



# Extension of Fourier Neural Operator from Three-Dimensional $(x, y, t)$ to Four-Dimensional $(x, y, z, t)$ Subsurface Flow Simulation

Jianqiao Liu<sup>1</sup> · Huanquan Pan<sup>1</sup> · Wenyue Sun<sup>2</sup> · Hongbin Jing<sup>1</sup> · Bin Gong<sup>1</sup>

Received: 4 November 2023 / Accepted: 4 July 2024  
© International Association for Mathematical Geosciences 2024

## Abstract

Numerical simulation of subsurface flow in porous media is crucial for various geoscience applications. However, conducting numerical simulations for such problems, particularly in four dimensions  $(x, y, z, t)$ , presents significant computational challenges and cost. These challenges are primarily due to the highly nonlinear governing partial differential equations (PDEs) and the need for refined mesh discretization. Deep learning methods offer promising alternatives to traditional simulators by leveraging neural network models. Previous studies have demonstrated that the surrogate model constructed using a Fourier neural operator (FNO) exhibits superior speed, accuracy, and data efficiency in addressing three-dimensional  $(x, y, t)$  problems compared to mainstream machine learning methods. However, applying the existing FNO algorithm to four-dimensional problems is hindered by the large number of network parameters and high GPU memory consumption. In this study, we propose a novel framework for dynamically predicting four-dimensional subsurface flow properties utilizing the FNO network and the domain decomposition method. Our approach leverages the three-dimensional FNO and time components in the  $x$  and  $y$  directions of each  $z$ -layer. The predicted results for subsequent time steps in four dimensions are generated by consolidating the results across all  $z$ -layer dimensions. We use examples of CO<sub>2</sub> injection for enhanced oil recovery (EOR) in compositional simulations for fracture systems and apply the deep learning framework that is trained from physics-based simulation data to emulate the complex flow state. Furthermore, we integrate our framework with a multilayer perceptron (MLP) network capable of processing sequential data to predict gas production under variable well control in a realistic reservoir. The key contribution of this work is the enhanced training and prediction efficiency of the FNO method for large-scale four-dimensional heterogeneous reservoir models. Consequently, our

---

✉ Hongbin Jing  
hongbinjin@cug.edu.cn

Jianqiao Liu  
jqliu@cug.edu.cn

<sup>1</sup> Department of Petroleum Engineering, China University of Geosciences (Wuhan), Wuhan, China

<sup>2</sup> China University of Petroleum (East China), Qingdao, China

approach can be further extended to expedite workflows for reservoir optimization in closed-loop reservoir management.

**Keywords** Fourier neural operator · Simulation · Subsurface flow · Deep neural network

## 1 Introduction

Simulation of complex physical systems described by nonlinear partial differential equations (PDEs) is central to reservoir engineering. The governing equations for subsurface flow in porous media involve multiphase flow equations, and they may be coupled with other physical laws to describe geomechanical and thermal effects (Ajayi and Gupta 2019). Conventional numerical simulations employ finite element methods (FEM) and finite difference methods (FDM) to solve these PDEs through discretization (Wen and Benson 2021; Benson and Cole 2008; Benson et al. 2012). However, the combination of discretization schemes with time-stepping iteration strategies is time-consuming, particularly due to the multi-physics, nonlinearity, and multi-scale nature of these problems. Large-scale reservoir development projects require extensive computational effort for uncertainty quantification, history matching, and development optimization, which requires a large number of numerical simulations. Hence, leveraging the latest advancements in artificial intelligence to develop surrogate models for the oil industry (Bagheri and Riahi 2015; Rahmanifard and Plaksina 2019; Bagheri and Rezaei 2019; Shi et al. 2020; Zare et al. 2020; Moosavi et al. 2022; Li et al. 2024) can enhance the traditional iterative process of solving partial differential equations, leading to a notable reduction in computational costs and a faster workflow..

The surrogate model, based on deep learning methods, can achieve real-time predictions once trained with a reasonable amount of data from full-physics simulations (Zhu and Zabarar 2018; Mo et al. 2019; Tang et al. 2020; Wen et al. 2021). Numerous deep learning techniques have been proposed to construct fast and accurate alternative models for numerical reservoir simulations.

Among these techniques, pure data-driven models, based on convolutional neural networks (CNN), focus on learning mappings in Euclidean space from traditional numerical simulation data (Mo et al. 2019; Liu et al. 2023; Fu et al. 2023). In this approach, surrogate models are constructed to rapidly predict pressure and saturation fields, as well as wellbore flow rates. These models are employed for history matching, well control optimizations, uncertainty quantification, and various other applications (Tang et al. 2021; Wang et al. 2021; Huang et al. 2023; Zhong et al. 2019; Liu et al. 2019; Kim and Durlafsky 2021). To date, this method has successfully predicted multiphase flow in three-dimensional reservoirs (Jiang et al. 2021; Tang et al. 2021; Wen et al. 2021; Wu and Qiao 2021). However, the mapping of data-driven finite-dimensional operators between finite-dimensional spaces is prone to overfitting and may be inconsistent with the governing physics laws (Anandkumar et al. 2020; Li et al. 2023). It requires a substantial number of numerical simulation results as input data, which may be unmanageable as the dimension of the problem increases (Wen et al. 2022; Zhang et al. 2023). Additionally, the data-driven approach only produces

valid results for specific spatial and temporal meshes, limiting its practical applications for reservoir simulation on an industrial grid scale of tens of millions (Li et al. 2020b; Nasir and Durllofsky 2023; Jiang and Durllofsky 2023).

Physics-informed or physics-constrained deep learning methods embed the governing equations in the loss function of neural networks (Haghighat and Juanes 2021; Raissi et al. 2019; Zhu et al. 2019). One popular approach is the physics-informed neural network (PINN) method, which integrates domain knowledge and physical principles into the architecture, thereby transforming the traditional numerical simulation process into an optimization task. A key advantage of this method is its ability to make accurate predictions for PDE solutions even with limited labeled data (Raissi et al. 2019). Researchers have employed the PINN method to solve the one-dimensional Buckley–Leverett problem involving two-phase flow in porous media (Fuks and Tchelepi 2020), simulate the two-dimensional single-phase groundwater flow process (Wang et al. 2021), and forecast variables such as pressure, gas saturation, the water extraction rate in CO<sub>2</sub> EOR (Shokouhi et al. 2020) process. However, merely penalizing deviations from the governing physics equations or domain knowledge by adding hard physical constraints to the loss function is not suitable for large-scale spatiotemporal domain simulation problems and struggles with complex PDEs (Zhang et al. 2022). For PINN and its extension methods, changes in the parameters of the equations or the initial conditions require the model to be retrained. Pang et al. (2019); Yu et al. (2022); Wu et al. (2023).

In recent years, the deep neural operator has emerged as a novel approach designed to learn the infinite-dimensional mapping from PDE solution (Bhattacharya et al. 2021; Lu et al. 2019; Li et al. 2020c). Unlike other deep learning methods such as CNN and PINNs, the deep neural operator is aimed at learning a family of PDEs. The trained model can achieve better generalization under varying PDE parameters and initial boundary conditions. The surrogate model established by this method has demonstrated promising potential for applications in reservoir engineering. However, due to the high cost of evaluating global neural integral operators, previously proposed deep neural operators have not yet reached the desired computational efficiency. The Fourier neural operator (FNO) is a new framework for solving PDEs, proposed by NVIDIA's machine learning group in the 2021 International Conference on Learning Representations (ICLR) (Li et al. 2020b). FNO enhances the computational efficiency of previous deep neural operators through two-dimensional fast Fourier transform (FFT). Zhang et al. (2022) first evaluated the potential of the FNO surrogate model for two-dimensional oil–water subsurface flow. Wen et al. (2022) evaluated the potential of the FNO surrogate model for two-dimensional CO<sub>2</sub>–water flows. Yan et al. (2022a) developed a deep learning workflow based on FNO to predict pressure distributions and CO<sub>2</sub> plumes during injection and post-injection periods of geological CO<sub>2</sub> sequestration (GCS) operations. Kuang et al. (2023) recently compared the performance of FNO and U-net for well rate forecasting in a two-dimensional fractured reservoir, and observed that the FNO-based approach showed faster training speed, higher accuracy, and more robust ability to describe fracture features.

For three-dimensional geometric systems  $(x, y, z)$ , FNO suffers from high GPU memory consumption and computational cost due to a large number of model parameters, along with the use of computationally expensive three-dimensional FFT

operations (Jiang et al. 2023). Wen et al. (2023) proposed a nested FNO architecture, which integrates the FNO with a semi-adaptive local grid refinement (LGR) modeling method for numerical simulations. This approach significantly reduces the computational costs associated with data collection in simulating four-dimensional problems. Furthermore, Yan et al. (2022b) developed a deep learning workflow that employs FNOs alongside feature coarsening methods, which improved the prediction accuracy of large-scale geological CO<sub>2</sub> storage models, reducing the GPU memory requirements and enhancing the training efficiency. Overall, however, limited work has been done to reduce the time consumption of FNO during the training process for subsurface flow problems when dealing with four-dimensional spatiotemporal state variables.

In this work, to improve the computational speed of the FNO in a four-dimensional spatiotemporal reservoir, we developed a new framework that can simulate four-dimensional subsurface flows based on the three-dimensional FNO network with a domain decomposition method. This new framework decomposes the four-dimensional spatiotemporal domain in the  $z$ -dimension, trains the FNO network independently in the three-dimensional spatiotemporal domain in each  $z$ -layer, and predicts the distribution of pressure and flow fields of the four-dimensional spatiotemporal domain by combining the three-dimensional domain ( $x, y, t$ ) results in all  $z$ -layers. This method employs three-dimensional FNOs along the  $x$ -axis and  $y$ -axis and incorporates the time component, thus circumventing the substantial computational expenses associated with direct predictions of four-dimensional flow state variables using existing FNOs. To demonstrate the applicability of our proposed approach to large and complex reservoirs, we conducted several tests, which included injecting CO<sub>2</sub> into a reservoir with complex fractures in case 2 to predict the time-series flow state distribution. Additionally, in another real reservoir in case 3, we successfully established a surrogate model for well rate production under time-varying well conditions.

The remainder of this paper is organized as follows. Section 2 describes a new framework that extends FNO-net from the three-dimensional spatiotemporal domain to the four-dimensional spatiotemporal domain. Section 3 describes three test cases and discusses the corresponding results. In Sect. 4, the applications and potential extensions of the proposed approach are discussed. Section 5 concludes this study. The detailed loss functions for boundary conditions and initial conditions are provided in the Appendix.

## 2 Methodology

In this section, firstly, we describe the subsurface flow governing equations in porous media and the FNO-net architecture. Then, we introduce a novel methodology that simulates four-dimensional subsurface flow based on the FNO and domain decomposition method. The workflow of this new method, model training details, loss function design, and error measurement method are also illustrated. Compositional simulation

is used in the study, and the governing equation for each component is written as

$$\frac{\partial}{\partial t} \left( \phi \sum_j (x_{i,j} \rho_j S_j) \right) + \nabla \sum_j (x_{i,j} \rho_j q_j) + \sum_j x_{i,j} \rho_j Q_j = 0, \tag{1}$$

where subscript  $i$  denotes the primary fluid components,  $j$  denotes the fluid phase,  $t$  represents time,  $\phi$  is the porosity,  $x_{ij}$  is the concentration of component  $i$  in phase  $j$ ,  $\rho_j$  is the fluid phase density,  $S_j$  is the phase saturation,  $Q_j$  is the source term,  $\mu_j$  is the viscosity of phase  $j$ , and  $q_j$  is the Darcy flux of phase  $j$ . Darcy’s law for each phase flow can be written as

$$q_j = -\mathbf{k} \frac{k_{rj}}{\mu_j} (\nabla P - \rho_j g \nabla Z), \tag{2}$$

where  $k_{rj}$  is the relative permeability of phase  $j$  that is a function of  $S_j$ ,  $P$  is pressure (capillary pressure is ignored in this study),  $g$  is the gravity constant, and  $Z$  represents depth.

