partial differential equations along with losses associated with boundary and initial conditions. The minimization process can be accomplished, in theory, without running any forward simulations, though simulation data are also commonly used. Raissi et al. [5] first proposed the idea of PINNs to approximate multiple different governing equations. Theory-guided neural networks (TgNNs), which are also physics-informed treatments, were developed by Wang et al. [6,7,8] and have been used to approximate 2D single-phase flow and other problems. PINN methods incorporating physical loss and simulation data mismatch have been successfully applied by He et al. [9] and Tartakovsky et al. [10,11] for single-phase flow and transport problems. Extending these methods to realistic 3D multiphase flow problems, or to multiphysics settings, will require additional development.

Data-driven deep-learning-based surrogate models are trained with labeled data. After training, these models provide a mapping from simulation input to output. These procedures may require a substantial amount of training data, though they have been applied in complicated flow settings. The spatial correlations are generally captured through use of convolutional neural networks (CNNs), and time evolution is modeled using approaches for the treatment of time series. Zhu and Zabarar [12] introduced a convolutional encoder-decoder framework to predict single-phase flow solutions. Mo et al. [13,14,15] further developed the framework through application of autoregressive strategies to predict temporal evolution and simulation output over long time frames. Applications of this framework include 2D CO₂ storage and 2D and 3D contaminant transport in channelized systems.

Tang et al. [1] developed a recurrent residual U-Net (R-U-Net) model for 2D oil-water problems with flow driven by wells. A residual U-Net and a convolutional long short-term memory (convLSTM) recurrent network were integrated to capture pressure and saturation fields at a number of discrete time steps. Tang et al. [2,3] extended the method to treat 3D oil-water systems with channelized geomodels. This procedure was later modified to model 3D CO₂ storage problems involving coupled flow and geomechanics (multiphysics) problems [4]. Wen et al. [16] introduced a residual U-Net surrogate model for predicting plume migration in CO₂ storage settings. This model, which was developed to be predictive over a wide range of flow and engineering parameters, was trained using 20,000 simulation runs. Transfer learning, with a few hundred additional training samples, was applied to fine-tune the model when new injection settings were considered.

From the above discussion we see that, with sufficient training, data-driven surrogate models can predict complicated subsurface flow behavior, but the large number of high-fidelity simulation runs required for training may limit their overall applicability. The use of multifidelity data (from high-fidelity and low-fidelity simulation output) for surrogate model construction can mitigate this problem. Geneva and Zabarar [17] developed a conditional invertible neural network to generate the probability distribution for high-fidelity turbulent flows conditioned to low-fidelity data. Physically accurate turbulent flows and statistics were constructed with a generative method trained with a reduced data set. Meng and Karniadakis [18] utilized multifidelity data to train a composite physics-informed neural network for unsaturated flow and reactive transport. Three networks were coupled to capture the linear and nonlinear correlations between the high and low-fidelity data. High accuracy was achieved through use of a small set of high-fidelity training data.

Multifidelity data can be combined with transfer learning to construct surrogate models that provide high-fidelity predictions. This entails first training the surrogate model with low-fidelity simulation results and then using transfer learning to fine-tune the model with a small number of high-fidelity samples. De et al. [19] applied transfer learning and developed a bi-fidelity-weighted learning for uncertainty propagation in engineering problems. For the process of bi-fidelity-weighted learning, a Gaussian process model was trained with a few high-fidelity samples. This provided synthetic data samples that were used to update the surrogate model trained with low-fidelity data. The use of a Gaussian process model may affect prediction accuracy for more complicated processes. Song and Tartakovsky [20] applied transfer learning to build a CNN-based surrogate model to solve two-phase flow problems in 2D Gaussian geomodels. The high-fidelity models had four times as many grid blocks as the low-fidelity models. The optimal ratio of low-fidelity to high-fidelity training runs was found to be around five.

In this work, we develop a transfer-learning-based surrogate model that builds on the existing recurrent R-U-Net framework. The target application here is two-phase (well-driven) subsurface flow in 3D channelized systems. A global single-phase transmissibility upscaling procedure is applied to generate coarse-scale models from high-fidelity geomodels.

Two-phase flow simulations performed on these models provide the low-fidelity training samples. In our approach, the high-fidelity geomodel is always used as network input, and the decoder output is also always at high-fidelity. In the first training step, 2500 low-fidelity simulation results are used, and the high-fidelity decoder output is fed to a separate low-fidelity output layer. In subsequent training steps, a small number (~ 200) of high-fidelity simulation results are used to train a high-fidelity output layer, which enables the overall network to provide high-fidelity pressure and saturation fields at a specified set of time steps. The surrogate model is then applied for history matching with an ESMDA (ensemble smoother with multiple data assimilation) procedure to generate posterior geomodels. The surrogate model predictions are compared to reference fine-scale simulations to assess the performance of the overall procedure.

Many aspects of our procedure differ from those used in previous investigations. The detailed transfer learning and three-stage training treatments applied here, and the way in which we use high- and low-fidelity data to train the recurrent R-U-Net, differ from the approaches in earlier studies. We quantify the impact of the number of high-fidelity samples, evaluate the performance of different training strategies, and assess the effect of the weighting of well-block data on prediction accuracy. Our findings in these assessments may be useful for a range of related subsurface flow problems. In addition, in this work we consider two-phase flow in complicated 3D channelized systems, with flow driven by wells operating at specified pressures. This appears to be a new setting for multifidelity transfer learning. Our use of an accurate global flow-based upscaling approach, rather than a simple local upscaling procedure or heuristic averaging to generate low-fidelity models, is also new in the context of transfer learning.

This paper proceeds as follows. In Section 2, we present the governing equations for the two-phase flow problem and describe the upscaling procedure, which enables the generation of multifidelity data. Then, in Section 3, we discuss the recurrent R-U-Net surrogate model and the use of transfer learning with multifidelity data for training. Surrogate model results for oil-water flow in 3D channelized systems are presented in Section 4. Comparisons between a surrogate model trained with multifidelity data, a surrogate model trained with only high-fidelity data, and reference numerical simulation results are provided. In Section 5, the transfer-learning-based recurrent R-U-Net is combined with ESMDA to generate posterior models and predictions. Conclusions and suggestions for future work in this area appear in Section 6.

2. Governing equations and upscaling procedure

In this section, we present the governing equations for two-phase subsurface flow problems and describe the upscaling procedure used to provide low-fidelity geomodels.

2.1. Governing equations

The two-phase immiscible subsurface flow equations considered in this work are applicable for oil reservoir simulation (e.g., oil production via water injection) and to environmental remediation modeling involving water and a nonaqueous-phase liquid (NAPL) contaminant. The governing equations are

$$\nabla \cdot (\rho_j \mathbf{u}_j) + q_j + \frac{\partial}{\partial t}(\phi \rho_j S_j) = 0, \quad j = o, w, \quad (1)$$

where j denotes phase, with o indicating oil phase and w water phase, ρ_j is the density of phase j , \mathbf{u}_j is the Darcy velocity of phase j , q_j denotes the source/sink term, ϕ is porosity and S_j is the phase saturation (volume fraction). Darcy velocity \mathbf{u}_j is given by

$$\mathbf{u}_j = -\frac{\mathbf{k}k_{rj}(S_j)}{\mu_j(p_j)}(\nabla p_j - \rho_j g \nabla z), \quad j = o, w, \quad (2)$$

where \mathbf{k} represents the absolute permeability tensor, k_{rj} is the relative permeability of phase j , μ_j is the viscosity of phase j , p_j is the pressure of phase j , g is gravitational acceleration, and z is depth. The relative permeability k_{rj} and viscosity μ_j are nonlinear functions of saturation and pressure, respectively. As is common in reservoir-scale simulations, we neglect capillary pressure in this work, so we have $p = p_w = p_o$.

In the context of oil reservoir simulation, which is the setting considered here, the governing equations (1) and (2) are usually discretized using finite volume methods. Newton's method is applied to solve the discrete nonlinear equations. In this work, we use Stanford's Automatic Differentiation General Purpose Research Simulator ADGPRS [21] for both high-fidelity (HF) and low-fidelity (LF) simulation.

High-fidelity geomodels are denoted by $\mathbf{m}^h \in \mathbb{R}^{n^h \times 1}$, where $n^h = n_x^h \times n_y^h \times n_z^h$ is the number of grid blocks in the HF model, with n_x^h , n_y^h and n_z^h the number of blocks in each coordinate direction (superscripts h and l indicate HF and LF). Note that here the geological parameters are represented in terms of a single quantity in each block, which defines (isotropic) permeability and porosity. The governing equations are solved, with \mathbf{m}^h as input, to provide HF solutions for the state variables. In oil-water problems these are pressure $\mathbf{P}^h \in \mathbb{R}^{n^h \times n_{ts}}$ and saturation $\mathbf{S}^h \in \mathbb{R}^{n^h \times n_{ts}}$ in every grid block at n_{ts} simulation time steps. The simulation process can be represented as

$$\mathbf{x}^h = [\mathbf{P}^h, \mathbf{S}^h] = g(\mathbf{m}^h), \quad (3)$$

where g represents the forward numerical simulation process and \mathbf{x}^h is the simulation output.

Well injection/production rates are primary quantities of interest in subsurface flow problems. These rates are computed through application of

$$(q_j^w)_i = WI_i \left(\frac{k_{rj} \rho_j}{\mu_j} \right)_i (p_i - p_i^w), \quad (4)$$

where $(q_j^w)_i$ denotes the source/sink mass flow rate for phase j in well block i , p_i denotes the well-block pressure, p_i^w denotes the wellbore pressure evaluated at the center of well block i , and WI_i is the well index, computed as [22]

$$WI_i = \frac{2\pi k_i \Delta z}{\ln \frac{r_0}{r_w}}, \quad (5)$$

where k_i is the well block permeability, Δz is the thickness of the well block, and $r_0 = 0.2\Delta x$ for isotropic permeability and $\Delta x = \Delta y$, where Δx and Δy denote the dimensions of the well block. Eq. (5) is for a vertical well that fully penetrates the grid block. The wellbore pressure p_i^w , which varies with depth, is given by

$$p_{i+1}^w = p_i^w + (\rho_{i,i+1} g) \Delta z_{i,i+1}. \quad (6)$$

Here $\rho_{i,i+1}$ is the average fluid density between well blocks i and $i+1$ and $\Delta z_{i,i+1}$ denotes the difference in depth. Bottom-hole pressure (BHP), which will be specified in our simulations, corresponds to the wellbore pressure at the uppermost perforation.

2.2. Upscaling procedure

A flow-based, single-phase global transmissibility upscaling procedure [23–25] is applied to generate coarse-scale (LF) models from fine-scale (HF) geomodels. The specific implementation used in this work is that of Crain [25]. This discussion follows [26], where upscaled models of the type used here were applied with error models for history matching in data space.

The first step in the upscaling process is the solution of the global single-phase steady-state pressure equation over the HF geomodel, i.e.,

$$\nabla \cdot (k \nabla p) = q, \quad (7)$$

where the source term q , which corresponds to flow driven by wells, is expressed using the single-phase-flow analog of Eq. (4) (note Eq. (7) is written in dimensionless form). From the HF pressure solution, the block-to-block volumetric flow rate f_α^h is computed via

$$f_\alpha^h = T_\alpha (p_{\alpha^+}^h - p_{\alpha^-}^h), \quad (8)$$

where T_α is the transmissibility, α denotes the interface of two fine-scale grid cells, and $p_{\alpha^\pm}^h$ denotes the pressures of the grid blocks on either side of interface α . Transmissibility, which is essentially the numerical analog of permeability, is given by $T_\alpha = k_\alpha \Delta y \Delta z / \Delta x$, for cells connected in the x -direction, where k_α is the harmonic average of the permeabilities of the blocks on either side of interface α . Analogous expressions define transmissibilities for cells connected in the y and z -directions.

Given the HF pressure solution and flow rates through all HF block-to-block interfaces (Eq. (8)), the coarse-scale (LF) transmissibilities and well indices can be constructed. Let β denote the interface between two LF grid blocks. The LF block pressures $p_{\beta^\pm}^l$ for block β^+ and β^- are estimated as the volume average ($\langle p \rangle_{\beta^\pm}$) of the fine-scale pressures p^h that lie within the range of coarse blocks β^+ and β^- . The flow rate f_β^l through LF interface β is estimated to be the sum of the fine-scale flow rates f_α^h over the range of interface β . In analogy to Eq. (8), the upscaled transmissibility T_β^* , over the LF interface β , is then given by

$$T_\beta^* = \frac{f_\beta^l}{p_{\beta^-}^l - p_{\beta^+}^l} \approx \frac{\sum_{\alpha \in \beta} f_\alpha^h}{\langle p \rangle_{\beta^-} - \langle p \rangle_{\beta^+}}. \quad (9)$$

The above estimate of LF transmissibility, which can be viewed as a volume average of the HF flow result, has been shown to provide accurate LF transmissibilities. It can be improved by iterating on the LF model to force very close agreement between $\sum_{\alpha \in \beta} f_\alpha^h$ and f_β^l [24]. Methods for handling anomalous T_β^* , which can occur when f_β^l and/or $(p_{\beta^-}^l - p_{\beta^+}^l)$ are very small, are discussed by Crain [25].

Additional treatments are required for coarse blocks containing wells. For these blocks, we compute an upscaled well index WI^* (the well index in the HF model is defined in Eq. (5)). This upscaled well index provides a transmissibility

relating well injection/production rate to the difference in pressure between the wellbore and the well block. In analogy to Eq. (9), WI^* is given by

$$WI_i^* = \frac{f_i^{w,l}}{p_i^l - p_i^w} \approx \frac{\sum_{k \in i} f_k^{w,h}}{\langle p \rangle_i - p_i^w}, \quad (10)$$

where $f_i^{w,l}$ denotes the flow rate between the well and LF well-block i . This is simply the sum of the HF flow rates $f_k^{w,h}$ in the range of coarse well-block i . The LF block pressure p_i^l is again the volume average of the corresponding HF block pressures.

In the upscaling procedure, we solve the single-phase flow problem (Eq. (7)) with well BHPs corresponding to those specified in the two-phase flow problem. By computing WI_i^* for each well block, and T_β^* for each face connecting the well block to a neighboring cell, we can achieve very accurate single-phase fluxes. This procedure also corrects for effects that occur if the well is off-center in the coarse block. The use of WI_i^* and T_β^* (computed from a single-phase flow solution) in LF simulations of the two-phase system defined by Eqs. (1) and (2) leads to some error, relative to the HF solution, in the fluxes of the oil and water phases. The accuracy of the LF model could be improved by upscaling the relative permeability functions. Relative permeability upscaling is not used here because it is much more expensive than the single-phase parameter upscaling described above.

The upscaling procedure is applied to generate LF geomodels $\mathbf{m}^l \in \mathbb{R}^{n^l \times 4}$ from high-fidelity models $\mathbf{m}^h \in \mathbb{R}^{n^h \times 1}$, where $n^l = n_x^l \times n_y^l \times n_z^l$ is the number of LF grid blocks. Here $n^l \times 4$ appears in the LF model dimension because this geomodel is characterized, in general, by LF porosity and different transmissibilities in the x , y and z -directions (even when the HF permeability field is isotropic). The two-phase flow simulation of the LF model can be expressed as

$$\mathbf{x}^l = [\mathbf{P}^l, \mathbf{S}^l] = g(\mathbf{m}^l), \quad (11)$$

where $\mathbf{P}^l \in \mathbb{R}^{n^l \times n_{ts}}$, $\mathbf{S}^l \in \mathbb{R}^{n^l \times n_{ts}}$ denote the pressure and saturation from LF simulation and \mathbf{x}^l represents LF simulation output.