Furthermore, we utilize an embedded discrete fracture model (EDFM) to simulate the fluid flow process in heterogeneous reservoirs with complex fracture distributions. Initially introduced by Li and Lee (2008), the EDFM has been further developed in subsequent studies by Moïnfar et al. (2014) and Tene et al. (2017). The fluid cross-flow between the porous matrix and fractures is described in this study by the following equation (Zeng et al. 2019)

$$q_{mf} = \frac{2l_{mf} (\mathbf{K}_m \cdot \mathbf{n}) \cdot \mathbf{n}}{\mu} \nabla P, \tag{3}$$

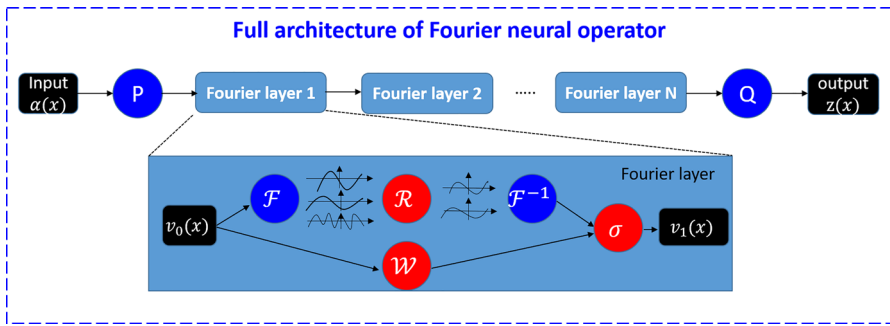
where  $\mathbf{n}$  is the normal unit vector of the fracture element,  $\mathbf{K}_m$  represents the absolute permeability tensor in the global meshes,  $l_{mf}$  is the length between the fracture and mesh matrix, and  $\mu$  is the fluid viscosity. In the EDFM, the fracture element and matrix mesh are defined in two systems, and there is no requirement to generate complex meshes, which alleviates the computational cost for complex fracture modeling. Within a physics-driven reservoir simulator, the state variables such as pressure and saturation are computed through an iterative process.

### 2.1 FNO Architecture

The FNO operator is a newly proposed neural operator whose main objective is to learn the mapping between two infinite-dimensional spaces from the observed finite input–output pairs by the Fourier transform (Li et al. 2020a). The mapping is

$$G : A \rightarrow U, \tag{4}$$

where  $A$  is the input function space, and  $U$  is the output function space. As shown in Fig. 1, the FNO approximation of this mapping mainly includes the following



**Fig. 1** Full architecture of FNO: input PDE properties  $a(x)$  go through fully connected network  $P$  to  $v_0(x)$  and then go through the iteration in the Fourier layers to obtain  $v_{n+1}(x)$ , and finally go through  $Q$  to the spatial dimension of the solution  $z(x)$

three steps: (1) The input observation  $a(x)$  is lifted to a potential representation  $v_0$  in higher-dimensional space through a simple shallow fully connected network  $v_0(x) = P(a(x))$ . (2) Iterative Fourier layers are implemented, namely  $v_0 \rightarrow v_1 \rightarrow v_2 \rightarrow \dots \rightarrow v_N$ . The concrete iterative process between elements of this sequence  $v_n \rightarrow v_{n+1}$  can be written as

$$v_{n+1}(x) = \sigma(Wv_n(x) + (K(a; \phi)v_n(x))(x)), \tag{5}$$

where  $\sigma$  is the activation function,  $W$  is the network weight, and  $(K(a; \phi)v_n(x))(x)$  is the bias. The kernel integral transformation parameterized in a neural network is denoted as  $K$ . The FNO kernel operator can be written as

$$(K(a; \phi)v_i(x))(x) = F^{-1}(R_\phi \cdot (Fv_k))(x), \tag{6}$$

where  $F$  denotes the Fourier transform and  $F^{-1}$  denotes the inverse Fourier transform. The linear transform  $R_\phi$  represents a restriction operator which can filter the higher models in Fourier space. (3) After  $N$  Fourier layer iterations, the output  $z(x)$  can be obtained by the fully connected neural network transformation  $Q$ ,  $z(x) = Q(v_N(x))$ . The full architecture of the FNO is shown in Fig. 1 and is described in detail in the original publication Li et al. (2021).

## 2.2 New Methodology to Extend FNO-Net from Three Dimensions $(x, y, t)$ to Four Dimensions $(x, y, z, t)$

In this section, to circumvent the computational burden incurred by directly predicting four-dimensional flow state variables, our framework proposes the application of three-dimensional FNO across the  $x$ -axis and  $y$ -axis, together with the time component. The detailed procedure for the new framework is as follows: (1) Decompose the four-dimensional spatiotemporal domain in the  $z$ -dimension. (2) Train multiple FNO-net in the three-dimensional spatiotemporal domain. (3) Predict the state variables in

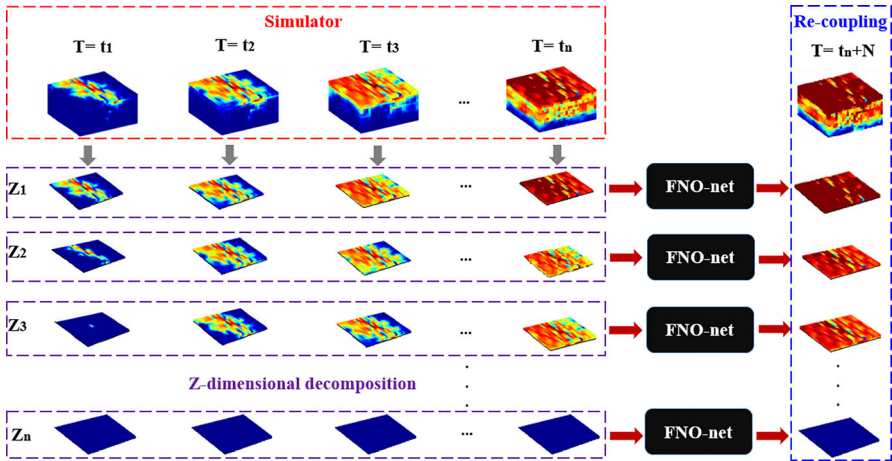


Fig. 2 Framework of new methodology to extend FNO-net from three-dimensional  $(x, y, t)$  to four-dimensional  $(x, y, z, t)$

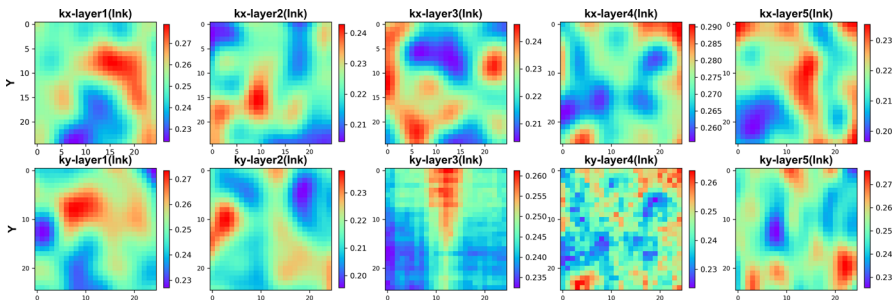


Fig. 3 Permeability fields  $k_x, k_y$  (layer1 to layer5)

subsequent time steps for the entire three-dimensional domain. (4) Obtain the four-dimensional spatiotemporal solution by re-coupling the three-dimensional prediction results in the  $z$ -dimension.

The whole framework can be divided into the following three parts in Fig. 2. Part 1 (red box in Fig. 2) illustrates the solution of global governing equations. After running numerical simulations, we obtain the simulation results for  $n$  time steps  $(M_{l(x,y,z,t_1)}, M_{l(x,y,z,t_2)}, M_{l(x,y,z,t_3)}, \dots, M_{l(x,y,z,t_n)})$ . Part 2 (purple box in Fig. 2) depicts the decomposition of the simulation results across  $n$  time steps in the  $z$ -dimension. The decomposition process can be expressed as

$$\begin{aligned}
 M_{l(x,y,z_1,t)} &:: M_{l(x,y,z_1,t_1)}, M_{l(x,y,z_1,t_2)}, M_{l(x,y,z_1,t_3)}, \dots, M_{l(x,y,z_1,t_n)} \\
 M_{l(x,y,z_2,t)} &:: M_{l(x,y,z_2,t_1)}, M_{l(x,y,z_2,t_2)}, M_{l(x,y,z_2,t_3)}, \dots, M_{l(x,y,z_2,t_n)} \\
 M_{l(x,y,z_3,t)} &:: M_{l(x,y,z_3,t_1)}, M_{l(x,y,z_3,t_2)}, M_{l(x,y,z_3,t_3)}, \dots, M_{l(x,y,z_3,t_n)} \\
 &\vdots \\
 M_{l(x,y,z_n,t)} &:: M_{l(x,y,z_n,t_1)}, M_{l(x,y,z_n,t_2)}, M_{l(x,y,z_n,t_3)}, \dots, M_{l(x,y,z_n,t_n)},
 \end{aligned} \tag{7}$$

where  $M_l$  denotes the flow field distributions of pressure, phase saturation, and component concentration in each phase;  $::$  denotes an operator of the domain decomposition which can decompose the simulation results of  $n$  time steps in the  $z$ -dimension. After domain decomposition in the  $z$ -dimension at every time step, we can obtain a simulation results matrix  $R$ . The matrix can be written as

$$R = \begin{bmatrix} M_l(x,y,z_1,t_1), M_l(x,y,z_1,t_2), M_l(x,y,z_1,t_3) \cdots M_l(x,y,z_1,t_n) \\ M_l(x,y,z_2,t_1), M_l(x,y,z_2,t_2), M_l(x,y,z_2,t_3) \cdots M_l(x,y,z_2,t_n) \\ M_l(x,y,z_3,t_1), M_l(x,y,z_3,t_2), M_l(x,y,z_3,t_3) \cdots M_l(x,y,z_3,t_n) \\ \vdots \\ M_l(x,y,z_n,t_1), M_l(x,y,z_n,t_2), M_l(x,y,z_n,t_3) \cdots M_l(x,y,z_n,t_n) \end{bmatrix}. \tag{8}$$

The first row  $R$  in Eq. 8 denotes the first-layer results for  $n$  time steps. The second row of  $R$  contains the second-layer results for  $n$  time steps, and so on. Each column of  $R$  contains the simulation results from time step  $t_1$  to time step  $t_n$  that can be used as input data to train a three-dimensional FNO-net and predict the subsequent time step state variables at  $t_n + N$  ( $N$  is the number of predicted time steps). The training and prediction process can be written as

$$FNO_i (R(i, :)^{t_1-t_n}) = \delta_i^{t_n+N}, \tag{9}$$

where  $R(i, :)$  are the simulation results for the  $i$ th layer. In the training process,  $R(i, :)$  is first divided into a training dataset and a test dataset. Then these datasets are used as input data to the  $i$ th FNO-net ( $FNO_i$ ) for training. Part 3 (blue box in Fig. 2) shows the re-coupling module for reassembling the prediction results of  $t_n + N$  time steps of each layer.