3. Network training with transfer learning and multifidelity data

In this section we first provide a brief description of the recurrent R-U-Net surrogate model [1,2]. The use of transfer learning with multifidelity data to train the surrogate model is then described.

3.1. 3D recurrent residual U-Net

With the existing deep-neural-network surrogate [2], the forward model can be expressed as

$$\hat{\mathbf{x}}^h = [\hat{\mathbf{P}}^h, \hat{\mathbf{S}}^h] = \tilde{g}(\mathbf{m}^h, \theta^h), \quad (12)$$

where $\hat{\mathbf{x}}^h$ represents the HF surrogate model output, $\hat{\mathbf{P}}^h \in \mathbb{R}^{n^h \times n_t}$ and $\hat{\mathbf{S}}^h \in \mathbb{R}^{n^h \times n_t}$ are the surrogate model pressure and saturation output, \tilde{g} denotes the surrogate model, and θ indicates the tunable network parameters. Here n_t is the number of time steps at which surrogate model output is constructed. This is generally somewhat less than the number of simulation time steps n_{ts} , e.g., in this work $n_{ts} \sim 30$ – 35 , and $n_t = 10$.

The 3D recurrent R-U-Net architecture is used as the framework in this study. The residual U-Net, illustrated in Fig. 1, includes encoding and decoding networks to capture the spatial correlation in simulation input and output. The encoding network maps the input geomodel \mathbf{m}^h to low-dimensional latent features $\mathbf{F}_1, \dots, \mathbf{F}_5$. These feature maps are concatenated with the upsampled features in the decoding net, which facilitates the representation of multiscale effects in the solution.

The recurrent network is introduced to capture the temporal evolution of the simulation output. Fig. 2 displays the recurrent component of the recurrent R-U-Net. The overall surrogate model includes the residual U-Net and a long short-term memory convolutional (convLSTM) recurrent network. The most compressed (and most global) feature \mathbf{F}_5 is mapped to a time series of latent features, $\mathbf{F}_5^1, \mathbf{F}_5^2, \dots, \mathbf{F}_5^{n_t}$, by the convLSTM module. The feature \mathbf{F}_5^t from the convLSTM (for $t = 1, \dots, n_t$), along with extracted features $\mathbf{F}_1, \dots, \mathbf{F}_4$ from the encoding net, are combined as they move through the decoding net. The decoding net then generates a prediction $\hat{\mathbf{x}}^{h,t}$ for each of the n_t time steps. Please see [1,2] for more details on the recurrent R-U-Net architecture.

The recurrent R-U-Net is trained by minimizing the loss between the simulated pressure and saturation fields and the corresponding surrogate model predictions. A set of forward simulations is performed with specified well locations and settings (BHPs in our case). The pressure and saturation data from n_t time steps are collected. The minimization problem is given by

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \frac{1}{n_{smp}} \frac{1}{n_t} \sum_{i=1}^{n_{smp}} \sum_{t=1}^{n_t} \|\hat{\mathbf{x}}_i^{h,t} - \mathbf{x}_i^{h,t}\|_2^2 + \lambda_w \frac{1}{n_{smp}} \frac{1}{n_t} \frac{1}{n_w} \sum_{i=1}^{n_{smp}} \sum_{t=1}^{n_t} \sum_{w=1}^{n_w} \|\hat{\mathbf{x}}_i^{h,t,w} - \mathbf{x}_i^{h,t,w}\|_2^2, \quad (13)$$

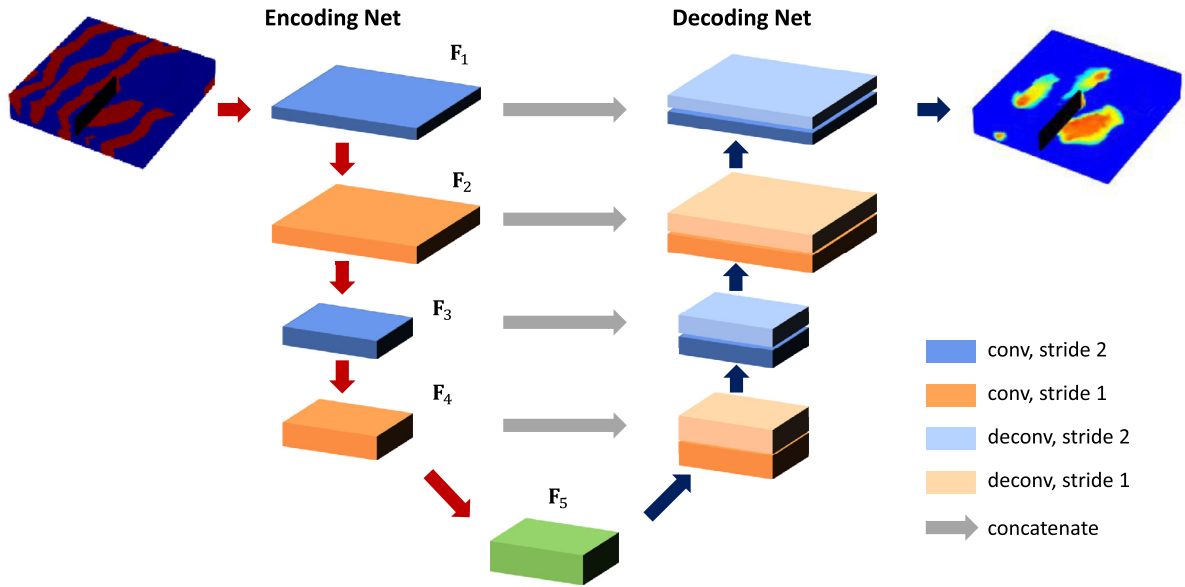


Fig. 1. 3D residual U-Net architecture. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

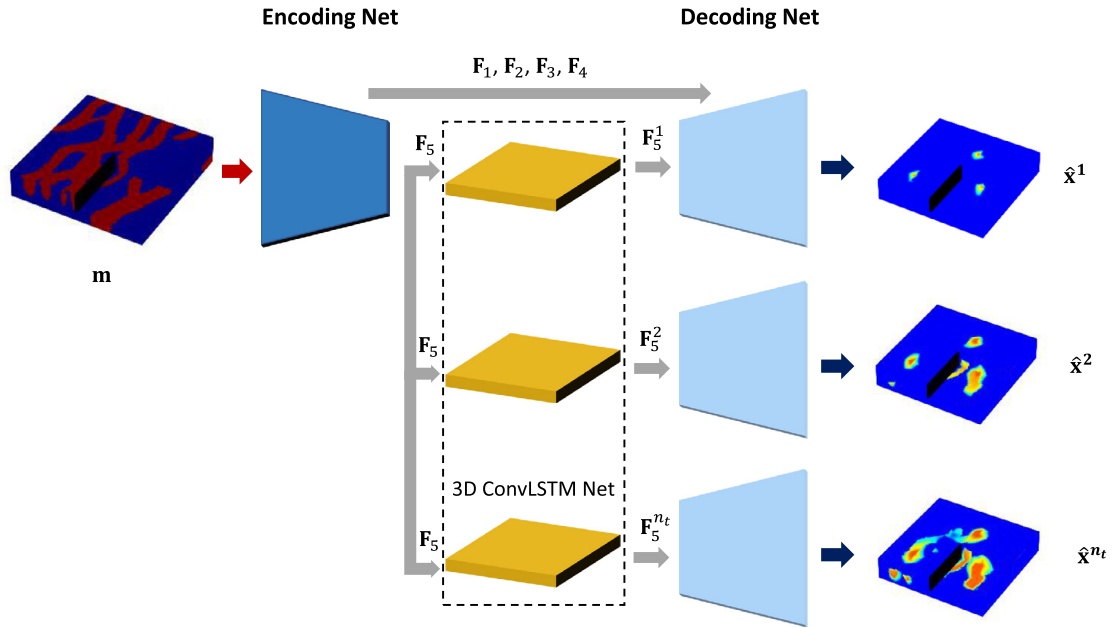


Fig. 2. Recurrent R-U-Net architecture [2].

where n_{smp} is the number of training samples, n_w is the number of well blocks, and λ_w represents the weighting for well-block data. The second term in Eq. (13) enables additional weighting at well blocks, which acts to enhance the accuracy of pressure and saturation at well locations. This in turn improves well rate predictions (see Eq. (4)), which are often the key observations used in data assimilation problems. Note that, following [2], separate networks are trained for pressure and saturation predictions.

3.2. Transfer learning with multifidelity data

The training of the existing recurrent R-U-Net requires a large number of high-fidelity simulation runs as training samples. For example, in Tang et al. [3], 2500 runs were used to train the network to predict pressure and saturation in 3D channelized models containing 128,000 grid blocks. Each high-fidelity simulation for the 3D models considered in this work (which also contain 128,000 blocks) requires 20 minutes using ADGPRS on a single CPU. Practical 3D models, which may

contain millions of grid blocks, may entail hours of computation for a single forward simulation. Thus there is a practical need to reduce the computational demands associated with network training.

The idea of transfer learning is to apply a neural network trained for one task to a different but related task. With the knowledge embodied in the neural network trained for the first task, the training cost for the second task may be substantially reduced. In the approach used here, the model is first trained with 2500 LF (coarse-scale) simulation runs. The coarse-scale flow physics (i.e., simulation operator g) is the same as that on the fine scale, and the coarse-scale flow responses approximate (with some error) the HF simulation results. A relatively small number of HF simulations (100–400 are considered here) are then applied to fine-tune the neural networks and thus complete the HF surrogate model.

The transfer learning process involves three steps, which are shown in Fig. 3. First, we predict LF model output $\hat{\mathbf{x}}^l$ from the input high-fidelity geomodel $\mathbf{m}^h \in \mathbb{R}^{n^h \times 1}$,

$$\hat{\mathbf{x}}^l = [\hat{\mathbf{P}}^l, \hat{\mathbf{S}}^l] = \tilde{g}(\mathbf{m}^h, \theta^l). \quad (14)$$

Here θ^l denotes the parameters for the LF neural networks. We train the surrogate model with $n_{\text{smpl}}^l = 2500$ LF samples. The model parameters $\theta^l = [\theta_{\text{enc}}^l, \theta_{\text{convLSTM}}^l, \theta_{\text{dec}}^l, \theta_{\text{output}}^l]$ include parameters for the encoder (θ_{enc}^l), 3D convLSTM network ($\theta_{\text{convLSTM}}^l$), decoder (θ_{dec}^l), and output layers (θ_{output}^l). Note that the LF geomodel \mathbf{m}^l is not input to the network (here or in subsequent steps). Rather, the network always works with \mathbf{m}^h , even when it is trained using \mathbf{x}^l .

In the Step 1 training process, the surrogate model is trained to capture spatial distributions and temporal evolution with parameters $\theta_{\text{enc}}^l, \theta_{\text{convLSTM}}^l, \theta_{\text{dec}}^l$. These parameters are associated with major portions of the recurrent R-U-Net. Step 1 training also provides θ_{output}^l . In Step 2, parameters $\theta_{\text{enc}}^l, \theta_{\text{convLSTM}}^l$ and θ_{dec}^l are not updated, though the output-layer parameters are modified. Specifically, in this step we map the geomodel input \mathbf{m}^h to HF simulation output $\hat{\mathbf{x}}^h \in \mathbb{R}^{n^h \times n_t}$ through

$$\hat{\mathbf{x}}^h = [\hat{\mathbf{P}}^h, \hat{\mathbf{S}}^h] = \tilde{g}(\mathbf{m}^h, \theta^h). \quad (15)$$

The parameter set is now defined as $\theta^h = [\theta_{\text{enc}}^h, \theta_{\text{convLSTM}}^h, \theta_{\text{dec}}^h, \theta_{\text{output}}^h]$. The output layers with parameters θ_{output}^h are trained with n_{smpl}^h HF samples. In this work, we evaluate network performance over the range $n_{\text{smpl}}^h = 100\text{--}400$.

The last step of the transfer learning procedure is the fine-tuning of all parameters using the n_{smpl}^h HF samples. The parameter set is now denoted by $\theta^h = [\theta_{\text{enc}}^h, \theta_{\text{convLSTM}}^h, \theta_{\text{dec}}^h, \theta_{\text{output}}^h]$. The parameters $\theta_{\text{enc}}^h, \theta_{\text{convLSTM}}^h$ and θ_{dec}^h are well-approximated by $\theta_{\text{enc}}^l, \theta_{\text{convLSTM}}^l$ and θ_{dec}^l , and an initial estimate for θ_{output}^h is already available. For this reason this training step requires relatively few epochs and is thus fast to converge.

Table 1 presents the detailed architecture of the 3D recurrent R-U-Net with HF and LF data. The input model is of dimensions $n_x^h \times n_y^h \times n_z^h \times 1$. The HF and LF models share the same encoder, 3D convLSTM and decoder networks (though these are updated in Step 3). Different output layers are applied to map the decoder output to LF data (of dimensions $n_x^l \times n_y^l \times n_z^l \times 1 \times n_t$) and to HF data (dimension of $n_x^h \times n_y^h \times n_z^h \times 1 \times n_t$).

The encoder is composed of four ‘conv’ blocks and two residual blocks. Each conv block includes one 3D convolutional layer, a batch normalization layer and ReLU nonlinear activation. The residual block includes one conv block with 64 filters of size $3 \times 3 \times 3$ with stride (1, 1, 1), followed by one 3D convolutional layer and a batch normalization layer. The output of the residual block is the sum of the input of the residual block and the output of the second convolutional layer. The output of the encoder is fed to the 3D convLSTM net, which uses 64 filters of size $3 \times 3 \times 3$ with stride (1, 1, 1) in all LSTM blocks.

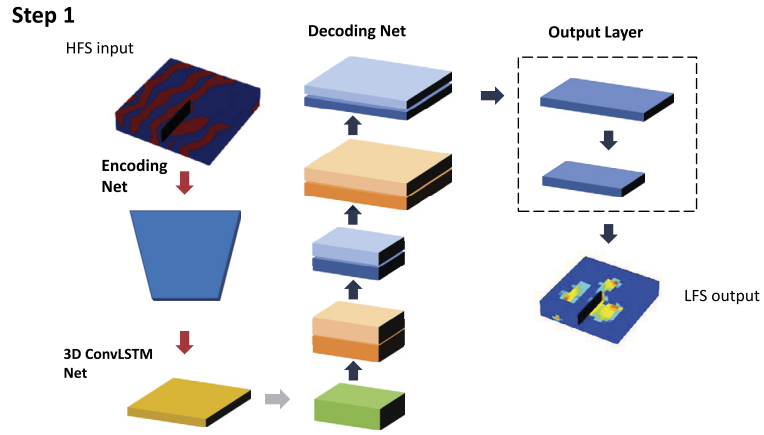
The output size of the 3D convLSTM net is $(\frac{n_x^h}{4}, \frac{n_y^h}{4}, \frac{n_z^h}{4}, 64)$ for n_t time steps. The decoder includes two residual blocks and four ‘deconv’ blocks. The deconv block is a stack of a 3D deconvolutional (upsampling) layer, a batch normalization layer and ReLU nonlinear activation. The decoder output is of dimensions $(n_x^h, n_y^h, n_z^h, 16, n_t)$.