After FNO predicts the resulting state distribution at  $t_n + N$  for each layer, we collect the solutions for all layers and reconstruct the three-dimensional state distribution. The re-coupling process is

$$\mathcal{C} : \left\{ \delta_1^{t_n+N}, \delta_2^{t_n+N}, \delta_3^{t_n+N} \dots \delta_n^{t_n+N} \right\} = \delta^{t_n+N}, \tag{10}$$

where  $\mathcal{C}$  represents the process of re-coupling the predicted results in  $z$ -dimension, and  $\delta^{t_n+N}$  denotes the final results of four-dimensional subsurface flow distributions. In solving physical equations to obtain the training data needed for our model (state distribution data from previous time steps), the simulator already takes into account both vertical and cross-flows. Therefore, hierarchical training of FNO can capture the change of  $z$ -direction during the prediction process.

### 2.3 Loss Function Design and Evaluation Metrics

The total loss function employed in this study comprises three components: (1)  $L_2$ -loss, (2) boundary condition (BC) loss, and (3) initial condition (IC) loss. The total



loss function is formulated as

$$\text{Loss}_{\text{total}} = L_2(u, \hat{u}) + \lambda_1 \text{Loss}_{BC} + \lambda_2 \text{Loss}_{IC}. \tag{11}$$

In our study, the reservoir is modeled with no-flow boundary conditions. Both BC and IC can be easily computed via penalty terms due to their straightforward structure. Details of the loss functions for boundary conditions and initial conditions are provided in the Appendix. The  $L_2$ -loss used to train the deep learning models is defined as Pecha and Horák (2018)

$$L_2(u, \hat{u}) = \frac{\|u - \hat{u}\|_2}{\|u\|_2} + \beta \frac{\|du/dr - d\hat{u}/dr\|_2}{\|du/dr\|_2}, \tag{12}$$

where  $\hat{u}$  denotes the predicted output values, which can be a matrix, and  $du/dr$  denotes the first derivative of the input data;  $d\hat{u}/dr$  is the first derivative of the predicted output values,  $\|\cdot\|_2$  is  $L_2$  norm, and  $\beta$  is a hyperparameter of the network.

Now, we introduce several error metrics which will be used to evaluate the performance of the model proposed in this work. The relative error for predicted results is defined as

$$A_i = \frac{1}{T} \frac{1}{N} \sum_{t=1}^T \sum_{n=1}^N |u_i - \hat{u}_i|, \tag{13}$$

$$a_i = \frac{1}{T} \frac{1}{N} \sum_{t=1}^T \sum_{n=1}^N |u_i|, \tag{14}$$

$$\text{RE}_i = \frac{A_i}{a_i}, \tag{15}$$

where  $T$  represents the number of simulated time steps,  $N$  represents the number of grids,  $u_i$  represents the true value of pressure, phase saturation, component concentration in one phase, and so on, and  $\hat{u}_i$  represents the corresponding predicted value in the output. The variable  $A_i$  is the average absolute error,  $a_i$  is the average value of the actual value, and  $\text{RE}_i$  represents the relative error.

### 3 Application Cases

In this section, we demonstrate the performance of our new framework across three cases. The first and second cases involve complex CO<sub>2</sub> injection for enhanced oil recovery. The first case studies a three-dimensional heterogeneous reservoir measuring 250 feet × 250 feet × 50 feet with an eight-component fluid. The second case is a three-dimensional heterogeneous fractured reservoir, measuring 1,280 feet × 1,280 feet × 50 feet, with the same fluid components. The number of fractures in this case is 42. The FNO-net structure for each layer of training is illustrated in Table 7 for our two test cases. The third case focuses on dynamic production prediction under varying

**Table 1** Effect of the different number of Fourier layers

Linear layer	1 Fourier layer	3 Fourier layers	5 Fourier layers	7 Fourier layers	9 Fourier layers	Linear layer	ARE (%)	Per Epoch Time(s)
✓	✓					✓	3.2%	13
✓		✓				✓	2.2%	13.5
✓			✓			✓	1.2%	15
✓				✓		✓	1.1%	35
✓					✓	✓	1.2%	35

ARE is the average relative error of pressure (case 1)

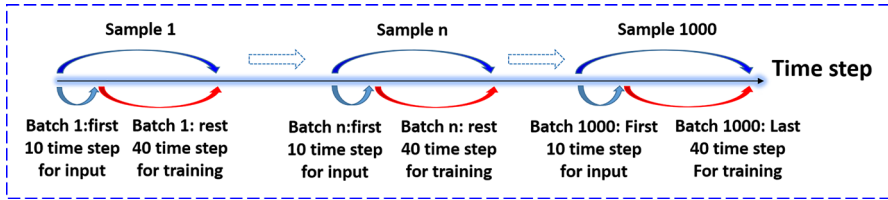


Fig. 4 Dataset organization

well control conditions in a three-dimensional heterogeneous gas reservoir measuring 1,120 feet  $\times$  1,350 feet  $\times$  670 feet. Details regarding data processing (preprocessing and post-processing), model training, and testing are discussed later in the section. Predictions and error measurements are provided for gas saturation, pressure, and CO<sub>2</sub> concentration in the gas phase.

The construction of the four-dimensional FNO neural network structure in this paper is based on PyTorch, a popular open-source machine learning software library. The three different numerical cases in this work were implemented on a DELL Precision T7920 workstation with two Inter(R) Xeon(R) Gold 6230R CPUs@ 2.10GHz, and two NVIDIA GeForce RTX 3090 Ti GPUs. All code and data required to reproduce the results presented in the paper will be available at <https://github.com/HPMPS> upon publication.

### 3.1 Case 1: Prediction of Time Series State Distribution in a Three-Dimensional Reservoir

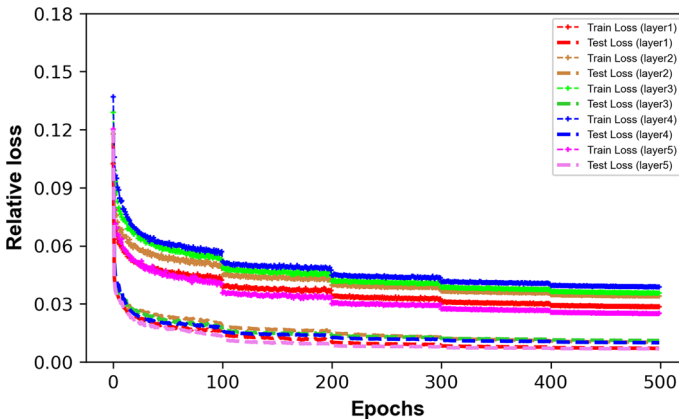
In this case study, we utilized the method proposed in Sect. 2 to predict subsurface flow state distribution, including the gas saturation, pressure, and CO<sub>2</sub> concentration in the gas phase, in the subsequent time steps under a three-dimensional heterogeneous reservoir. The Stanford University new-generation reservoir simulator ADGPRS (Younis 2011; Zhou 2012) was used to generate synthetic data for model training and testing.

In case 1, the reservoir domain is discretized using a Cartesian grid with dimensions of  $25 \times 25 \times 5$  grid cells in the  $x$ ,  $y$ , and  $z$  directions, respectively. Each grid cell measures 10 feet  $\times$  10 feet  $\times$  10 feet. The total number of grid cells is 3,125, and the reservoir contains two wells. The injection well is located at coordinates (25,25), and the production well is at (1,1), with each well perforating all five layers vertically. The CO<sub>2</sub> injection rate is set at 3,500 Mscf/day at the injection well, and the total liquid production rate is 2,000 STB/day at the production well. Permeability distributions ( $k_x$ ,  $k_y$ ) were generated using the SGeMS (Stanford Geostatistical Modeling Software) sequential Gaussian simulation for a grid size of  $25 \times 25$ . The values for  $k_x$  and  $k_y$  in each layer are displayed in Fig. 3. The fluid behavior is modeled using the Peng-Robinson equation of state (EOS) (Robinson and Peng 1978), suitable for an eight-component fluid, with fluid properties summarized in Table 8.

We take the pressure and saturation distribution of the first 10 time steps as input and the state distribution of last 40 time steps as output. The simulator runs for 10,500 days with 1,050 time steps, 10 days for each time step. After the numerical simulator

**Table 2** The time and spatial complexity analysis of the original FNO method and our proposed method on four-dimensional problems,  $N \propto m_x X \cdot m_y Y \cdot m_t T$

	Algorithm	Time complexity $T(n)$	Space complexity $S(n)$
Original FNO	$FFT_4(x, y, z, t)$	$O(Z \cdot m_z N \log(Z \cdot m_z N))$	$O(m_z Z \cdot N)$
Our proposed method	$Z \times FFT_3(x, y, t)$	$O(Z \cdot N \log N)$	$O(Z \cdot N)$



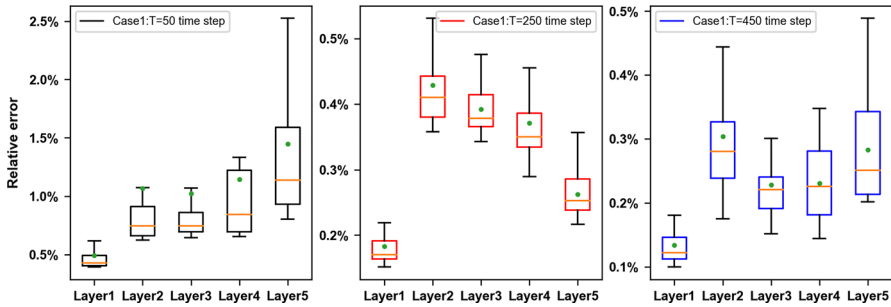
**Fig. 5** Training and testing accuracy versus the number of the epochs for case 1. Each epoch takes about 15 s in the NVIDIA GeForce RTX 3090 Ti GPU

runs, the four-dimensional subsurface flow state distribution results obtained by the simulation are decomposed into the three-dimensional state distribution of five layers in the  $z$ -direction according to Eq. 7. In each layer, the FNO model training, testing, and validation are described as follows. Dataset organization is such that each batch has 50 consecutive time steps: batch 1 is 1 to 50 time steps, batch 2 is 2 to 51 time steps, and so on. The last batch is 1,000 to 1,050 time steps. The total batch number is 1,000.

Figure 4 illustrates the preparation of batches. For each batch, the values of the first 10 time steps are used as input data to train the FNO-net and predict the state values as  $\hat{u}$  in Eq. 12. Then, the true values in the remaining 40 time steps are introduced into Eq. 12 as  $u$  to obtain  $L_2$  loss. By combining BC loss and IC loss, we optimize the total loss in Eq. 11 by the Adam optimization method. Among the 1,000 batches, 800 batches were randomly selected as the training data set and the remaining 200 batches as the testing data set. This method of assigning training sets and verification sets is widely used in machine learning research (Hu and Zhang 2022). To demonstrate the prediction capability in time series, the values in the three 50, 250, and 450 time steps are used as the validation data and are excluded in training and testing. The input shape for the network is (1,000, 25, 25, 1, 10), where 1,000 denotes the number of samples, 25 denotes the number of grids in the  $x$ -dimension, the other 25 is the number of the grids in the  $y$ -dimension, 1 is the required placeholder for variables, and 10 is the first 10 time steps in the time series. The output shape is (1,000, 25, 25, 40), where 40 is the remaining 40 time steps. The results show that our model accurately predicted the

**Table 3** Comparison between original FNO method and our proposed method on four-dimensional problems. Batch size is set as 1 for all the experiments

	No. of parameters	GPU memory consumption	Single-epoch training speed
Original FNO	19,798,679	45.6 GB	170.4 s
Our proposed method	1,247,323	9.1 GB	15 s



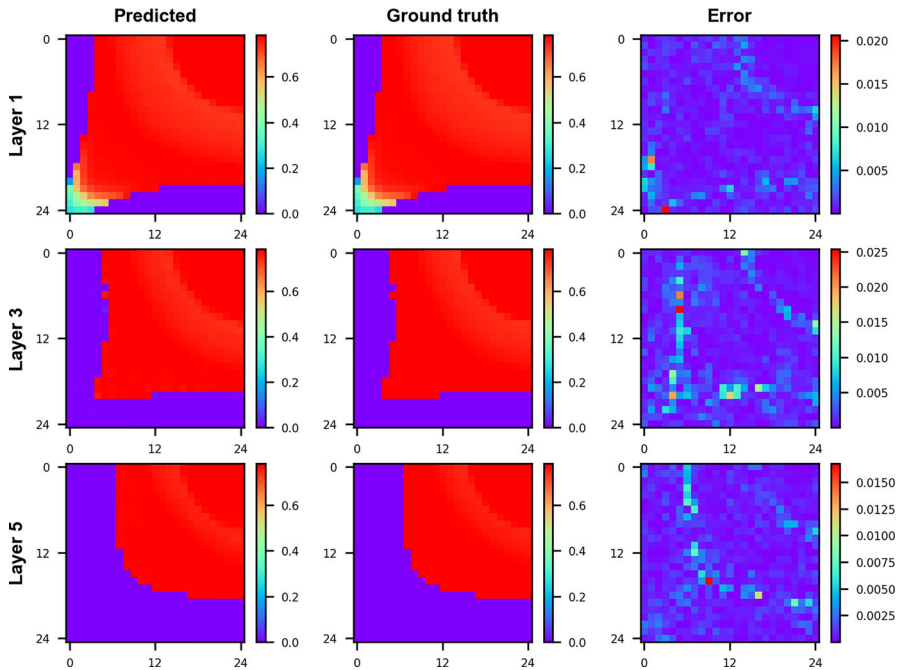
**Fig. 6** Relative errors of pressure field in different layers (layer 1 to layer 5) with 500 epochs training for case 1. The orange line in each box indicates the relative error of the median, and the two sides of the bottom and top indicate the 25th and 75th percentile error, respectively. The “whiskers” protruding from the box indicate minimum and maximum errors

subsurface flow state distributions in the future 40 time steps based on the current state distributions of 10 time steps. Because the three-dimensional FNO model training in a layer is independent of all other layers, we can train all three-dimensional FNO models in parallel to speed up the training process.

Training for 500 epochs takes about 7,500 s with the data from 1,050 time steps on an NVIDIA GeForce RTX 3090 Ti GPU computer. Each epoch takes 15 s in training. Then, it takes 3.2 s to predict the outputs for subsurface flow state distributions in the future 40 time steps. For case 1, we studied the effect of different numbers of Fourier layers on the model accuracy. The results are shown in Table 1. The results of the ablation study show that our network structure with five Fourier layers can achieve the best performance in training time and average relative error. Therefore, all test cases in the work adopt the same network structure, which includes five Fourier layers.

We now consider the relationships between the relative total loss and the number of training runs. Figure 5 shows the losses for the five layers based on training and validation data over 500 epochs. Generally, decreasing the learning rate by 0.1 every 100 epochs has a positive effect on the entire training process. Both training and validation losses were reduced similarly in all scenarios, indicating that the optimization did not overfit the model to the training data. The relative total loss from training can be reduced to about 0.03 with 100 epochs, and the losses decrease steadily and rapidly. As the number of epochs increases, the relative total loss from training can be further reduced to about 0.01 with 500 epochs.

Our model accurately predicted the gas saturation and pressure fields in the three-dimensional heterogeneous reservoir for the compositional modeling of the eight-component fluid. To investigate the relative error on every layer of our method, we

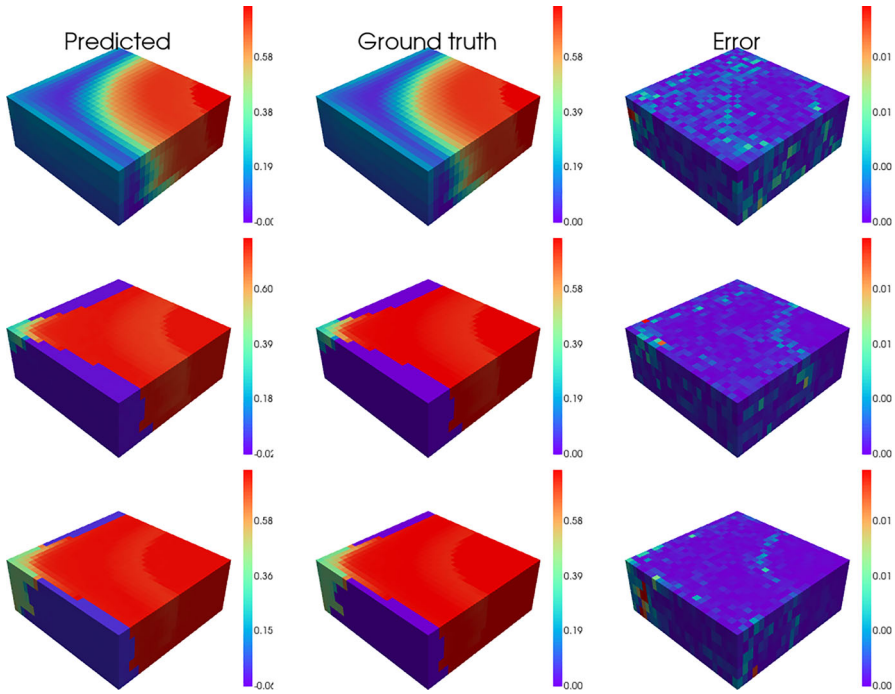


**Fig. 7** Prediction of gas saturation distribution in layer 1, layer 3, and layer 5 at the 250th time step (case 1): network results (first column), numerical simulation results (second column), errors (third column)

employed Eqs. 13–15 to evaluate the relative errors. Figure 6 displays the relative errors in pressure across five layers at three different testing time steps: 50, 250, and 450. The results, as shown in Fig. 6, indicate that the values of average relative error range from 0.2% to 1.2% for all test time steps.

Figure 7 depicts the gas saturation fields obtained by both the simulator and our network model using 1,000 samples at the 250th time step. The figure demonstrates that the model achieves higher approximation accuracy for the strongly nonlinear and discontinuous gas saturation field, as expected. It also highlights areas where the position of front discontinuity has higher approximation errors in all instances. Increasing the training sample size may allow the model to capture better characterization of the local nonlinear features and improve accuracy in the gas saturation at the front where  $\text{CO}_2$  is injected. Even with a small portion of training data used in this study, the maximum error of the gas saturation remains within the acceptable range.

In addition, for the gas saturation distribution, we compare the prediction results of different layers at the same time step. For example, in Fig. 7, layer 1 (the first row), layer 3 (the second row), and layer 5 (the third row) demonstrate that as  $\text{CO}_2$  injection progresses, the rate of change in the gas saturation field in the deeper layers is slower than in the shallower layers. This phenomenon may be attributed to the influence of the gravity field in the  $z$ -direction. After numerical simulation of the three-dimensional reservoir, the field distribution at each time step, considering vertical flow, is obtained. We decouple the simulation results of the entire four-dimensional reservoir along the

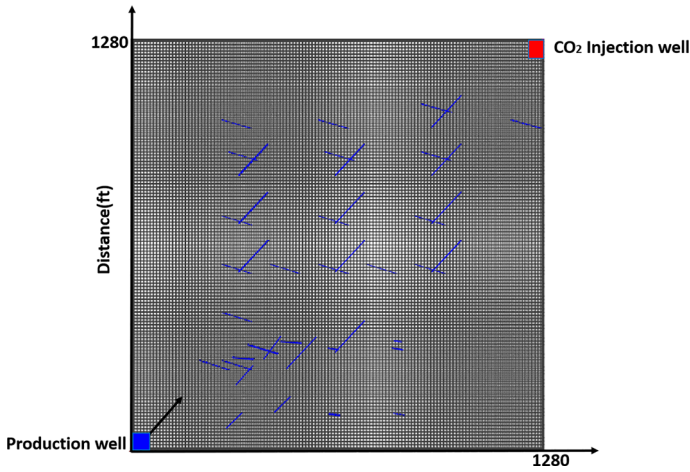


**Fig. 8** Visualization of four-dimensional gas saturation distribution: network results (first column), numerical simulation results (second column), errors (third column) for case 1 at the 50th (first row), 250th (second row), and 450th (third row) time steps, respectively

$z$ -dimension to obtain  $n_z$  layer fields at each time step. These data are then used to train the three-dimensional FNO network models in every  $z$ -layer. Thus, the network accounts for the influence of vertical flow during the training process. The predicted results further validate that our model effectively captures the impact of the gravity field on the four-dimensional CO<sub>2</sub> EOR simulation in the spatial  $z$ -axis.

As computations are independent in each layer, training and prediction in the four-dimensional spatiotemporal domain can be performed in parallel to expedite the process. We also developed four-dimensional visualization Python code to display the four-dimensional prediction results. Figure 8 shows the visualization of the four-dimensional gas saturation distribution for case 1 at the 50th (first row), 250th (second row), and 450th (third row) time steps, respectively. With this four-dimensional visualization software, we can quickly and clearly view the prediction results and overall errors of four-dimensional subsurface flow.

Here, we compared the performance of our proposed method with the original FNO method for time-dependent three-dimensional problems (four-dimensional problems) in terms of GPU memory consumption, number of parameters, and single-epoch training speed. Performing the fast Fourier transform (FFT) on a tensor with input data



**Fig. 9** Reservoir and wells in case 2. The blue square at the lower left corner of the configuration is the production well. The red square at the upper right corner of the configuration is the CO<sub>2</sub> injection well. The blue lines at the domain are the 42 natural fractures

**Table 4** The cost of training and prediction

Case	Time to train each epoch time (s)	Prediction 40 time steps (s)	Epoch number
Case 1	15	3.2	500
Case2	16.3	3.5	500

size  $N$  remains time-consuming, despite the optimized  $O(N \log N)$  complexity introduced by Cooley and Tukey (Cooley and Tukey 1965). The FFT layers are the most time-consuming steps in the FNO architecture, dominating the overall time complexity.

Table 2 presents the time and spatial complexity analysis for both methods. For input data with mesh size  $X \times Y \times Z \times T$ , where  $T$  represents the total time snapshots and  $m_x, m_y, m_z, m_t$  are the FFT modes in each dimension:

- (1) The time complexity for training an epoch using the original FNO method is  $O((m_x X \cdot m_y Y \cdot m_z Z \cdot m_t T) \log(m_x X \cdot m_y Y \cdot m_z Z \cdot m_t T))$ .
- (2) The time complexity for our proposed method is  $Z \times O((m_x X \cdot m_y Y \cdot m_t T) \log(m_x X \cdot m_y Y \cdot m_t T))$ .
- (3) The spatial complexity for the original FNO method is  $O(m_x X \cdot m_y Y \cdot m_z Z \cdot m_t T)$ .
- (4) The spatial complexity for our proposed method is  $Z \times O(m_x X \cdot m_y Y \cdot m_t T)$ .