In the system considered in this work, the fine-scale (HF) geomodel is defined on a grid of dimensions $80 \times 80 \times 20$, and it is upscaled to a $20 \times 20 \times 10$ LF grid. For the LF surrogate model in Step 1, the output layers include two 3D convolutional layers to map the decoder output to LF output of dimensions $(n_x^l, n_y^l, n_z^l, 1, n_t)$. The HF output layer in Steps 2 and 3 is a 3D convolutional layer that generates output of dimensions $(n_x^h, n_y^h, n_z^h, 1, n_t)$.

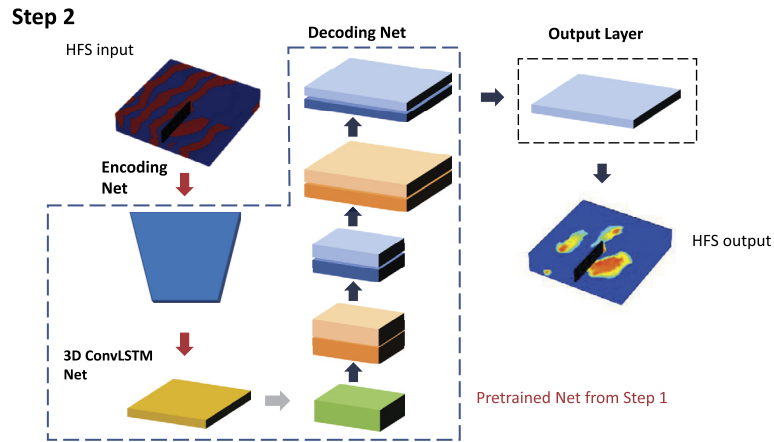
Data preprocessing is essential, and the detailed treatments impact the performance of the overall procedure. In this study the network is applied to predict pressure and saturation for binary geomodels with wells operating under BHP control. The geomodels \mathbf{m}^h are defined in terms of ones (indicating the cell contains sand) and zeros (indicating shale/mud), so no preprocessing of \mathbf{m}^h is required. Saturation values are in the range of 0 to 1, so these can also be used directly. For pressure normalization, the BHPs of the injectors and producers are taken as the maximum and minimum values, respectively. Min-max normalization is applied for both the LF and HF pressure output.

The minimization applied in Step 1 training, involving LF data and predictions, is expressed as

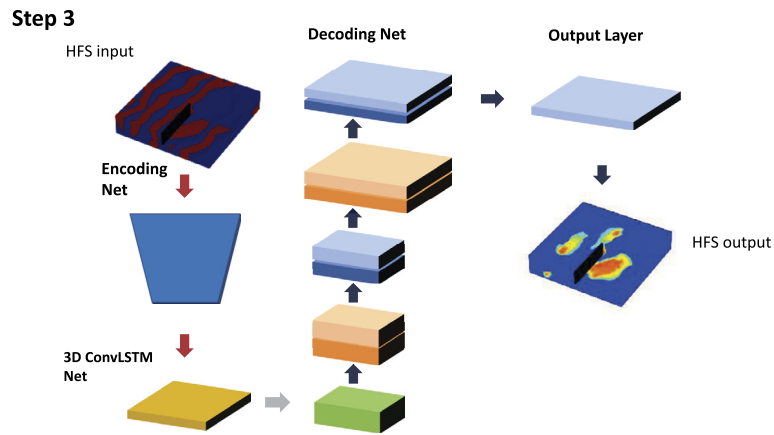
$$(\theta^l)^* = \underset{\theta^l}{\operatorname{argmin}} \frac{1}{n_{\text{smpl}}^l} \frac{1}{n_t} \sum_{i=1}^{n_{\text{smpl}}^l} \sum_{t=1}^{n_t} \|\hat{\mathbf{x}}_i^{l,t} - \mathbf{x}_i^{l,t}\|_2^2 + \lambda_w^l \frac{1}{n_{\text{smpl}}^l} \frac{1}{n_t} \frac{1}{n_w^l} \sum_{i=1}^{n_{\text{smpl}}^l} \sum_{t=1}^{n_t} \sum_{w=1}^{n_w^l} \|\hat{\mathbf{x}}_i^{l,t,w} - \mathbf{x}_i^{l,t,w}\|_2^2, \quad (16)$$



(a) Step 1: surrogate model training with LF output layer



(b) Step 2: transfer learning with HF output layer



(c) Step 3: fine-tuning of full network with HF data

Fig. 3. Workflow for surrogate model training with multifidelity data and transfer learning.

Table 1
Architecture of 3D recurrent R-U-Net with multifidelity data.

Network	Layer	Output
Encoder (Θ_{enc})	Input	$(n_x^h, n_y^h, n_z^h, 1)$
	conv, 16 filters of size $3 \times 3 \times 3$, stride 2	$(\frac{n_x^h}{2}, \frac{n_y^h}{2}, \frac{n_z^h}{2}, 16)$
	conv, 32 filters of size $3 \times 3 \times 3$, stride 1	$(\frac{n_x^h}{2}, \frac{n_y^h}{2}, \frac{n_z^h}{2}, 32)$
	conv, 32 filters of size $3 \times 3 \times 3$, stride 2	$(\frac{n_x^h}{4}, \frac{n_y^h}{4}, \frac{n_z^h}{4}, 32)$
	conv, 64 filters of size $3 \times 3 \times 3$, stride 1	$(\frac{n_x^h}{4}, \frac{n_y^h}{4}, \frac{n_z^h}{4}, 64)$
	residual block, 64 filters of size $3 \times 3 \times 3$, stride 1	$(\frac{n_x^h}{4}, \frac{n_y^h}{4}, \frac{n_z^h}{4}, 64)$
	residual block, 64 filters of size $3 \times 3 \times 3$, stride 1	$(\frac{n_x^h}{4}, \frac{n_y^h}{4}, \frac{n_z^h}{4}, 64)$
		$(\frac{n_x^h}{4}, \frac{n_y^h}{4}, \frac{n_z^h}{4}, 64)$
3D ConvLSTM ($\Theta_{convLSTM}$)	convLSTM3D, 64 filters $3 \times 3 \times 3$, stride 1	$(\frac{n_x^h}{4}, \frac{n_y^h}{4}, \frac{n_z^h}{4}, 64, n_t)$
Decoder (Θ_{dec})	residual block, 64 filters of size $3 \times 3 \times 3$, stride 1	$(\frac{n_x^h}{4}, \frac{n_y^h}{4}, \frac{n_z^h}{4}, 64, n_t)$
	residual block, 64 filters of size $3 \times 3 \times 3$, stride 1	$(\frac{n_x^h}{4}, \frac{n_y^h}{4}, \frac{n_z^h}{4}, 64, n_t)$
	deconv, 64 filters of size $3 \times 3 \times 3$, stride 1	$(\frac{n_x^h}{4}, \frac{n_y^h}{4}, \frac{n_z^h}{4}, 64, n_t)$
	deconv, 32 filters of size $3 \times 3 \times 3$, stride 2	$(\frac{n_x^h}{4}, \frac{n_y^h}{4}, \frac{n_z^h}{4}, 32, n_t)$
	deconv, 32 filters of size $3 \times 3 \times 3$, stride 1	$(\frac{n_x^h}{2}, \frac{n_y^h}{2}, \frac{n_z^h}{2}, 32, n_t)$
	deconv, 16 filters of size $3 \times 3 \times 3$, stride 2	$(n_x^h, n_y^h, n_z^h, 16, n_t)$
Output Layers (LF, Θ_{output}^l)	multi conv, 1 filters of size $3 \times 3 \times 3$, stride 2	$(n_x^l, n_y^l, n_z^l, 1, n_t)$
Output Layers (HF, Θ_{output}^h)	conv, 1 filters of size $3 \times 3 \times 3$, stride 1	$(n_x^h, n_y^h, n_z^h, 1, n_t)$

where λ_w^l represents the additional weight for the n_w^l well blocks in the LF model. For Step 2 and Step 3 training (HF data and predictions) the minimization problem is

$$(\theta^h)^* = \operatorname{argmin}_{\theta^h} \frac{1}{n_{smp}^h} \frac{1}{n_t} \sum_{i=1}^{n_{smp}^h} \sum_{t=1}^{n_t} \|\hat{\mathbf{x}}_i^{h,t} - \mathbf{x}_i^{h,t}\|_2^2 + \lambda_w^h \frac{1}{n_{smp}^h} \frac{1}{n_t} \frac{1}{n_w^h} \sum_{i=1}^{n_{smp}^h} \sum_{t=1}^{n_t} \sum_{w=1}^{n_w^h} \|\hat{\mathbf{x}}_i^{h,t,w} - \mathbf{x}_i^{h,t,w}\|_2^2, \quad (17)$$

where λ_w^h is the well-block weighting for the n_w^h well blocks in the HF model. Pressure quantities in Eqs. (16) and (17) are normalized as described earlier. Note that in Step 2, only the output layer parameters θ_{output}^h are determined (meaning θ_{output}^h replaces θ^h in Eq. (17)). In Step 3, all network parameters θ^h are updated. Following training, the surrogate model provides flow predictions for a new geomodel in about 0.05 seconds on a single GPU.

4. Surrogate model evaluation

In this section, we describe the problem setup and then present detailed results using the transfer-learning-based surrogate model. Comparisons with flow results from HF simulation, LF simulation, and a reference deep-learning surrogate procedure are provided. The impacts of algorithmic treatments and parameters are illustrated.

4.1. Problem setup

The 3D channelized system considered in this work corresponds to that used in [3], though the specific realizations here are all newly generated. The well types and completion intervals are also different from those in [3]. The models are binary and include channel sand and background mud/shale. The realizations are randomly generated through application of a convolutional neural network – principal component analysis (CNN-PCA) parameterization procedure [27]. The CNN-PCA representation itself is trained using reference realizations generated through application of geostatistical software (see [27] for details on the parameterization procedure).

The 3D CNN-PCA model in [3] is applied to generate the geological realizations used for training and testing. The fine-scale (HF) geomodels are defined on an $80 \times 80 \times 20$ grid, with blocks of size $20 \text{ m} \times 20 \text{ m} \times 2 \text{ m}$. Fig. 4 presents three geomodel realizations. Well locations are shown in Fig. 4(a). Porosity is set to 0.25 in sand facies and 0.1 in mud/shale facies. Permeability is prescribed as 2000 md for sand and 20 md for mud/shale. The realization in Fig. 4(c) is used for some of the results shown later in this section, and it also represents the ‘true’ model applied for history matching in Section 5.

The geological realizations are conditioned to facies type at well locations. In contrast to the setup in [3], here there are three injection wells (denoted I1 to I3) and five production wells (denoted P1 to P5), as shown in Fig. 4(a). The initial reservoir pressure is 325 bar. The injectors and producers operate at fixed bottom-hole pressures (BHPs) of 330 bar and 310 bar, respectively. Wells I2, I3, P3, P4 and P5 are completed (open to flow) in the top eight layers of the model (layers 1–8), while wells I1, P1 and P2 are completed in layers 13–20.

The flow problem involves two immiscible fluids (oil and water). Fig. 5 shows the oil-water relative permeability curves used for both the HF and LF flow simulations. The initial water and oil saturations are 0.1 and 0.9. Water viscosity is constant

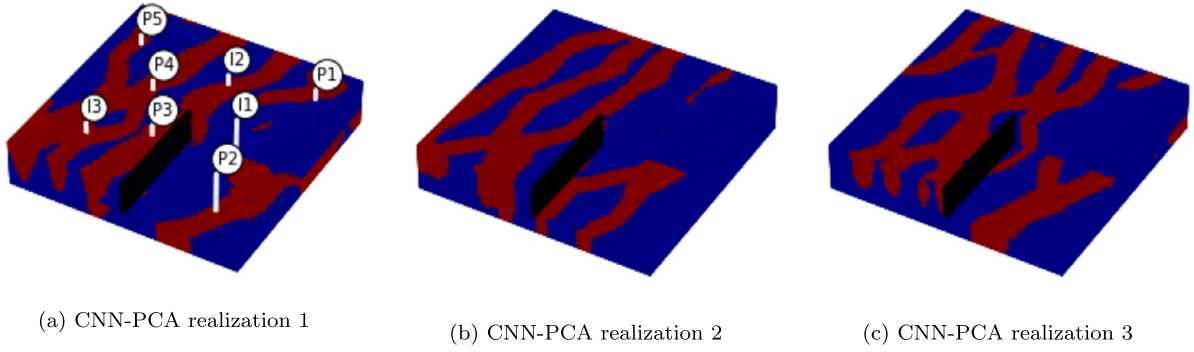


Fig. 4. CNN-PCA geomodel realizations conditioned to hard data at eight well locations. Well locations are shown in (a). Realization in (c) corresponds to 'true' model used for history matching. Cutaway view used for better 3D visualization.

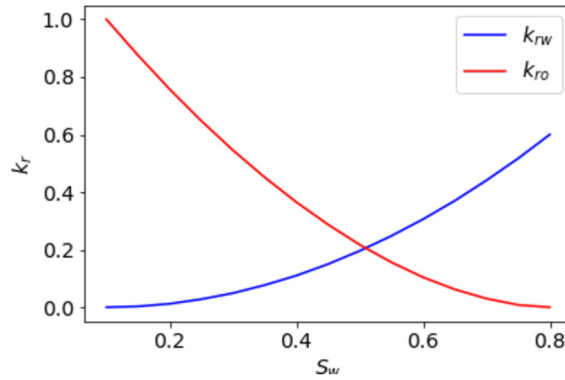


Fig. 5. Oil-water relative permeability curves.

at 0.31 cp; oil viscosity varies with pressure and is 1.14 cp at 325 bar. All flow simulations are performed using ADGPRS [21]. The simulation time frame is 1000 days. The global transmissibility upscaling procedure described in Section 2.2 is applied to generate the LF models. These models are defined on grids of dimensions $20 \times 20 \times 10$, which corresponds to an upscaling factor of 32.

4.2. Surrogate model training and overall performance

The process for generating the multifidelity data used for training is as follows. We generate 2500 HF binary geomodel realizations with CNN-PCA. These realizations are upscaled to LF models using the procedure described in Section 2.2. A total of $n_{smp}^l = 2500$ LF simulation runs are performed with ADGPRS. These represent the samples used in Step 1 of the training procedure. The pressure and saturation fields for these LF simulations are collected at $n_t = 10$ time steps (specifically at 50, 100, 150, 300, 400, 500, 600, 700, 850 and 1000 days). Separate surrogate models are trained for pressure and saturation with these 2500 LF simulation results.

The hyperparameters affecting performance include the learning rate, the weight on pressure and saturation data at well blocks, the number of epochs, and the batch size. In Step 1 of the training, we set the well-block weight λ_w^l to 20 and the batch size to 4. The impact of λ_w^l on prediction accuracy will be shown below. The networks are trained for a maximum of 150 epochs using the adaptive moment estimation (ADAM) optimizer [28] with an initial learning rate of 0.003. A learning rate decay schedule, with a decay factor of 2 and patience of 10 epochs, is applied. This means the learning rate is reduced by a factor of 2 if the loss does not decrease over 10 epochs. The parameters β_1 and β_2 , which control the exponential decay rates of the moving averages of the gradient and squared gradient in ADAM, are set to 0.9 and 9.999, respectively. Training may also be terminated after 70 epochs, as discussed later.