The improvement in time complexity is quantified as

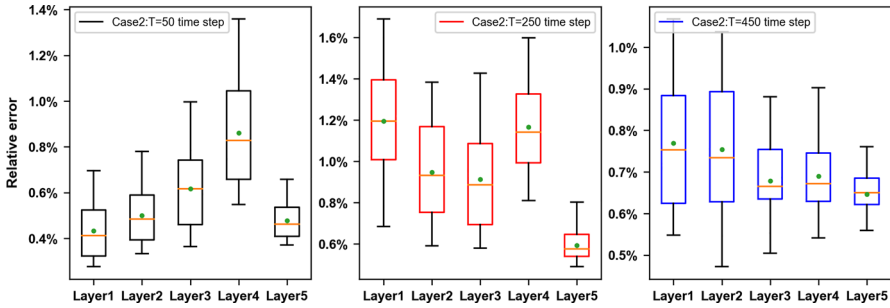
$$R_{T(n)} \approx m_z \left( 1 + \frac{\log(m_z Z)}{\log(N)} \right), \tag{16}$$

where  $N$  is proportional to  $m_x X \cdot m_y Y \cdot m_t T$ . This indicates a reduction factor greater than  $m_z$  as the number of layers  $Z$  increases.



**Table 5** Effect of the different numbers of frequency terms. ARE is the average relative error of pressure

Frequency term	ARE % (case 1)	ARE % (case 2)
600	8.60%	12.30%
800	4.50%	8.50%
1,000	1.20%	1.60%
1,200	1.10%	1.20%



**Fig. 10** Relative errors of pressure field in different layers (layer 1 to layer 5) with 500 epochs training for case 2. The orange line in each box indicates the relative error of the median, and the two sides of the bottom and top indicate the 25th and 75th percentile error, respectively. The “whiskers” protruding from the box indicate minimum and maximum errors

The improvement in space complexity is given by

$$R_{S(n)} \approx m_z, \tag{17}$$

indicating that the space complexity reduction is directly related to the number of FFT modes in the  $z$ -direction.

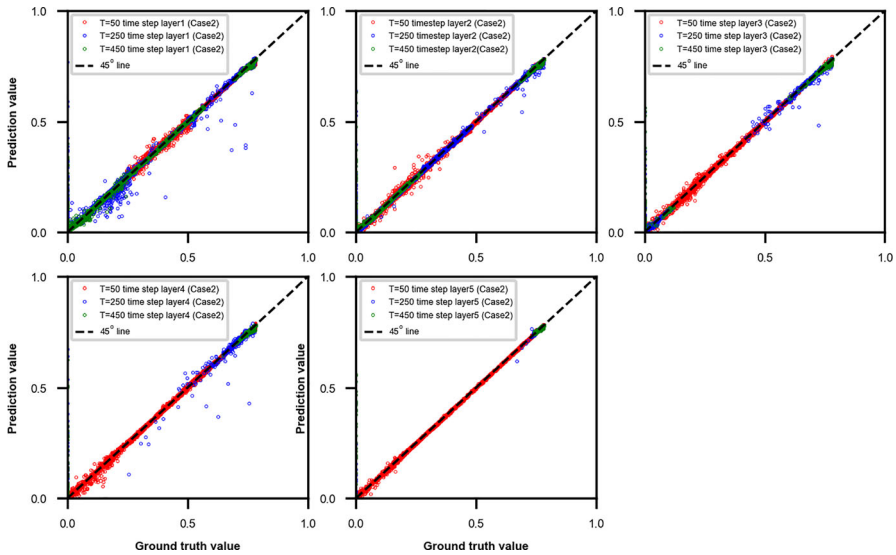
In this experiment, we chose the dataset for case 1 with mesh sizes of 25, 25, 5, and 10. Considering the high GPU memory consumption of the original FNO method, we chose an NVIDIA A800 with 80-GB GPU memory for this comparison. The detailed results are shown in Table 3. Compared with the original FNO, our proposed method only has 6.3% of the number of parameters and needs much less GPU memory (<20%).

### 3.2 Case 2: Prediction of Time Series State Distribution in a Three-Dimensional Fractured Reservoir

In the second test case, the reservoir was divided uniformly into a  $128 \times 128 \times 50$  Cartesian grid, with each grid measuring 10 feet  $\times$  10 feet  $\times$  10 feet. Additionally, 42 fractures were integrated into the reservoir. The total number of grids is 81,920 with two wells. The injection well is located in (128,128) and the production well in (1,1). Both wells penetrate the whole reservoir vertically. The injection well operates under gas rate control at 3,500 Mscf/day, and the production well under a total liquid rate control of 2,000 STB/day. The well configurations are depicted in Fig. 9. We employed the SGeMS sequential Gaussian simulation to generate permeability distributions ( $k_x, k_y$ )

**Table 6** Summary of each FNO network input. The shape of the field inputs is (112, 135). The  $N[a, b]$  indicates that the scalar input value ranges from  $a$  to  $b$ . The scalar variables are also broadcast into a matrix with dimensions (112, 135)

	Parameter	Notation	Distribution	Unit
Field inputs	Pressure field distribution at $t_n$	$P_{t_n}$	–	Bar
	Saturation field distribution at $t_n$	$S_{t_n}$	–	–
Scalar inputs	Specified well controls at $t_{n+1}$	$W_{t_{n+1}}$	$N[50362.2, 252411.0]$	$m^3/d$



**Fig. 11** Cross-plots of predicted pressure and ground truth values for five layers (layer 1 to layer 5, in order from left to right, top to bottom) for case 2. Red dots represent results at the 50th time step, blue dots at the 250th time step, and green dots at the 450th time step

with a grid size of  $128 \times 128$ . The fracture parameters including position, permeability in horizontal directions, permeability in vertical directions, porosity in fracture, and aperture of fracture are listed in Table 9. The distribution of these fractures is shown in Fig. 9. In this case, we adopted the eight-component fluid model shown in Table 8, which is the same as those used in case 1.

We ran ADGPRS to obtain synthetic data for training and testing, including pressure, gas saturation, and  $CO_2$  concentration in the gas phase. The simulation runs a total of 10,00 days with 1,050 time steps, 10 days for each time step. The selection of training data and the training process are the same as in case 1.

We evaluated the relative error in the pressure field by Eqs. 13–15. Figure 10 displays the relative error of pressure in five layers at the 50th, 250th, and 450th testing time steps. Note that the average values for relative error across all test time steps range from 0.5% to 1.4%. These results are from training over 500 epochs. For this case, training 500 epochs takes about 8,150s with the data from 1,050 time steps on an NVIDIA GeForce RTX 3090 Ti GPU computer. Each epoch takes 16.3 s. Then, it takes 3.5 s to

**Table 7** FNO-net structure for each layer of training. The padding operator represents the network's complement to a non-periodic boundary; linear transformation mainly to lift the input shape to the high-dimensional potential space, and the last linear operation is the projection output value back to the original space,  $\mathcal{F}_1$ - $\mathcal{F}_5$  is five Fourier layer

Network layer	Output shape (case 1)
Input	(1000, 25, 25, 40, 10)
Padding	(1000, 25, 25, 32, 10)
Linear	(1000, 25, 25, 32, 10)
$\mathcal{F}_1$ :Fourier3d/conv3d/BatchNorm3d	(1000, 25, 25, 32, 36)
$\mathcal{F}_2$ :Fourier3d/conv3d/BatchNorm3d	(1000, 25, 25, 32, 36)
$\mathcal{F}_3$ :Fourier3d/conv3d/BatchNorm3d	(1000, 25, 25, 32, 36)
$\mathcal{F}_4$ :Fourier3d/conv3d/BatchNorm3d	(1000, 25, 25, 32, 36)
$\mathcal{F}_5$ :Fourier3d/conv3d/BatchNorm3d	(1000, 25, 25, 32, 36)
Projection1	(1000, 25, 25, 32, 128)
Projection2	(1000, 25, 25, 32, 1)
Output	(1000, 25, 25, 40, 1)
Network layer	Output shape (case2)
Input	(1000, 128, 128, 40, 10)
Padding	(1000, 128, 128, 32, 10)
Linear	(1000, 128, 128, 32, 10)
$\mathcal{F}_1$ :Fourier3d/conv3d/BatchNorm3d	(1000, 128, 128, 32, 36)
$\mathcal{F}_2$ :Fourier3d/conv3d/BatchNorm3d	(1000, 128, 128, 32, 36)
$\mathcal{F}_3$ :Fourier3d/conv3d/BatchNorm3d	(1000, 128, 128, 32, 36)
$\mathcal{F}_4$ :Fourier3d/conv3d/BatchNorm3d	(1000, 128, 128, 32, 36)
$\mathcal{F}_5$ :Fourier3d/conv3d/BatchNorm3d	(1000, 128, 128, 32, 36)
Projection1	(1000, 128, 128, 32, 128)
Projection2	(1000, 128, 128, 32, 1)
Output	(1000, 128, 128, 40, 1)

predict the outputs for subsurface flow state distributions in the subsequent 40 time steps. In comparison, the traditional solver ADGPRS, which uses the GMRES linear solver with the two-stage CPR preconditioners, takes 18.01 s for every single time step, and about 720 s for all 40 time steps. Table 4 shows the cost of training and prediction for case 1 and case 2. In this study, we chose 600, 800, 1,000, and 1,200 samples to optimize the number of batches. Table 5 shows that the frequency term 1,000 can keep the average relative error within a reasonable range.

For the pressure distribution, to evaluate the predictive capability of our network, we present cross-plots of predicted and ground truth values for five layers (layer 1 through layer 5) in Fig. 11. Figure 11 shows that all testing points fall near the 45° line, demonstrating that our predicted solutions are in close agreement with the ground truth values.

**Table 8** Fluid properties in the case 1–2 study

Component	$P_c$ (psi)	$T_c$ (R)	Acentric factor	Molecular weight
$C O_2$	1069.87	547.56	0.22	44.01
$C_1 - N_2$	665.21	341.76	0.01	16.12
$C_2 - H_2S$	708.63	549.78	0.09	30.07
$C_3 - C_5$	543.67	762.06	0.19	57.27
$C_6 - C_7$	444.96	993.11	0.33	91.649
$C_8 - C_{10}$	355	1108.11	0.46	119.16
$C_{11} - C_{19}$	269.70	1155.99	0.62	192.23
$C_{20+}$	221.02	1428.18	1.04	414

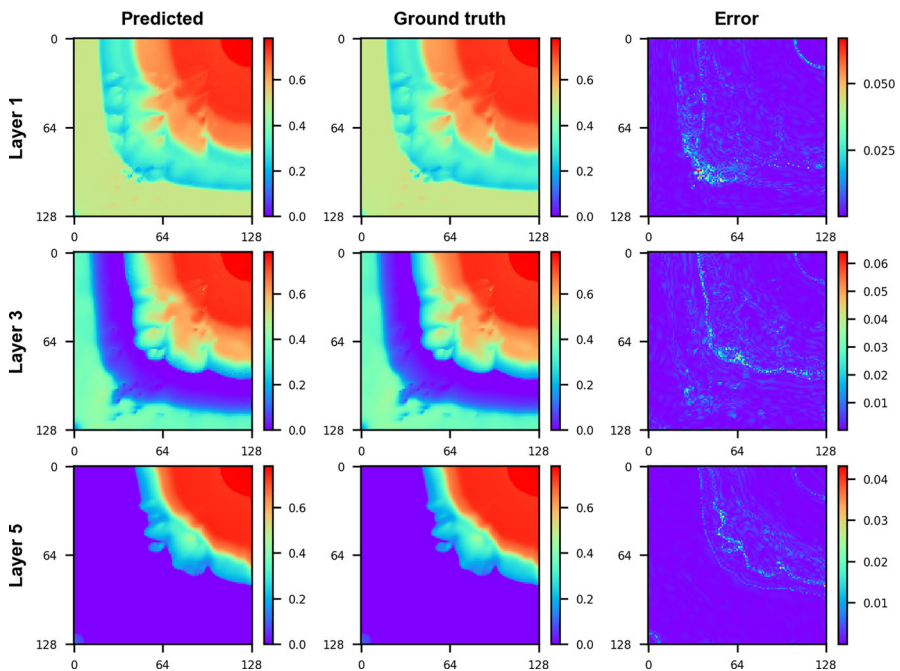
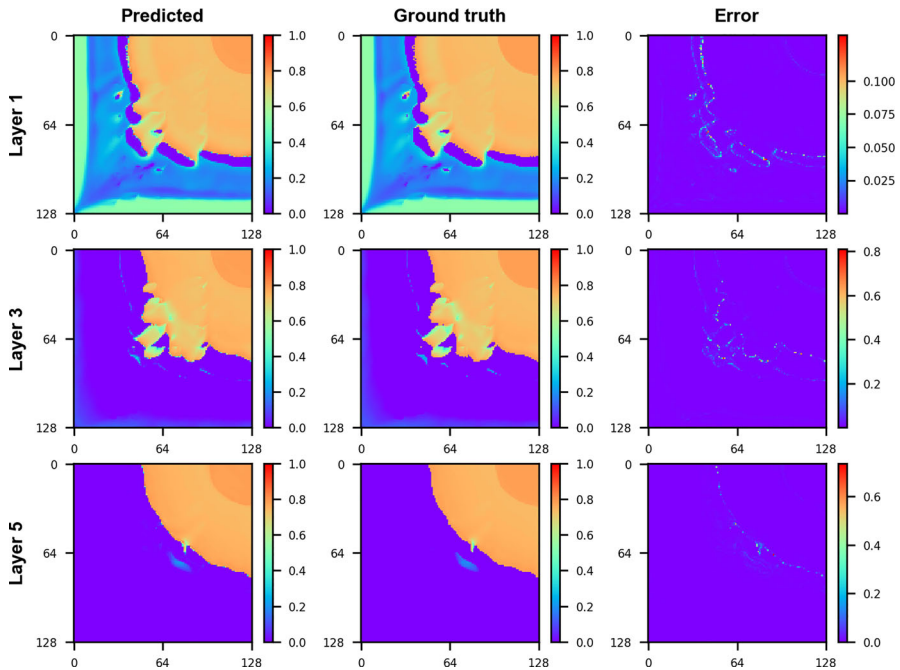
**Fig. 12** Prediction of gas saturation distribution in layer 1, layer 3, and layer 5 at the 250th time step (case 2): network results (first column), numerical simulation results (second column), errors (third column)

Figure 12 illustrates the comparison of the gas saturation distribution at the 250th time step between the simulator and our network model. Figure 12 shows that the distribution of gas saturation is very complex due to the presence of fractures, while the overall gas saturation error is within 0.06. The error on the boundary is within 0.01. Our model, as expected, achieves higher approximation accuracy for the strongly nonlinear and discontinuous gas saturation field in the fractured reservoir. In addition, the error of the model mainly exists in the injected gas front.



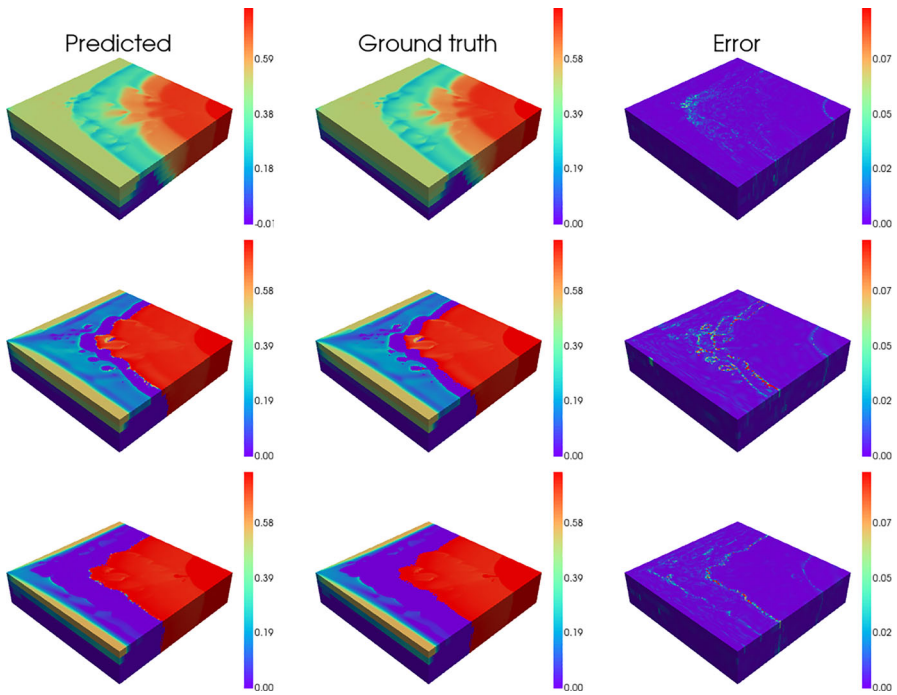
**Fig. 13** Prediction of CO<sub>2</sub> concentration in the gas phase in layer 1, layer 3, and layer 5 at the 450th time step (case 2): network results (first column), numerical simulation results (second column), errors (third column)

Figure 13 depicts a comparison of the CO<sub>2</sub> concentration in the gas phase at the 450th time step between the simulator and our network model. Similar to the prediction of gas saturation, the maximum predicted error is around the gas front (about 0.1 mole fraction). The results also show that the error of CO<sub>2</sub> concentration in the gas field is larger than that in the pressure field and gas saturation field. This discrepancy may be attributed to the compositional simulation, which presents a significant challenge to our model's predictive capability for CO<sub>2</sub> concentration in the gas phase. To address this issue, our subsequent work will involve incorporating physical laws related to thermodynamic phase equilibrium as constraints in our network loss function to enhance the prediction accuracy of CO<sub>2</sub> concentration in the gas phase.

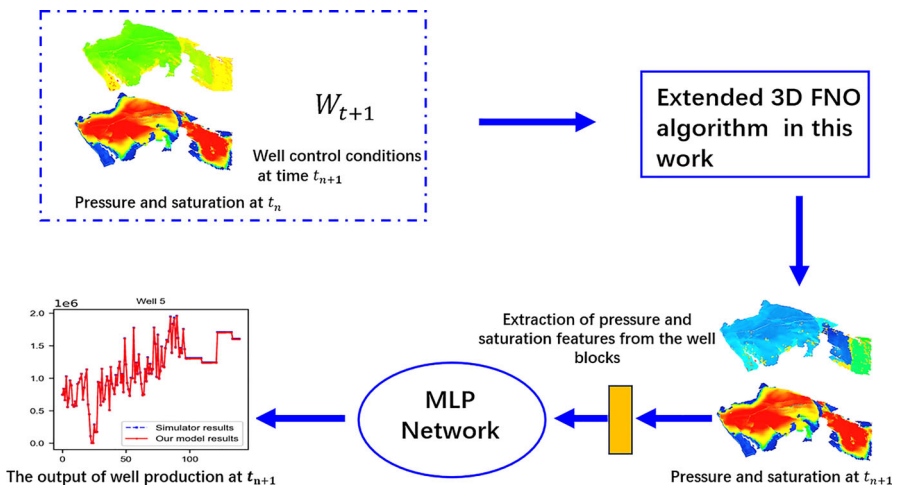
Figure 14 shows the visualization of the four-dimensional gas saturation distribution for case 2 at the 50th, 250th, and 450th time steps, respectively. The figure reveals good agreement between the predicted values and reference results.

### 3.3 Case 3: Well Production Prediction in the Three-Dimensional Reservoir with Varying Well Controls

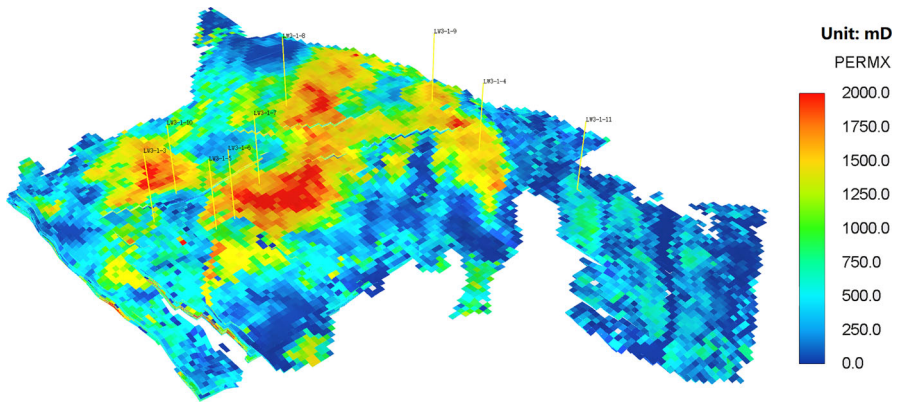
In reservoir engineering applications, it is usually necessary to allocate daily or weekly production or injection rates among hundreds of wells for a fixed geological model. Reservoir engineers employ optimization algorithms in conjunction with surrogate



**Fig. 14** Visualization of four-dimensional gas saturation distribution: network results (first column), numerical simulation results (second column), errors (third column) for case 2 at the 50th (first row), 250th (second row), and 450th (third row) time step, respectively



**Fig. 15** The overall architecture of the surrogate modeling process for the test case 3



**Fig. 16** The permeability field and the well configurations for test case 3

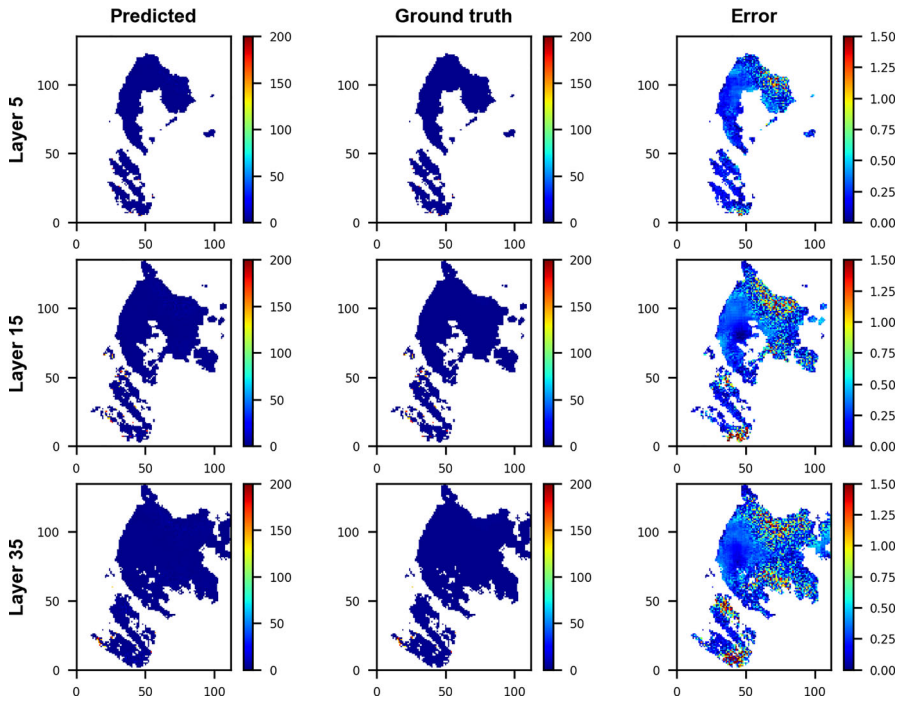
models to adjust well controls based on predicted outputs. Surrogate modeling must accurately predict production under varying well control conditions. To demonstrate the performance of our proposed algorithm in well rate production prediction under time-varying well conditions, we test a new application case. This surrogate model built by our proposed algorithm can make a dynamic prediction of the well rate production under time-varying well conditions.