The number of HF samples used for transfer learning affects the computational cost and performance of the surrogate model. It is useful to conduct numerical experiments to determine an optimal number of HF samples. We consider $n_{smp}^h = 100, 200, 300$ and 400 (representative) realizations selected from the full set of 2500 HF geomodels. These HF models are then simulated, and Steps 2 and 3 of the training framework are applied. It is important to note that even 400 HF training samples are not sufficient, by themselves, to achieve accurate results with the recurrent R-U-Net surrogate model. This will be demonstrated later when we present results for a range of training procedures.

The HF models to be simulated are selected through use of the clustering strategy suggested in [29]. Specifically, a K-means procedure is first applied on the normalized flow response data constructed from the full set of LF simulation results.

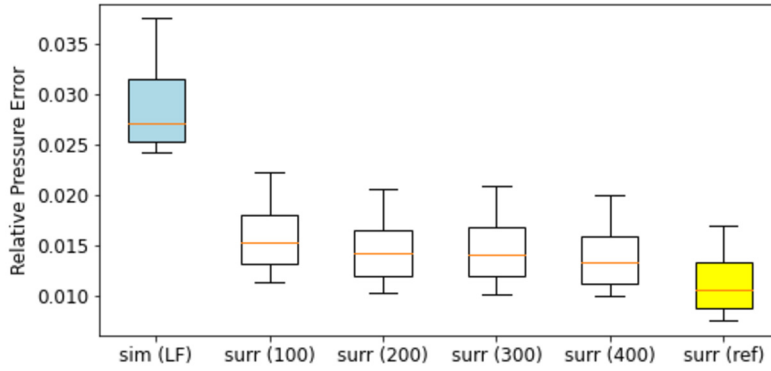


Fig. 6. Relative pressure errors for 400 test samples (compared to HF simulation results) for LF (coarse) simulation, surrogate model trained with 100, 200, 300 and 400 HF samples, and reference surrogate model trained with 2500 HF simulations.

A total of n_{smp}^h clusters are generated, and a k-medoids method is then used to select the ‘centers’ of the n_{smp}^h clusters. These (cluster center) realizations comprise the representative set. The overall clustering procedure requires only a few seconds of computation, so this step is very fast compared to the training simulations and the training of the networks. HF simulation is then performed on all n_{smp}^h realizations to provide the HF training data.

The HF simulation results are used in Steps 2 and 3 of the training. In these steps, consistent with Tang et al. [2], we specify $\lambda_w^h = 1000$. This increased weighting ($\lambda_w^l = 20$ in Step 1) is applied because we now have a much lower fraction of well blocks with pressure and saturation data ($64/128,000 = 0.0005$ for HF in contrast to $32/4000 = 0.008$ for LF), and because HF well data are especially important to capture. The initial learning rates are set to 0.003 (Step 2) and 0.001 (Step 3). We train the output layers for 200 epochs in Step 2 and fine-tune the model for 100 epochs in Step 3. Because a small number of HF samples are used in Steps 2 and 3, the training times for these steps are much smaller than that for Step 1. The detailed training times for the framework will be provided later. We will also present results showing the effectiveness of different training strategies (e.g., only applying training Steps 1 and 2, or Steps 1 and 3) in the transfer learning process.

For comparison purposes, we also train a reference recurrent R-U-Net surrogate model using only high-fidelity samples. The same original set of 2500 HF geomodels is used for this training. The HF networks (one for pressure and one for saturation) are trained for 150 epochs with an initial learning rate of 0.003.

A new set of $n_e = 400$ HF geological realizations is then generated and simulated. These represent the test samples used to evaluate the performance of the transfer-learning-based surrogate model. The performance of the surrogate model is compared to HF simulation results. The relative error for pressure for test sample i , denoted δ_p^i , for $i = 1, \dots, n_e$, is given by

$$\delta_p^i = \frac{1}{n^h n_t} \sum_{j=1}^{n^h} \sum_{t=1}^{n_t} \frac{|\hat{p}_{i,j}^t - p_{i,j}^t|}{p_{max} - p_{min}}. \quad (18)$$

Here $p_{i,j}^t$ is the HF pressure from simulation output in grid block j at time t for test sample i , $\hat{p}_{i,j}^t$ is the corresponding pressure output for the surrogate model, and p_{max} and p_{min} are the injector and producer BHPs (330 bar and 310 bar). The relative error for saturation is given by

$$\delta_s^i = \frac{1}{n^h n_t} \sum_{j=1}^{n^h} \sum_{t=1}^{n_t} \frac{|\hat{S}_{i,j}^t - S_{i,j}^t|}{S_{i,j}^t}, \quad (19)$$

for $i = 1, \dots, n_e$, where $S_{i,j}^t$ and $\hat{S}_{i,j}^t$ represent saturation from HF simulation and the surrogate model, respectively. The initial (and minimum) water saturation is 0.1, so the denominator in Eq. (19) does not approach zero.

Relative errors for pressure and saturation for the 400 test samples, computed using Eqs. (18) and (19), are shown in Figs. 6 and 7. These errors are presented as box plots. In this and all subsequent box plots, the top and bottom of each box show the P_{75} and P_{25} (75th and 25th percentile) relative errors, and the solid red line provides the P_{50} result. The P_{90} and P_{10} relative errors are indicated by the lines extending beyond the boxes. In the figures, the blue boxes display the relative errors associated with LF (coarse-scale) simulation results. The yellow boxes indicate the relative errors for the reference surrogate model (average errors of 1.18% for pressure and 3.70% for saturation) – these represent the best performance that could be expected from the transfer-learning-based surrogate. The white boxes show the relative errors for the transfer-learning-based surrogate model trained with different numbers of HF samples.

The relative errors for the LF simulation results (blue boxes) are computed by projecting the LF pressure or saturation solution to the $80 \times 80 \times 20$ grid, which gives pressure or saturation fields that are constant over $4 \times 4 \times 2$ regions.

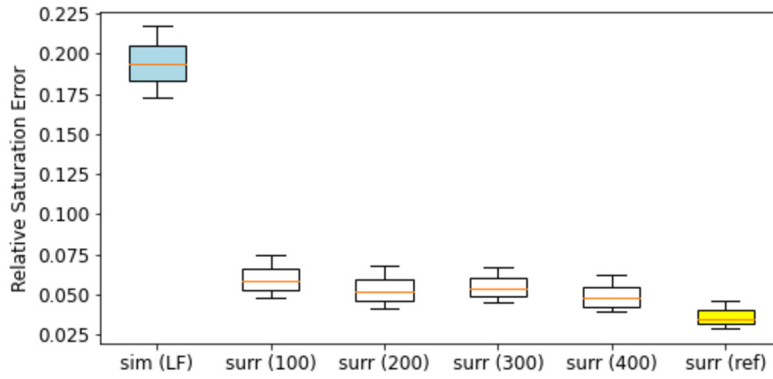


Fig. 7. Relative saturation errors for 400 test samples (compared to HF simulation results) for LF (coarse) simulation, surrogate model trained with 100, 200, 300 and 400 HF samples, and reference surrogate model trained with 2500 HF simulations.

Equations analogous to Eqs. (18) and (19) are then applied to compute δ_p^i and δ_s^i , for $i = 1, \dots, n_e$. Particularly in the case of saturation, the LF error is large. This is mainly due to errors around fluid fronts in the LF simulations.

A key observation from Figs. 6 and 7 is that the errors from the transfer-learning-based surrogate are considerably less than those from LF simulation. This accuracy is achieved even though the significant majority of the training is accomplished using LF models. Though there is a slight trend of decreasing relative error with increasing n_{smp}^h , the use of more than 100 HF samples has only a minor impact on results. For both pressure and saturation, the errors for the reference surrogate (yellow boxes) are less than those with the transfer-learning-based surrogate, as would be expected. The differences are relatively small, however.

In many subsurface flow problems, the flow rates associated with wells represent key quantities of interest. Here we calculate an aggregate relative well rate error δ_r^i as

$$\begin{aligned} \delta_r^i = & \frac{1}{N_{inj}} \sum_{j=1}^{N_{inj}} \frac{\int_0^T |\hat{q}_{j,i}^{w,inj}(t) - q_{j,i}^{w,inj}(t)| dt}{\int_0^T |q_{j,i}^{w,inj}(t)| dt} \\ & + \frac{1}{N_{prod}} \sum_{j=1}^{N_{prod}} \frac{\int_0^T |\hat{q}_{j,i}^{w,prod}(t) - q_{j,i}^{w,prod}(t)| dt}{\int_0^T |q_{j,i}^{w,prod}(t)| dt} \\ & + \frac{1}{N_{prod}} \sum_{j=1}^{N_{prod}} \frac{\int_0^T |\hat{q}_{j,i}^{o,prod}(t) - q_{j,i}^{o,prod}(t)| dt}{\int_0^T |q_{j,i}^{o,prod}(t)| dt}, \end{aligned} \quad (20)$$

for $i = 1, \dots, n_e$, where N_{inj} and N_{prod} are the number of injectors and producers and T is the total simulation time. Here $q_{j,i}^{w,inj}(t)$ denotes the injection rate of injector j at time t for test sample i from HF simulation, and $\hat{q}_{j,i}^{w,inj}(t)$ is the corresponding injection rate for the surrogate model. Analogously, the quantities $q_{j,i}^{w,prod}(t)$ and $q_{j,i}^{o,prod}(t)$ are the HF water and oil production rates, and $\hat{q}_{j,i}^{w,prod}(t)$ and $\hat{q}_{j,i}^{o,prod}(t)$ are the corresponding production rates for the surrogate model.

Fig. 8 displays the aggregate relative well rate errors. These errors are larger than the pressure and saturation errors in Figs. 6 and 7, likely because well rates depend nonlinearly on predicted pressure and saturation values in just a few particular grid blocks (see Eq. (4)). There is again a slight decrease in relative error with increasing n_{smp}^h . The P_{90} relative error, for example, decreases from 0.18 with $n_{smp}^h = 100$ to 0.15 for $n_{smp}^h = 200$, and then stays nearly constant with increasing n_{smp}^h . We thus conclude that 200 HF samples is a reasonable choice for this case. In future work, it may be worthwhile to investigate the detailed selection strategy and approaches for determining the optimal value of n_{smp}^h . In the remainder of this paper, we use $n_{smp}^h = 200$ in all transfer-learning-based surrogate results unless otherwise indicated.

Before presenting additional results with $n_{smp}^h = 200$, we consider the impact of different training strategies and algorithmic parameters on surrogate model accuracy. Relative errors for pressure, saturation and well rates computed as described above, with different training procedures, are presented in Fig. 9. The ‘sim (LF),’ ‘surr (200),’ and ‘surr (ref)’ results in Fig. 9 are identical to results shown in Figs. 6, 7 and 8. The ‘HF (200)’ and ‘HF (400)’ results show errors when only 200 or 400 HF models are used for training (i.e., the multifidelity transfer learning approach is not applied). The pressure, saturation and aggregate well rate errors in these cases are clearly larger than those for the multifidelity transfer learning approach (surr (200)). This illustrates the benefit of using a large number of LF samples along with transfer learning.

Next, we compare errors for transfer-learning-based surrogate models trained using different strategies. These results are with 200 HF models. Errors are large – often even greater than those for LF simulation – when only Steps 1 and 2 are

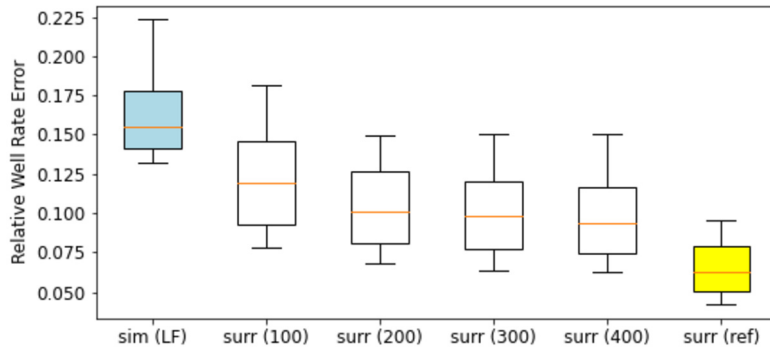


Fig. 8. Aggregate relative well rate errors for 400 test samples (compared to HF simulation results) for LF (coarse) simulation, surrogate model trained with 100, 200, 300 and 400 HF samples, and reference surrogate model trained with 2500 HF simulations.

used. All errors are reduced significantly when Steps 1 and 3 (rather than 1 and 2) are applied. These errors are, however, slightly larger than those observed when the surrogate model is trained using all three steps (surr (200)). Because Step 2 and 3 training is inexpensive (as will be shown below), it is probably worthwhile to perform all three steps of training, even though the benefit from Step 2 is small.

Finally, we consider the impact of the well-block weighting λ_w^l used in Step 1 training (recall we set $\lambda_w^h = 1000$ in Steps 2 and 3 in accordance with [2]). The maximum value assessed ($\lambda_w^l = 62.5$) corresponds to maintaining the same product of the well-block weighting and the fraction of well blocks in the LF model as in the HF case. Results for aggregate relative well rate error are presented in Fig. 10. Accuracy is not strongly affected by the value of λ_w^l , though we do observe that the median and P_{90} errors over the test set are the least for $\lambda_w^l = 20$. Thus, this is the value used in all subsequent results.

4.3. Computational costs of surrogate model construction

The computational cost of the transfer learning workflow includes the generation of LF and HF simulation data and the time for surrogate model training. Flow simulations and the upscaling computations are conducted with ADGPRS. As noted earlier, each HF simulation takes around 20 minutes on a single CPU. The overall cost for simulating 2500 HF samples is thus around 833 hours.

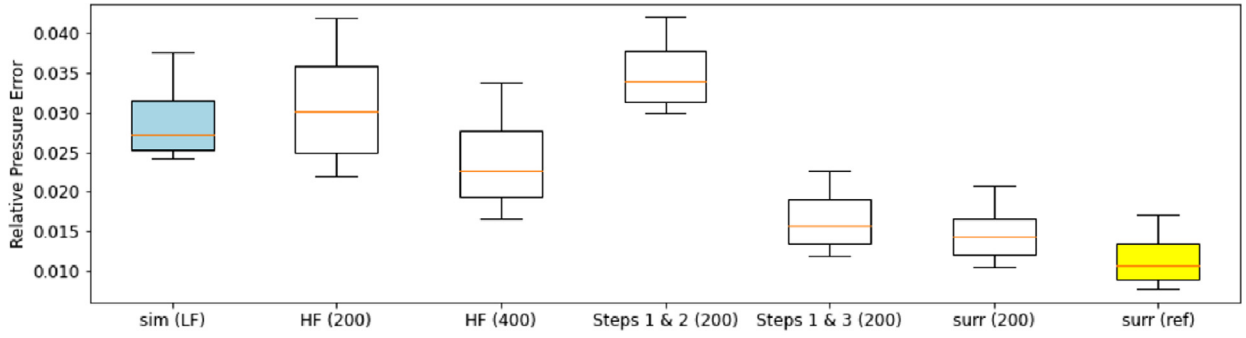
In our evaluation here, consistent with [26], we quantify simulation requirements in units of HF simulation runs. This provides timing assessments that are somewhat independent of the specific simulator and computational environment used. This is because different simulators involve varying amounts of overhead prior to the start of the dynamic computations, which impacts actual wall-clock time. In addition, computing clusters vary in terms of processor availability, i/o speed, job scheduling protocols, etc. Actual timings are also impacted by specific simulator settings (e.g., maximum time step size, convergence criteria). Thus, a more 'generic' way to assess timings is by considering computational complexity, which is how we proceed here. With this approach, for the reference surrogate model the computational cost is $C_{\text{ref}} = 2500$, because all 2500 training runs are performed with HF simulation.