The overall architecture of the surrogate modeling process for case 3 is shown in Fig. 15. The input is the pressure, saturation field at  $t_n$ , and the specified well control at  $t_{n+1}$ . The output is the pressure and saturation fields at  $t_{n+1}$ . Considering that case 3 is a gas reservoir that produced very little water in the early stages of development, gas saturation was only used for this work. Subsequently, the pressure and saturation values in the well grids at  $t_{n+1}$  were extracted, and the production was predicted by the multilayer perceptron (MLP) network.

Case 3 involves a three-dimensional gas reservoir with an oil-gas-water system. The permeability distributions and the well configurations are shown in Fig. 16. Nine production wells are located in the reservoir, and all producers are controlled by the gas rate target (GRAT). The reservoir is discretized using a Cartesian grid with dimensions of  $112 \times 135 \times 67$  grid cells in the  $x$ ,  $y$ , and  $z$  directions, totaling 1,013,040 grid cells.

For dataset preparation, we employed a Computer Modelling Group Ltd. (CMG) IMEX simulator to develop and simulate multiple cases as the training and testing datasets. We simulate production from January 1, 2011, to September 1, 2022, 1 month for each time step. All well controls are adjusted once every month. The control conditions (GRAT) are from 50,362.200 to 252,411.000  $m^3/day$  with normal distribution. A total of 550 well controls were generated and simulated, among which 500 cases were used for training and 50 for testing.

The network inputs for each FNO are shown in Table 6. The inputs can be divided into two types: space-dependent inputs (field inputs) and scalar inputs. Field inputs include state distribution of pressure and saturation at  $t_n$ . Considering that case 3 is an irregularly shaped reservoir, we use zero-padding to denote cells that are outside the real reservoir. Therefore, the data structures for field input variables are matrices with



**Fig. 17** Prediction of pressure distribution in layer 5, layer 15, and layer 35 at the 80th time step (case 3): network results (first column), numerical simulation results (second column), and errors (third column)

dimensions (112, 135). The scalar inputs include the well controls at  $t_{n+1}$ . The well controls are randomly sampled, ranging from 50,362.200 to 252,411.000  $m^3/d$ . The scalar variables are also broadcast into a matrix with dimensions (112, 135), and the values of other cells are filled with zeros. The final input data sample is constructed by connecting field variables, scalar variables, spatial grids, and temporal grids together. The final input data sample structure is (112, 135, 3).

Once the surrogate model is trained, it can predict the distribution of the pressure and saturation field and well production under various well control conditions. In the training process, we use the Adam optimizer to train for 500 epochs with an initial learning rate of 0.001 which is halved every 100 epochs. The neurons in the connection networks P and Q in Fig. 1 are activated using the sigmoid function.

After the surrogate model was trained, we predicted pressure and saturation distribution at the 80th month. Figures 17 and 18 show that the predicted pressure and saturation distribution present a good match with the reference results. The medium relative errors for well pressure and saturation are 0.85% and 0.35%, respectively. Additionally, we presented the gas production results for the nine wells in Fig. 19, in which the blue dotted lines represent the results from the simulator, and the red lines represent the results from our model. Figure 19 shows that our model achieves very high accuracy in predicting well production. Figure 20 compares the results of the three-dimensional ( $x, y, z$ ) gas saturation distribution for case 3 at time step 80.



**Table 9** Parameters of 42 fractures in the case 2 study

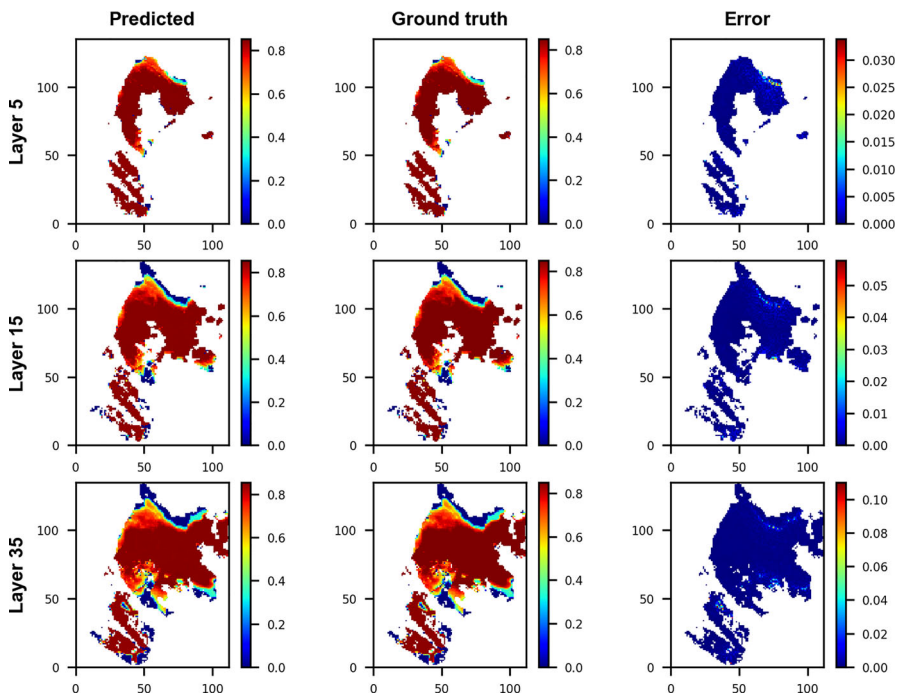
Natural fracture	PERM(mD)	PERMV(mD)	PORO	APERTURE
<i>Fracture1</i>	150	1.5	0.9	0.02
<i>Fracture2</i>	500	12.5	0.5	0.04
<i>Fracture3</i>	170	10.5	0.4	0.1
<i>Fracture4</i>	250	4.5	0.19	0.02
<i>Fracture5</i>	175	6.5	0.16	0.08
<i>Fracture6</i>	225	7.3	0.55	0.2
<i>Fracture7</i>	850	1.5	0.19	0.25
<i>Fracture8</i>	380	4.5	0.29	0.25
<i>Fracture9</i>	720	3.5	0.29	0.52
<i>Fracture10</i>	480	4.7	0.39	0.48
<i>Fracture11</i>	500	2.6	0.69	0.83
<i>Fracture12</i>	100	1.2	0.74	0.55
<i>Fracture13</i>	550	1.3	0.98	0.35
<i>Fracture14</i>	250	1.55	0.78	0.53
<i>Fracture15</i>	250	1.57	0.75	0.08
<i>Fracture16</i>	250	1.35	0.5	0.07
<i>Fracture17</i>	350	1.55	0.94	0.07
<i>Fracture18</i>	450	1.45	0.93	0.06
<i>Fracture19</i>	550	1.75	0.98	0.05
<i>Fracture20</i>	700	1.65	0.89	0.04
<i>Fracture21</i>	600	10.5	0.85	0.03
<i>Fracture22</i>	450	12.5	0.5	0.03
<i>Fracture23</i>	650	11.6	0.3	0.03
<i>Fracture24</i>	450	8.56	0.4	0.02
<i>Fracture25</i>	350	7.56	0.5	0.01
<i>Fracture26</i>	650	6.53	0.54	0.1
<i>Fracture27</i>	350	8.53	0.44	0.11
<i>Fracture28</i>	250	5.55	0.4	0.25
<i>Fracture29</i>	350	2.65	0.3	0.11

For computational efficiency, the 550 cases were simulated on an Intel Xeon(R) W-3275M dual CPU with 28 cores, and each run took approximately 1,500 s. On an NVIDIA GeForce RTX 3090 Ti GPU node with 24 GB video memory, it took 25,510 s to train our surrogate model. After training, the surrogate model took 125 s to forecast the outputs for all 50 test cases, with an average time cost of 2.5 s for each one. A speedup factor of over 600x is observed for this application scenario after model training.

In this case, the constructed surrogate model can provide flow predictions under time-varying well controls. By inputting the pressure and saturation field of the current time step and future well controls, the well production rates, pressure, and saturation

**Table 9** continued

Natural fracture	PERM(mD)	PERMV(mD)	PORO	APERTURE
<i>Fracture30</i>	454	7.52	0.2	0.32
<i>Fracture31</i>	455	1.45	0.93	0.06
<i>Fracture32</i>	652	1.75	0.98	0.05
<i>Fracture33</i>	720	1.65	0.89	0.04
<i>Fracture34</i>	650	10.5	0.85	0.03
<i>Fracture35</i>	400	12.5	0.5	0.03
<i>Fracture36</i>	620	11.6	0.3	0.03
<i>Fracture37</i>	455	8.56	0.4	0.02
<i>Fracture38</i>	355	7.56	0.5	0.01
<i>Fracture39</i>	650	6.53	0.54	0.1
<i>Fracture40</i>	250	8.53	0.44	0.11
<i>Fracture41</i>	650	5.55	0.4	0.25
<i>Fracture42</i>	450	2.65	0.3	0.11



**Fig. 18** Prediction of saturation distribution in layer 5, layer 15, and layer 35 at the 80th time step (case 3): network results (first column), numerical simulation results (second column), and errors (third column)

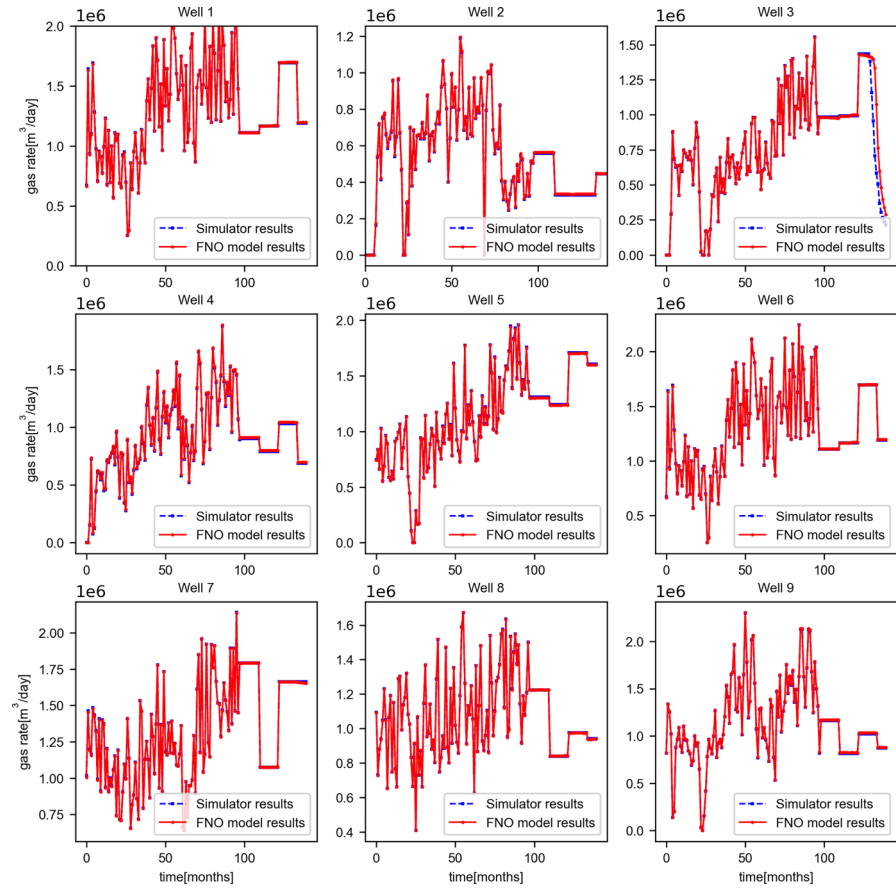


Fig. 19 Gas rate prediction in case 3

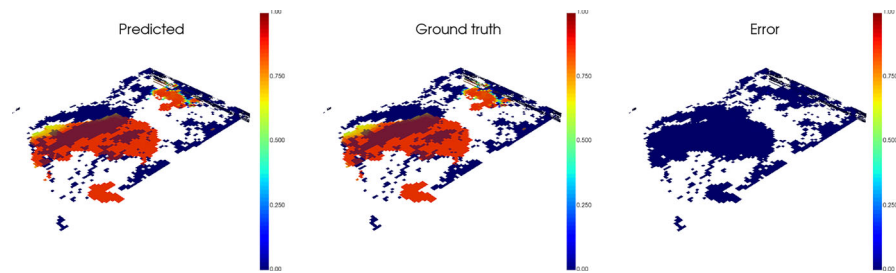


Fig. 20 Visualization of three-dimensional gas saturation distribution for case 3 at the 80th time step

distributions at the next time step can be predicted. The trained model has been tested under different well control scenarios, and the results showed that our proposed model can accurately predict the pressure and saturation fields. In addition, the production forecast for each well also matched the corresponding simulation results.