Linear solutions dominate the computational cost of the two-phase flow simulations performed in this work. Linear solvers ideally require $O(N)$ computation, where N is the number of equations/unknowns in the linear system (super-linear scaling occurs in many cases). In HF simulations, $N = 2n^h = 256,000$, while in LF simulations, $N = 2n^l = 8000$. Both HF and LF simulations require ~ 30 – 35 time steps with an average of ~ 6 Newton iterations per time step (this relatively large number of Newton iterations results from taking large time steps). Thus ~ 200 linear solutions are required for each (HF or LF) two-phase flow simulation run. The cost of simulating 2500 LF models is therefore about $C_{\text{LF}} = 2500 \times \frac{1}{32} \approx 78$. Upscaling computations involve HF models, but only one linear solution with one equation/unknown is required. Thus the cost of upscaling 2500 models is about $C_{\text{upscale}} = 2500 \times \frac{1}{200} \times \frac{1}{2} \approx 6$.

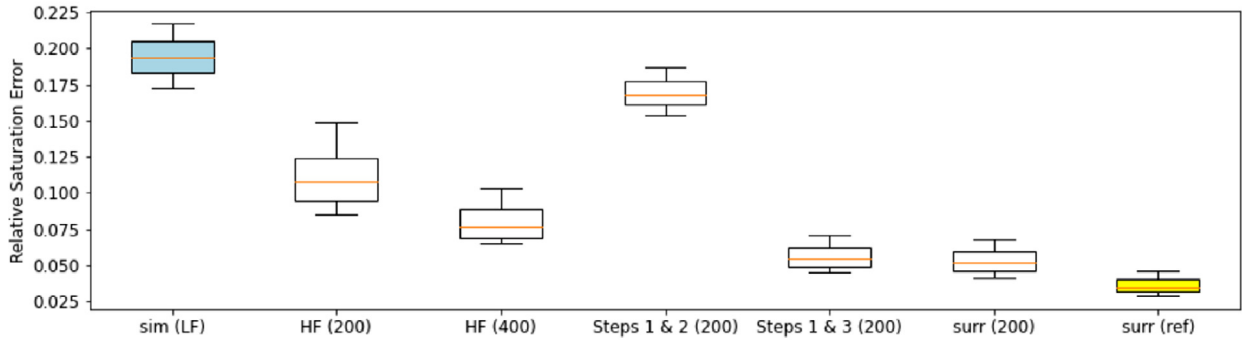
The total simulation cost of generating multifidelity data, with $n_{\text{smp}}^l = 2500$ and $n_{\text{smp}}^h = 200$, can be estimated as $C_{\text{multi}} = 6 + 78 + 200 = 284$. Compared to $C_{\text{ref}} = 2500$, this represents a computational savings of nearly 90%. The estimated computational time for generating the multifidelity data is about 95 hours with a single CPU, compared to 833 hours with HF data. Note that super-linear scaling in the linear solver will lead to larger savings, while significant amounts of simulation overhead/startup time will lead to smaller savings.

The training time depends on the complexity of the surrogate model, the number of training samples, and the number of epochs required for convergence. We train two separate surrogate models for pressure and saturation in parallel on two Nvidia Tesla V100 GPUs. The training for the transfer learning framework includes training with LF samples, transfer learning training, and fine-tuning with HF samples. To achieve convergence, the Step 1 training requires 150 epochs. This can be reduced, however, by stopping this training early.

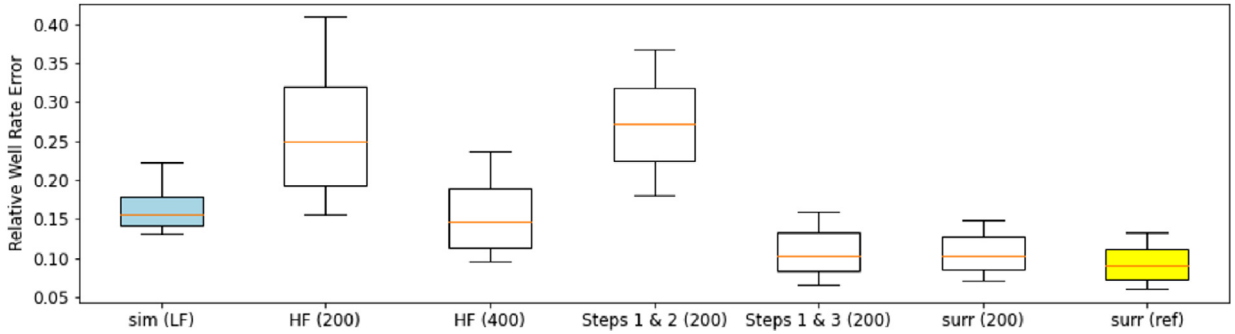
Table 2 presents cost and accuracy results for HF training and two multifidelity training strategies. The training process for the reference HF surrogate model converges in 15 hours (150 epochs, 360 seconds per epoch). For the surrogate model



(a) Relative pressure errors



(b) Relative saturation errors



(c) Aggregate relative well rate errors

Fig. 9. Relative pressure, saturation, and aggregate well rate errors for 400 test samples (compared to HF simulation results) for LF (coarse) simulation, surrogate model trained using 200 and 400 HF samples only, multifidelity surrogate constructed with Step 1 and 2 training (with 200 HF samples), with Step 1 and 3 training (with 200 HF samples), and with all three steps (with 200 HF samples). Results for reference surrogate model trained with 2500 HF simulations also shown.

Table 2

Computational costs of training reference and transfer-learning-based surrogate with multifidelity (MF) data (Step 1: training with LF data, Step 2: transfer learning with HF data, Step 3: fine-tuning with HF data). The relative errors are averages over 400 test samples.

	HF training	MF training	Accl MF training
Step 1	N/A	150 ep \times 360 s/ep = 15 h	70 ep \times 360 s/ep = 7 h
Step 2	N/A	200 ep \times 15 s/ep = 0.83 h	200 ep \times 15 s/ep = 0.83 h
Step 3	150 ep \times 360 s/ep = 15 h	100 ep \times 30 s/ep = 0.83 h	100 ep \times 30 s/ep = 0.83 h
Time	15 h	16.66 h	8.66 h
p error (δ_p)	1.18%	1.52%	1.58%
S error (δ_S)	3.70%	5.38%	5.57%

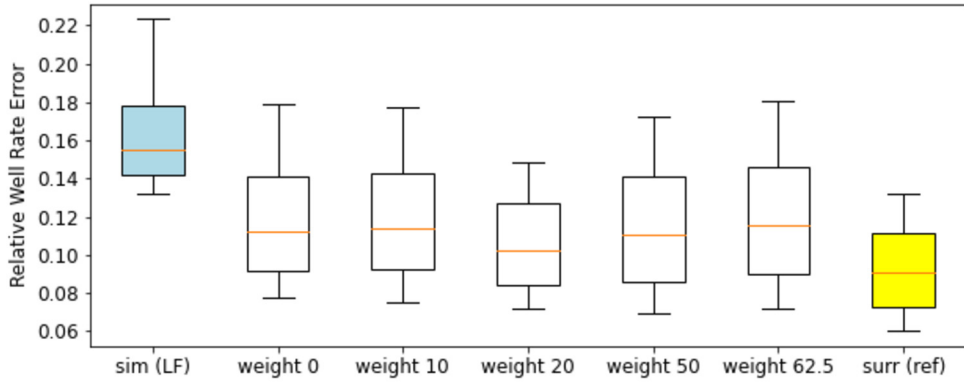


Fig. 10. Aggregate relative well rate errors for 400 test samples (compared to HF simulation results) for LF (coarse) simulation, surrogate models trained with 200 HF samples and different Step 1 well-block weights (λ_w^l), and reference surrogate model trained with 2500 HF simulations.

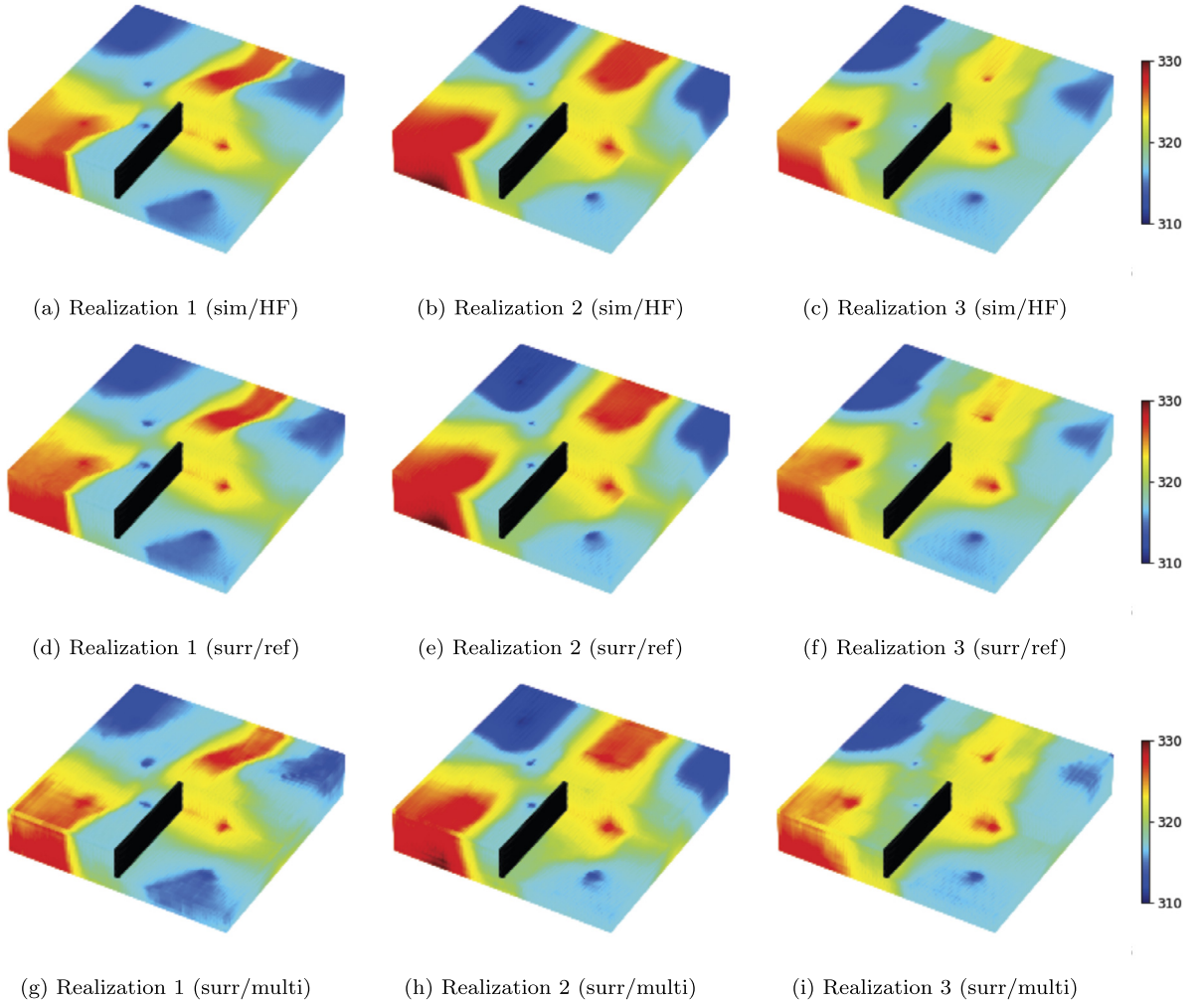


Fig. 11. Pressure fields from HF simulation (top row), surrogate model trained with HF data (middle row), and transfer-learning-based surrogate model trained with multifidelity data (bottom row). All results are at 400 days. Cutaway views highlight 3D effects.

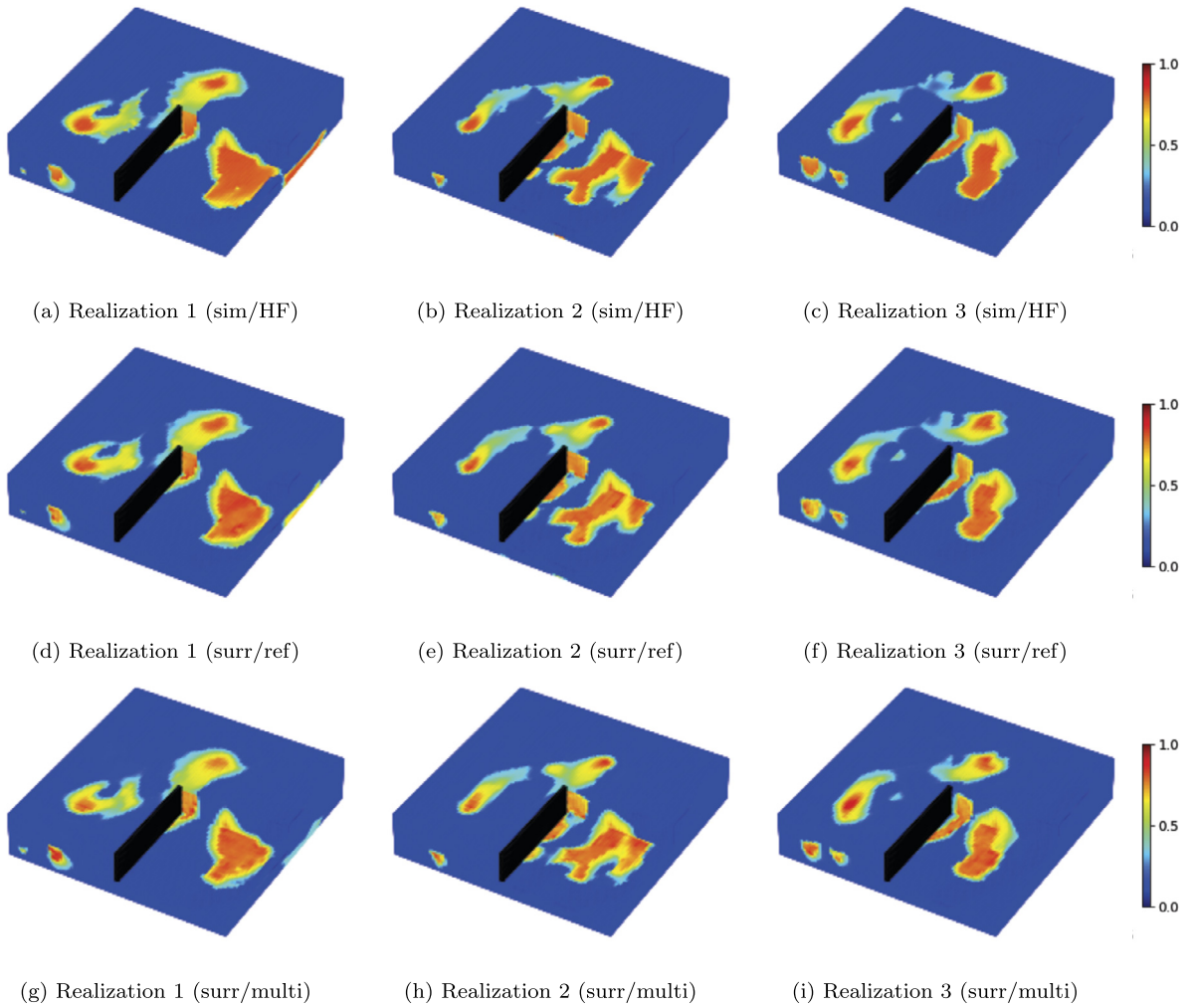


Fig. 12. Water saturation fields from HF simulation (top row), surrogate model trained with HF data (middle row), and transfer-learning-based surrogate model trained with multifidelity data (bottom row). All results are at 1000 days.

trained with multifidelity data with Step 1 iterated until convergence (MF training), the Step 1 training cost is about the same as the full HF training cost. This is because these trainings involve networks of similar complexity and the same number of training samples and epochs. As indicated in the table, the total training time for Steps 2 and 3 is 1.66 hours, which gives an overall training time of 16.66 hours. This is slightly larger than the reference HF training time.

It is not necessary, however, to continue Step 1 training until convergence (150 epochs) since additional training with HF data will be performed. Accelerated training (right-most column of Table 2) entails stopping Step 1 training at 70 epochs, at which point training loss is acceptable, although the training has not fully converged. Overall training time is now reduced to 8.66 hours. Relative errors with this approach are incrementally greater, but the computational savings are substantial. In subsequent results, in all cases the surrogate model is trained using this accelerated process.

In summary, we see that the use of multifidelity data and transfer learning can lead to a nearly 90% reduction in the computations associated with the training simulations, and a savings of about 42% in training time. It is possible these speedups could be further improved with additional tuning, though this was not attempted here.

4.4. Pressure and saturation predictions

To assess the performance of the transfer-learning-based surrogate model trained with multifidelity data, we now compare pressure and saturation predictions to results from HF simulation and the reference surrogate model (trained with 2500 HF simulation runs). The three realizations in Fig. 4, which correspond to test-case realizations with relative pressure and saturation errors slightly larger than the median, are considered. Fig. 11 shows the pressure fields for these three test cases at 400 days. The upper row displays the HF simulation results, the middle row shows results from the reference surrogate model, and the bottom row shows results from the transfer-learning-based surrogate model. There are noticeable differ-

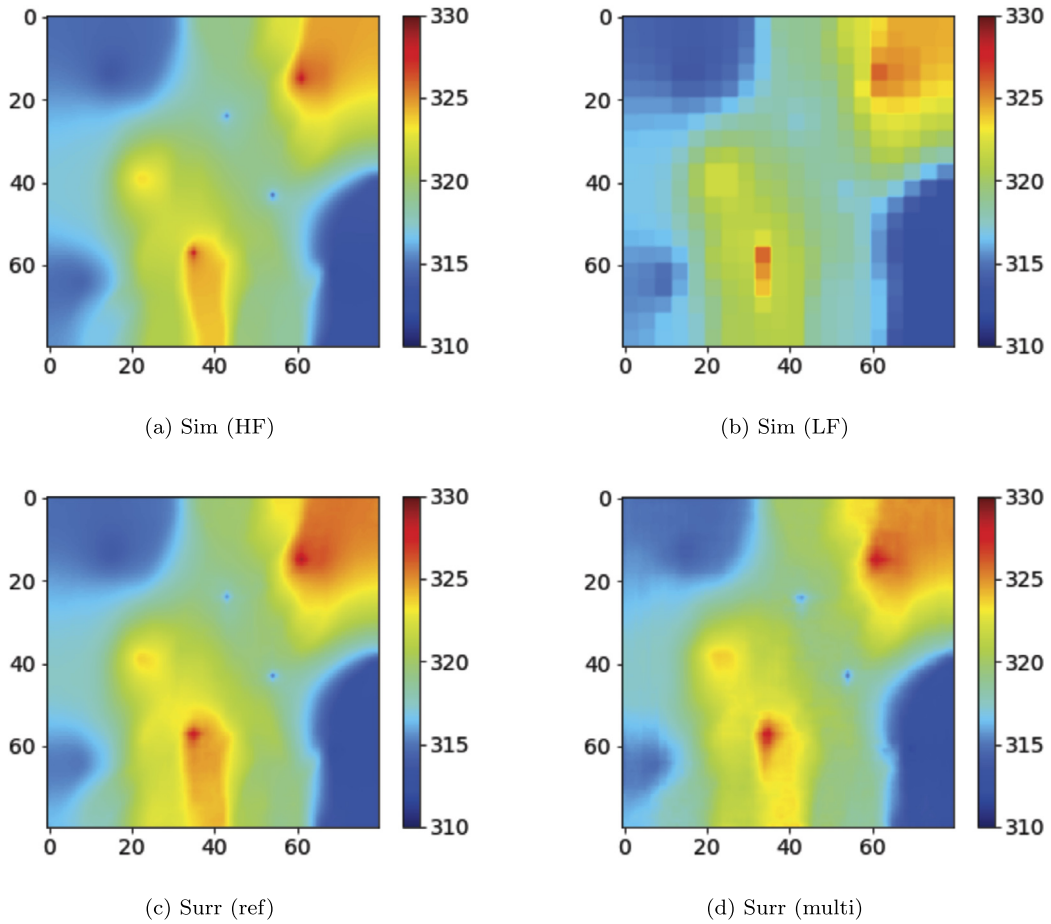


Fig. 13. Pressure maps from HF and LF simulation, reference surrogate, and transfer-learning-based surrogate trained with multifidelity data. Results are for layer 8 from realization in Fig. 4(c) at 1000 days.

ences in the pressure distributions between the three realizations, indicating a reasonable degree of variability over the test set. The reference surrogate model gives results that are visually close to the simulation results. The transfer-learning-based surrogate model also provides results that are in close visual agreement. Some discrepancies are evident, e.g., in the blue feature in the back right corner of the models in Fig. 11(i) and (c), though these are relatively minor.

Water saturation results for the same three realizations, at 1000 days (the end of the simulation), are presented in Fig. 12. We again observe variability between realizations and high accuracy in both the reference surrogate model and the transfer-learning-based surrogate model. The latter model is seen to provide sharp water saturation fronts, even though most of its training involves LF simulation results where fronts are diffused.

We next show pressure and saturation maps at 1000 days for one particular layer (layer 8) in the realization in Fig. 4(c). Fig. 13(a) and (b) display the pressure fields from HF and LF simulation. In Fig. 13(b), the 20×20 LF result is mapped onto an 80×80 grid. Fig. 13(c) and (d) show the pressure predictions from the two surrogate models. The reference surrogate results are in close agreement with the HF simulation result. Some differences are evident in the transfer-learning-based surrogate, though it is evident that some features that are not resolved in the LF simulation result in Fig. 13(b) are captured in the result in Fig. 13(d). Such details include the large yellow feature (midway in x , middle to bottom in y) and the light blue feature along the left edge (around the middle in y).

Analogous results for saturation appear in Fig. 14. The LF simulation result for this quantity, in Fig. 14(b), shows a high degree of smearing. The transfer-learning-based surrogate result in Fig. 14(d) is in close visual agreement with the HF simulation result, and it shows much better front resolution than the LF simulation result. This illustrates the ability of the multifidelity procedure to provide high-quality HF surrogate predictions with training that is largely based on LF simulation data.

We now present relative pressure and saturation error results for the 400 test samples. Fig. 15 displays histograms of these errors, for pressure and saturation, for LF simulation and both surrogate model results (these histograms correspond to the results shown in Figs. 6 and 7). Errors are all relative to HF simulation results. The orange histograms show errors for

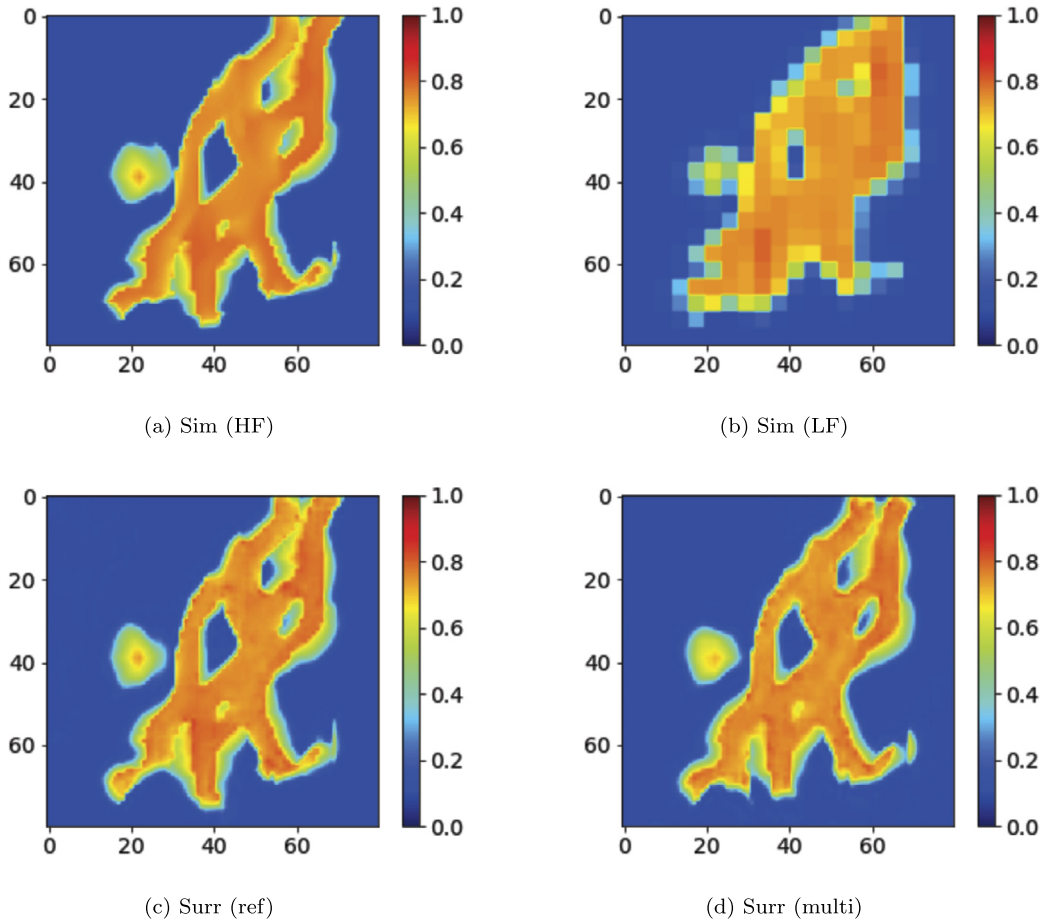


Fig. 14. Saturation maps from HF and LF simulation, reference surrogate, and transfer-learning-based surrogate trained with multifidelity data. Results are for layer 8 from realization in Fig. 4(c) at 1000 days.

the reference surrogate model predictions. These results are the most accurate, as expected, and display median pressure and saturation errors of 1.1% and 3.5%. The green histograms show relative errors for the LF simulation results. The median errors here are 2.7% and 19.4%. The errors for the surrogate model trained with multifidelity data (blue histograms) fall between the reference surrogate model errors and the LF simulation errors. Importantly, these errors (with medians 1.5% and 5.5%) are much closer to those of the reference surrogate model than to those from LF simulation. This again highlights the capabilities of the transfer-learning procedure.

4.5. Well rate statistics

Water injection rates and water and oil production rates are calculated from well-block pressure and saturation values and the specified BHP using Eqs. (4) and (5). We now present flow rate statistics evaluated over the 400 test samples. Fig. 16 presents flow statistics for oil and water production rates (OPR and WPR) of producers P1 and P4. Results for the P_{10} , P_{50} and P_{90} (10th, 50th and 90th percentile) rates are presented at $n_t = 10$ time steps. The red dashed lines in all subplots in Fig. 16 represent the P_{10} , P_{50} , P_{90} HF simulation results. The blue curves in the left column subplots show LF simulation results, while the black dash-dotted curves in the middle column present the reference surrogate results. The black solid lines in the right column display results for the transfer-learning-based surrogate trained with multifidelity data.

From the left-column subplots, it is clear that the LF simulation results display some error relative to the HF results. Water production rates are systematically under-predicted in Fig. 16(d) and (j), and oil rates show discrepancies in Fig. 16(a) and (g). The reference surrogate model (trained with HF simulation data) provides accurate flow statistics for all quantities considered. The results for the surrogate trained with multifidelity data (right column) are quite close to the reference surrogate results shown in the middle column. This demonstrates the ability of the transfer learning framework to provide high-quality predictions for key well rate quantities.

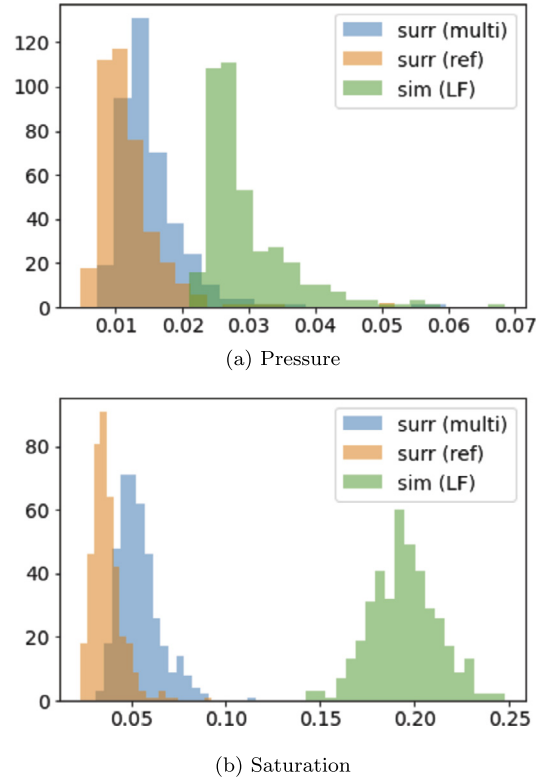


Fig. 15. Histograms of relative errors for pressure and saturation for the 400 test samples.

5. Data assimilation with surrogate model

In this section, we first describe the history matching framework, which involves the use of an ensemble smoother with multiple data assimilation (ESMDA) [30], with geomodels parameterized using CNN-PCA and flow responses determined using the transfer-learning-based surrogate model. We then present history matching results using this framework for a synthetic example.

5.1. ESMDA procedure

The ESMDA-based history matching procedure in this work is the same as in [2,3], except here we use the transfer-learning-based surrogate instead of the (reference) recurrent R-U-Net model. CNN-PCA enables us to generate HF geological realizations $\mathbf{m}_{\text{cnnpca}} \in \mathbb{R}^{n^h}$ from low-dimensional latent variables $\xi \in \mathbb{R}^{N_l}$, where N_l denotes the dimension of the latent variables. For the case considered in this work, $N_l = 400$, which is much less than the total number of HF grid blocks ($n^h = 128,000$). A geological realization in the history matching process is represented as $\mathbf{m}_{\text{cnnpca}}(\xi)$. The resulting flow predictions $\mathbf{d} \in \mathbb{R}^{N_{\text{hm}}}$ for the historical period, generated using the surrogate model, are denoted by $\mathbf{d} = \hat{f}(\mathbf{m}_{\text{cnnpca}}(\xi))$. Here N_{hm} denotes the number of observations in the history matching period.

In the data assimilation procedure, we first sample latent variables from their prior distribution $N(\mu_\xi, C_\xi)$, where μ_ξ and C_ξ denote the prior mean and covariance of ξ . ESMDA generates posterior distributions by assimilating data and updating the model parameters multiple times. The ESMDA update equation is

$$\xi_i^{k+1} = \xi_i^k + C_{\xi,d}^k (C_d^k + \alpha_k C_D)^{-1} (\mathbf{d}_{\text{obs}} + \sqrt{\alpha_k} \mathbf{e}_i^k - \mathbf{d}_i^k), \quad (21)$$

for $i = 1, \dots, N_r$ and $k = 1, \dots, N_a$, where N_r is the number of realizations considered and N_a is the number of data assimilation steps. Here α_k , $k = 1, \dots, N_a$, are the inflation coefficients, subject to $\sum_{k=1}^{N_a} \alpha_k^{-1} = 1$. The choice of N_a and α_k can affect the performance of the method. In this work, we specify $N_a = 10$ and use α_k values of 57.017, 35.0, 25.0, 20.0, 18.0, 15.0, 12.0, 8.0, 5.0, 3.0, as suggested in [31].

At each iteration k , we perturb the observation data $\mathbf{d}_{\text{obs}} \in \mathbb{R}^{N_{\text{hm}}}$ with $\mathbf{d}_{\text{obs}} + \sqrt{\alpha_k} \mathbf{e}_i^k$. The vector \mathbf{e} represents random noise sampled from $N(\mathbf{0}, C_D)$, where C_D is the covariance of the measurement error. The data in the historical period are generated through application of $\mathbf{d}_i^k = \hat{f}(\mathbf{m}_{\text{cnnpca}}(\xi_i^k))$. The covariance matrix $C_{\xi,d}^k$ and auto-covariance C_d^k are constructed from the data ensemble \mathbf{d}_i^k and model parameter ensemble ξ_i^k . After N_a iterations are completed, the posterior predictions \mathbf{d}_{post} are constructed via $\mathbf{d}_{\text{post}} = \hat{g}(\mathbf{m}_{\text{cnnpca}}(\xi_{\text{post}}))$.

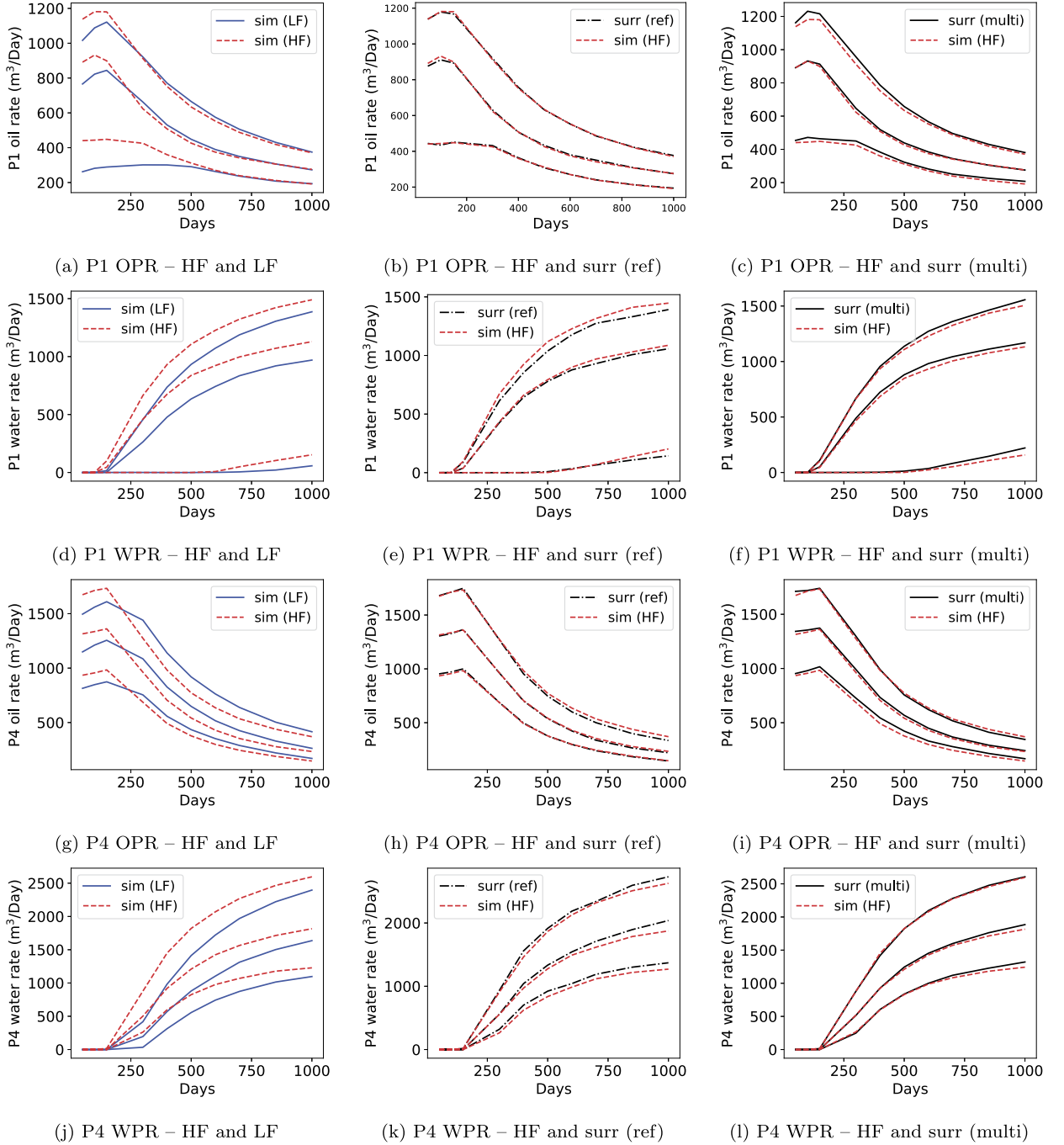


Fig. 16. Flow rate statistics for HF and LF simulation (left column), reference surrogate model (middle column), and surrogate model trained with multifidelity data (right column) for P1 and P4 oil and water production rates. Results correspond to P_{10} , P_{50} and P_{90} responses over 400 test cases.

5.2. History matching results

We now present history matching results using the recurrent R-U-Net surrogate model trained with transfer learning and multifidelity data. These predictions are compared to results using the reference surrogate model for the determination of $\mathbf{d}_i^k = \hat{f}(\mathbf{m}_{\text{cnpca}}(\xi_i^k))$. The geosim shown in Fig. 4(c) is taken to be the ‘true’ model. The observed data \mathbf{d}_{obs} are generated by performing high-fidelity simulation of this model, with noise added to represent measurement error. This noise is sampled from $N(\mathbf{0}, \mathbf{C}_D)$, with the standard deviation specified to be 5% of the simulated value. The observed data include the water and oil production rates for all five production wells at 150 and 300 days. We thus have $N_{\text{hm}} = 20$. We set $N_r = 400$.

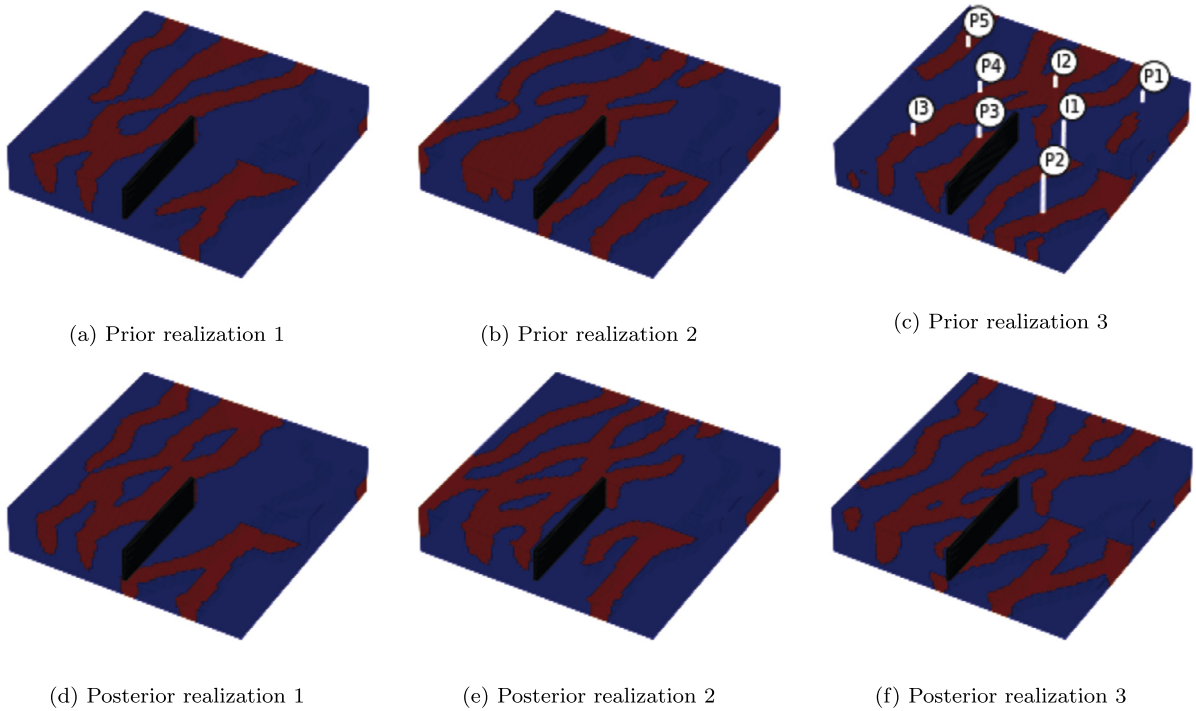


Fig. 17. Geomodel realizations for (a-c) three prior models and (d-f) corresponding posterior models. Well locations shown in (c).

The ESM DA history matching procedure provides a set of 400 posterior models. Three prior and (corresponding) posterior realizations are shown in Fig. 17. A degree of uncertainty reduction can be discerned in the well-to-well connectivity (through high-permeability channels) in the posterior models. For example, wells I1 and P2 are connected via channel sand in the true model. These two wells lack this connectivity in prior realization 3 (Fig. 17(c)), though it is introduced in posterior realization 3 (Fig. 17(f)).

Fig. 18 presents statistical results (over the 400 posterior samples) for water and oil production rates of wells P1, P4, and P5. These wells are representative of the overall system (the P2 flow rates resemble those of P1, and the P3 rates resemble those of P4). The prior P_{10} – P_{90} range for the reference high-fidelity simulation results is shown in each subplot as the gray-shaded region. The red dashed lines show the simulated flow rates of the ‘true’ model. The red circles present the observed data (which deviate from the red lines due to measurement error). The black dash-dotted lines indicate the P_{10} , P_{50} , P_{90} posterior results generated from the reference surrogate model and ESM DA. The black solid lines show the P_{10} , P_{50} , P_{90} posterior predictions from the surrogate model trained with multifidelity data.

In Fig. 18, we observe substantial uncertainty reduction in posterior flow rate predictions for some quantities (such as P1 water rate) and very little uncertainty reduction in other quantities (P4 water rate). The true-model well rates (red curves) are essentially within the posterior P_{10} – P_{90} ranges, as would be expected. Most importantly for current purposes, the posterior P_{10} , P_{50} , P_{90} results from the surrogate model trained with multifidelity data agree closely with the reference surrogate model predictions. There are some minor discrepancies between these two sets of results (e.g., at early time in the P_{10} prediction in Fig. 18(a)), but the differences in general are small, especially compared to the amount of uncertainty reduction achieved.

Flow simulation is now performed on the 400 posterior models found using ESM DA with the transfer-learning-based surrogate model. These simulation results are compared to surrogate model predictions, in terms of P_{10} , P_{50} , P_{90} flow responses, in Fig. 19. Note that prior results are not shown here, and the y-axis ranges differ from those in Fig. 18. There is slightly more error here than in Fig. 18, though the overall agreement is still quite satisfactory.

Finally, in Fig. 20, we show relative errors for pressure and saturation predictions for prior and posterior transfer-learning-based surrogate results. These results are of interest because they quantify global accuracy, in contrast to accuracy in (local) well rate quantities, as shown in Fig. 19. The blue boxes in Fig. 20 display the relative errors (compared to HF simulation) for pressure and saturation predictions for 400 prior samples, while the yellow boxes show these errors for the posterior results (box quantities are as described for Fig. 6). The errors in the prior models correspond to the histogram results in Fig. 15. The consistency between the prior and posterior errors indicates that the surrogate model retains accuracy during the data assimilation process.

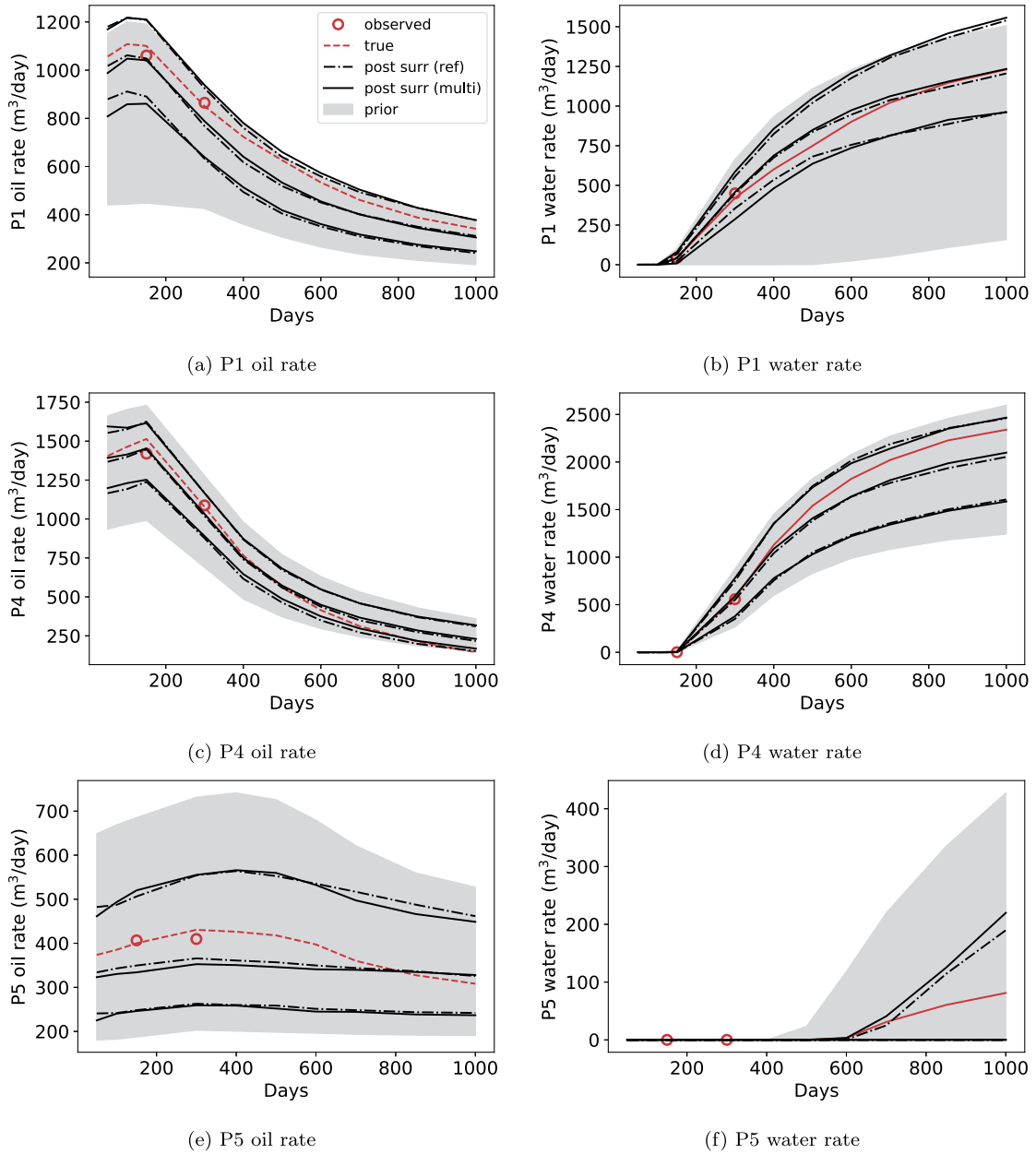


Fig. 18. Posterior results from reference surrogate model (black dash-dotted curves) and surrogate model trained with multifidelity data (black solid curves). Lower, middle and upper curves are P_{10} , P_{50} and P_{90} responses. Gray regions show the prior P_{10} – P_{90} range, and red circles and curves present observed and true data. Legend in (a) applies to all subplots.

6. Concluding remarks

In this work, a transfer-learning-based surrogate model that uses multifidelity training data was developed to approximate state variables and well rates in two-phase subsurface flow. The underlying network for this implementation is an existing 3D recurrent residual U-Net, which has been used previously for oil-water and CO₂ storage problems. The surrogate model is trained with both high and low-fidelity numerical simulation data. Most of the network parameters are first estimated from LF data, while output-layer training and network fine-tuning are accomplished with HF data. In the examples, we used 2500 LF simulations and 200 HF simulations. The network input in all cases is the high-fidelity (fine-scale) geomodel. The LF simulations are themselves performed using coarsened geomodels constructed from the original fine-scale geological description using a global flow-based transmissibility upscaling procedure. The multifidelity approach

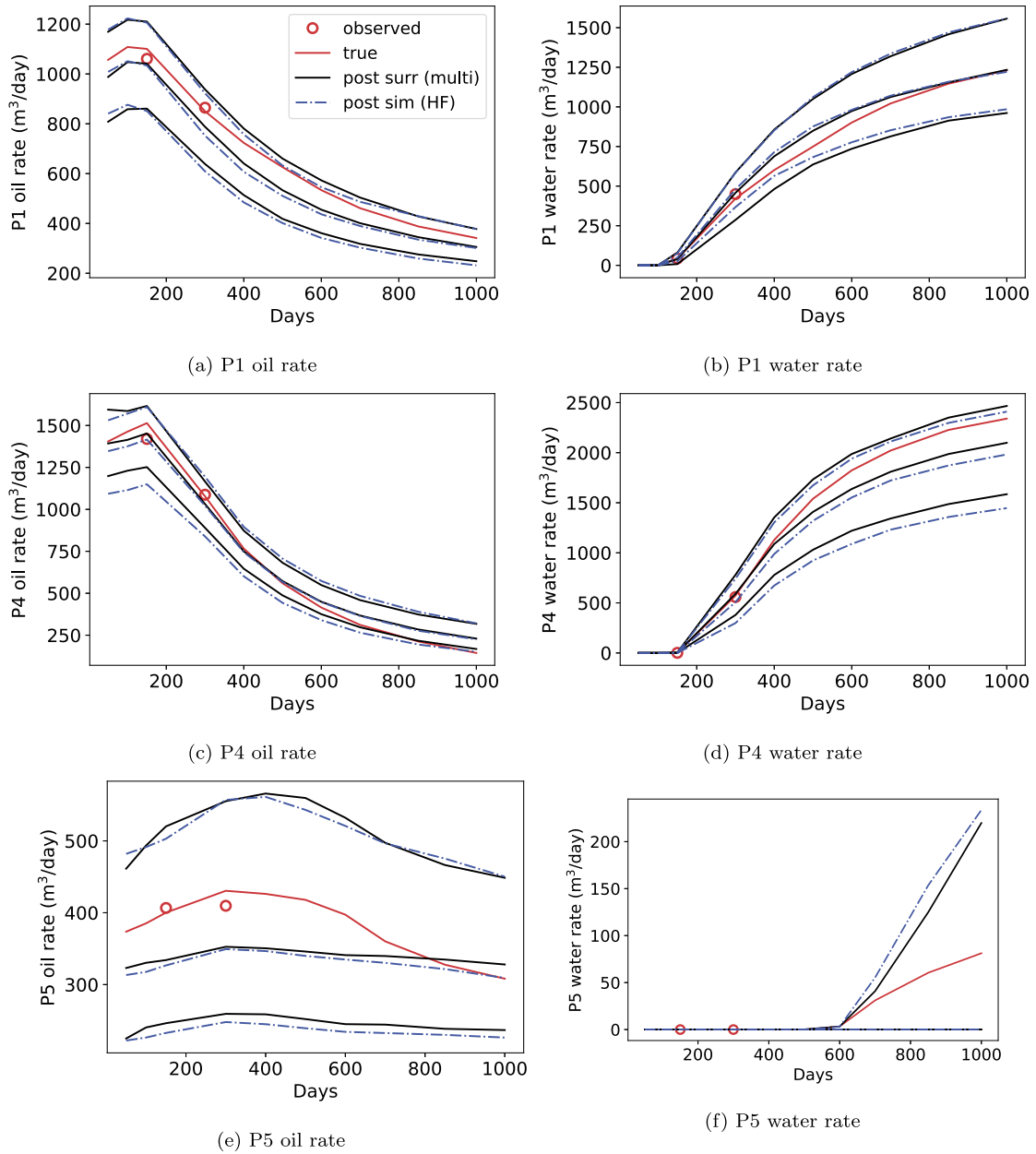


Fig. 19. Posterior results from surrogate model trained with multifidelity data (black solid curves) and corresponding fine-scale simulations (blue dash-dotted curves). Lower, middle and upper curves are P_{10} , P_{50} and P_{90} responses. Legend in (a) applies to all subplots.

provided about a 90% savings in training computations and a 40% savings in (GPU) training time compared to using only HF simulation data.

The transfer-learning-based surrogate model was applied for two-phase oil-water flow in 3D channelized systems, with eight wells operating under bottom-hole pressure control. The surrogate model was evaluated over a test set of 400 HF models. A reference surrogate model, in which all training runs were performed at high fidelity (the same 2500 data samples were used), was also constructed. The transfer-learning-based surrogate model was shown to provide accurate predictions for dynamic pressure and saturation fields, with errors very near those obtained by the reference surrogate model. Specifically, the median relative errors for pressure and saturation with the transfer-learning-based surrogate were 1.5% and 5.5%, compared to 1.1% and 3.5% for the reference surrogate. This represents much greater accuracy than that achieved in the LF numerical simulations, where the median saturation error was 19.4%. Close agreement in P_{10} , P_{50} and P_{90} well rates between the transfer-learning-based surrogate predictions and HF simulations was also demonstrated.

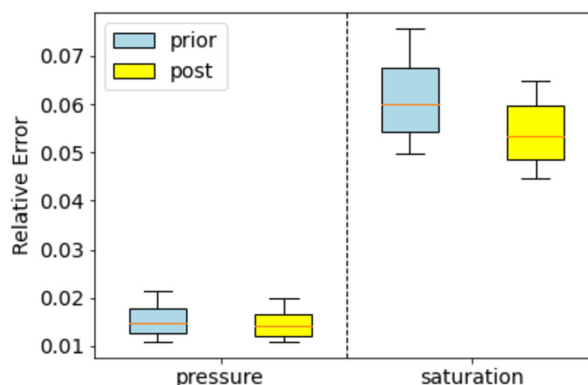


Fig. 20. Relative errors for prior and posterior pressure and saturation predictions for surrogate model trained with multifidelity data. Errors are for 400 realizations and are relative to HF simulation results.

The surrogate model was then combined with ESMDA for use in history matching. The geological models were parameterized using the 3D CNN-PCA representation, and flow predictions were generated using the transfer-learning-based surrogate model. The overall procedure provided clear uncertainty reduction, and comparisons of flow predictions for posterior models demonstrated reasonable agreement between surrogate and numerical simulation results for P_{10} , P_{50} and P_{90} well rate quantities.

There are a number of topics for future research in this area. In this work, we used simulation data from only two fidelity levels. It may be possible to improve the performance of the surrogate model, and/or reduce computational demands, through use of data at several fidelity levels. The transfer-learning-based framework can be extended to treat larger and more complicated models. This includes coupled flow and geomechanics, which is important in CO_2 storage settings. It will also be of interest to incorporate physical loss in the training process. This could act to further reduce the number of high-fidelity training simulations required. Finally, the general framework could be extended to treat problems involving variable well controls and well locations, which would enable its use for a wide range of optimization problems.

CRediT authorship contribution statement

Su Jiang: Conceptualization, Methodology, Software, Visualization, Writing – original draft. **Louis J. Durlofsky:** Conceptualization, Supervision, Writing – review & 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.

Data availability

The code used in this study is available at <https://github.com/sujiang94/surrogate-transfer-learning>.

Acknowledgements

We are grateful to the Stanford Smart Fields Consortium for partial funding of this work. We thank Meng Tang for providing geological models, parameterization code and recurrent R-U-Net code, and Dylan Crain for providing the upscaling code used in this study.

References

- [1] M. Tang, Y. Liu, L.J. Durlofsky, A deep-learning-based surrogate model for data assimilation in dynamic subsurface flow problems, *J. Comput. Phys.* 413 (2020) 109456.
- [2] M. Tang, Y. Liu, L.J. Durlofsky, Deep-learning-based surrogate flow modeling and geological parameterization for data assimilation in 3D subsurface flow, *Comput. Methods Appl. Mech. Eng.* 376 (2021) 113636.
- [3] M. Tang, Y. Liu, L.J. Durlofsky, History matching complex 3D systems using deep-learning-based surrogate flow modeling and CNN-PCA geological parameterization, in: *SPE Reservoir Simulation Conference, OnePetro*, 2021.
- [4] M. Tang, X. Ju, L.J. Durlofsky, Deep-learning-based coupled flow-geomechanics surrogate model for CO_2 sequestration, *Int. J. Greenh. Gas Control* 118 (2022) 103692.
- [5] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *J. Comput. Phys.* 378 (2019) 686–707.
- [6] N. Wang, D. Zhang, H. Chang, H. Li, Deep learning of subsurface flow via theory-guided neural network, *J. Hydrol.* 584 (2020) 124700.

- [7] N. Wang, H. Chang, D. Zhang, Efficient uncertainty quantification for dynamic subsurface flow with surrogate by theory-guided neural network, *Comput. Methods Appl. Mech. Eng.* 373 (2021) 113492.
- [8] N. Wang, H. Chang, D. Zhang, Theory-guided auto-encoder for surrogate construction and inverse modeling, *Comput. Methods Appl. Mech. Eng.* 385 (2021) 114037.
- [9] Q. He, D. Barajas-Solano, G. Tartakovsky, A.M. Tartakovsky, Physics-informed neural networks for multiphysics data assimilation with application to subsurface transport, *Adv. Water Resour.* 141 (2020) 103610.
- [10] A.M. Tartakovsky, C.O. Marrero, P. Perdikaris, G.D. Tartakovsky, D. Barajas-Solano, Physics-informed deep neural networks for learning parameters and constitutive relationships in subsurface flow problems, *Water Resour. Res.* 56 (2020) e2019WR026731.
- [11] A.M. Tartakovsky, D.A. Barajas-Solano, Q. He, Physics-informed machine learning with conditional Karhunen-Loeve expansions, *J. Comput. Phys.* 426 (2021) 109904.
- [12] Y. Zhu, N. Zabarar, Bayesian deep convolutional encoder-decoder networks for surrogate modeling and uncertainty quantification, *J. Comput. Phys.* 366 (2018) 415–447.
- [13] S. Mo, Y. Zhu, N. Zabarar, X. Shi, J. Wu, Deep convolutional encoder-decoder networks for uncertainty quantification of dynamic multiphase flow in heterogeneous media, *Water Resour. Res.* 55 (2019) 703–728.
- [14] S. Mo, N. Zabarar, X. Shi, J. Wu, Deep autoregressive neural networks for high-dimensional inverse problems in groundwater contaminant source identification, *Water Resour. Res.* 55 (2019) 3856–3881.
- [15] S. Mo, N. Zabarar, X. Shi, J. Wu, Integration of adversarial autoencoders with residual dense convolutional networks for estimation of non-Gaussian hydraulic conductivities, *Water Resour. Res.* 56 (2020) e2019WR026082.
- [16] G. Wen, M. Tang, S.M. Benson, Towards a predictor for CO₂ plume migration using deep neural networks, *Int. J. Greenh. Gas Control* 105 (2021) 103223.
- [17] N. Geneva, N. Zabarar, Multi-fidelity generative deep learning turbulent flows, *Found. Data Sci.* 2 (2020) 391.
- [18] X. Meng, G.E. Karniadakis, A composite neural network that learns from multi-fidelity data: application to function approximation and inverse PDE problems, *J. Comput. Phys.* 401 (2020) 109020.
- [19] S. De, J. Britton, M. Reynolds, R. Skinner, K. Jansen, A. Doostan, On transfer learning of neural networks using bi-fidelity data for uncertainty propagation, *Int. J. Uncertain. Quantificat.* 10 (2020) 543–573.
- [20] D.H. Song, D.M. Tartakovsky, Transfer learning on multi-fidelity data, *J. Mach. Learn. Model. Comput.* 2 (2022) 31–47.
- [21] Y. Zhou, Parallel General-Purpose Reservoir Simulation with Coupled Reservoir Models and Multisegment Wells, Ph.D. thesis, Stanford University, 2012.
- [22] D.W. Peaceman, Interpretation of well-block pressures in numerical reservoir simulation with nonsquare grid blocks and anisotropic permeability, *SPE J.* 23 (1983) 531–543.
- [23] P. Zhang, G.E. Pickup, M.A. Christie, A new practical method for upscaling in highly heterogeneous reservoir models, *SPE J.* 13 (2008) 68–76.
- [24] Y. Chen, B.T. Mallison, L.J. Durlofsky, Nonlinear two-point flux approximation for modeling full-tensor effects in subsurface flow simulations, *Comput. Geosci.* 12 (2008) 317–335.
- [25] D. Crain, Extended Framework for Multifidelity Uncertainty Quantification in Subsurface Flow Systems, Master's thesis, Stanford University, 2020.
- [26] S. Jiang, L.J. Durlofsky, Treatment of model error in subsurface flow history matching using a data-space method, *J. Hydrol.* 603 (2021) 127063.
- [27] Y. Liu, L.J. Durlofsky, 3D CNN-PCA: a deep-learning-based parameterization for complex geomodels, *Comput. Geosci.* 148 (2021) 104676.
- [28] D.P. Kingma, J. Ba Adam, A method for stochastic optimization, arXiv preprint, arXiv:1412.6980, 2014.
- [29] M.G. Shirangi, L.J. Durlofsky, A general method to select representative models for decision making and optimization under uncertainty, *Comput. Geosci.* 96 (2016) 109–123.
- [30] A.A. Emerick, A.C. Reynolds, Ensemble smoother with multiple data assimilation, *Comput. Geosci.* 55 (2013) 3–15.
- [31] A.A. Emerick, A.C. Reynolds, Investigation of the sampling performance of ensemble-based methods with a simple reservoir model, *Comput. Geosci.* 17 (2013) 325–350.