## 4 Discussion

We have extended the FNO from three-dimensional  $(x, y, t)$  to four-dimensional  $(x, y, z, t)$  based on the domain decomposition method. Although the general four-dimensional  $(x, y, z, t)$  FNO network, which is based on the three-dimensional FFT, is computationally very expensive and hard to employ in an application, we trained and used the four-dimensional  $(x, y, z, t)$  FNO network from the two-dimensional FFT. Therefore, our developed framework retains the speed advantage of FNO in three dimensions  $(x, y, t)$  and can be applied to large-scale four-dimensional  $(x, y, z, t)$  subsurface flow simulation. To demonstrate the developed four-dimensional FNO applicability in reservoir engineering, we use our method to build surrogate models in the dynamic prediction of a large three-dimensional reservoir (over 1 million grids) under changing well conditions in case 3. Excellent prediction is observed in well production and pressure and saturation fields.

Although we have been able to simulate the four-dimensional  $(x, y, z, t)$  subsurface flow problems using the FNO network and the domain decomposition method, this was only done on orthogonal grids at this time. For the unstructured grid, more reservoir characteristics are not considered, such as production splitting, reservoir inclination, and so on. Recent studies have shown that complex geometric shapes can be fed into a normal FNO network through a spatial transformation (Li et al. 2022), and this work can be extended to deal with large-scale irregularly shaped geometry. We can combine our current work with irregularly shaped geometry techniques to build surrogate models for large-scale three-dimensional complex reservoirs and use these models in history matching, dynamic prediction of the reservoir, optimization of well conditions, uncertainty quantification, and other scenarios. For example, we can combine the proxy model established by our proposed method with particle swarm optimization (PSO) or simultaneous perturbation stochastic approximation (SPSA) algorithms to optimize the total oil production of the oilfield in the future production process by adjusting the water injection rate of all injectors. This could be extremely helpful when large amounts of simulation are required for production optimization on realistic reservoirs.

## 5 Conclusions

The FNO is a fast and promising neural network method to speed up the simulation of engineering applications. Because of the complexity of this method, up to now, it has only worked for three-dimensional  $(x, y, t)$  simulation problems and has had little use in practice, because practical simulations are always in the four-dimensional  $(x, y, z, t)$  domain. In this work, we employ three-dimensional FNO along the  $x$ -axis

and  $y$ -axis and incorporate the time component, thus circumventing the substantial computational expense associated with directly predicting four-dimensional flow state variables using existing four-dimensional FNO. This new methodology successfully extends the current FNO network from three-dimensional spatiotemporal domains to four-dimensional spatiotemporal domains for the subsurface flow simulations. The performance comparison results in case 1 indicate that, compared to the original FNO, our proposed method has only 6.3% of the parameters and requires less GPU memory (<20%). In addition, the single-epoch training speed is improved eightfold. The time complexity improvement of our proposed method is not dominated by the number of layers in the  $z$ -direction, and the space complexity is directly related to the number of FFT modes in the  $z$ -direction.

The new framework was successfully tested for some complex cases of CO<sub>2</sub> injection for enhanced oil recovery by compositional simulations. The first case is a three-dimensional (250 feet  $\times$  250 feet  $\times$  50 feet) heterogeneous reservoir with an eight-component fluid. The second case is a three-dimensional (1,280 feet  $\times$  1,280 feet  $\times$  50 feet) heterogeneous reservoir with 42 fractures and an eight-component fluid. The framework is also applied to the surrogate model to accurately predict the pressure and saturation distribution of the reservoir and the well productions in an over-1-million-grid reservoir with nine wells (case 3). Key achievements in this work include the following:

- (1) Our new framework successfully extends the current FNO-network from three dimensions  $(x, y, t)$  to four dimensions  $(x, y, z, t)$  to predict the state distributions in subsurface flow problems.
- (2) The computational speed in four dimensions  $(x, y, z, t)$  by our method can be as fast as in three dimensions  $(x, y, t)$ , which overcomes the very low speed of the original four-dimensional  $(x, y, z, t)$  FNO method.
- (3) The tested results show that our new framework can efficiently simulate the complex EOR processes by injecting CO<sub>2</sub> into complex fracture reservoirs in compositional simulation, and predict the well productions and the field pressure and saturation distribution in a large three-dimensional reservoir (over 1 million grids) with time-varying well conditions.

Future research will mainly involve the following aspects. In terms of network framework innovation, we will combine the current work with it and introduce the irregularly shaped geometry techniques to further process the large-scale three-dimensional complex reservoirs. In terms of application, we will establish the surrogate model that can describe the layered injection production relationship based on current work, and combine optimization algorithms to achieve intelligent parameter optimization, solving the challenges brought by time-consuming numerical simulators in traditional optimization processes.

## Appendix: Detailed Loss Functions for Boundary and Initial Conditions

In our study, the boundary condition for all cases is the Newman boundary.

$$K(\mathbf{x})\nabla p(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) = 0, \mathbf{x} \in \Gamma_N \quad (\text{A.1})$$

Equation A.1 denotes the Newman boundary condition.  $K(\mathbf{x})$  is the permeability;  $P(\mathbf{x})$  is the pressure;  $\mathbf{n}(\mathbf{x})$  is the normal vector, which is normal to the boundary. The model parameters  $K(\mathbf{x})$  are known, and the boundary conditions and initial conditions are specified by

$$L_{BC}(\theta_P) = \frac{1}{NN_t} \sum_{i=1}^N \sum_{t=1}^{N_t} \|R_l(\theta_P)\|_2^2. \quad (\text{A.2})$$

Equation A.2 represents the loss function of the boundary condition.  $N$  is the number of grids.  $R_l(\theta_P)$  represents the residual of Eq. A.1.

$$R_{P,IC}(\theta_P) = \hat{P}(\theta_P) - P(\theta_P) \quad (\text{A.3})$$

$$R_{S,IC}(\theta_S) = \hat{S}(\theta_S) - S(\theta_S) \quad (\text{A.4})$$

Equations A.3 and A.4 are the difference between the predicted value and the initial value, respectively.  $R_{P,IC}$  is the residual of pressure.  $R_{S,IC}$  is the residual of saturation.

$$L_{IC} = \frac{1}{N} \sum_{i=1}^N \|R(\theta_P)\|_2^2 + \frac{1}{N} \sum_{i=1}^N \|R(\theta_S)\|_2^2 \quad (\text{A.5})$$

Equation A.5 represents the loss function of the initial condition, and  $R(\theta_P)$  and  $R(\theta_S)$  represent the residual of pressure and saturation, respectively.

**Acknowledgements** We gratefully acknowledge the SUPRI-B industrial affiliates program at Stanford University for the permission to use ADGPRS. This research is financially supported by special funding to Huanquan Pan from the Ministry of Science and Technology of China and the China University of Geosciences (Wuhan). The support is greatly appreciated.

**Data availability** Code and Data: The code and data required to reproduce the results presented in the paper will be available at <https://github.com/HPMPS> upon publication.

## References

- Ajayi T, Gupta I (2019) A review of reactive transport modeling in wellbore integrity problems. *J Pet Sci Eng* 175:785–803
- Anandkumar A, Azizzadenesheli K, Bhattacharya K, Kovachki N, Li Z, Liu B, Stuart A (2020) Neural operator: graph kernel network for partial differential equations. In: ICLR 2020 workshop on integration of deep neural models and differential equations. <https://openreview.net/forum?id=fg2ZFmXFO3>
- Bagheri M, Rezaei H (2019) Reservoir rock permeability prediction using SVR based on radial basis function kernel. *Carbonat Evaporit* 34:699–707

- Bagheri M, Riahi MA (2015) Seismic facies analysis from well logs based on supervised classification scheme with different machine learning techniques. *Arab J Geosci* 8:7153–7161
- Benson SM, Cole DR (2008) CO<sub>2</sub> sequestration in deep sedimentary formations. *Elements* 4(5):325–331
- Benson SM, Li B, Krause M, Krevor S, Kuo CW, Pini R, Zuo L (2012) Investigations in geologic carbon sequestration: multiphase flow of CO<sub>2</sub> and water in reservoir rocks annual report 2012. Technical report
- Bhattacharya K, Hosseini B, Kovachki NB, Stuart AM (2021) Model reduction and neural networks for parametric pdes. *SMAI J Comput Math* 7:121–157
- Cooley JW, Tukey JW (1965) An algorithm for the machine calculation of complex Fourier series. *Math Comput* 19(90):297–301
- Fu W, Zhang K, Ma X, Liu P, Zhang L, Yan X, Yang Y, Sun H, Yao J (2023) Deep conditional generative adversarial network combined with data-space inversion for estimation of high-dimensional uncertain geological parameters. *Water Resour Res* 59(3):e2022WR032553
- Fuks O, Tchelepi HA (2020) Limitations of physics informed machine learning for nonlinear two-phase transport in porous media. *J Mach Learn Model Comput* 1(1):21–28
- Haghighat E, Juanes R (2021) Sciann: a Keras/tensorflow wrapper for scientific computations and physics-informed deep learning using artificial neural networks. *Comput Methods Appl Mech Eng* 373. <https://doi.org/10.2139/ssrn.4166142>
- Hu JW, Zhang WW (2022) Mesh-conv: convolution operator with mesh resolution independence for flow field modeling. *J Comput Phys* 452:110896
- Huang H, Gong B, Liu Y, Sun W (2023) Deep-learning based surrogate modeling for fast and accurate simulation in realistic 3D reservoir with varying well controls. *Geoenergy Sci Eng* 211418
- Jiang S, Durlafsky LJ (2023) Use of multifidelity training data and transfer learning for efficient construction of subsurface flow surrogate models. *J Comput Phys* 474:111800
- Jiang Z, Tahmasebi P, Mao Z (2021) Deep residual u-net convolution neural networks with autoregressive strategy for fluid flow predictions in large-scale geosystems. *Adv Water Resour* 150:103878
- Jiang Z, Zhu M, Li D, Li Q, Yuan YO, Lu L (2023) Fourier-mionet: Fourier-enhanced multiple-input neural operators for multiphase modeling of geological carbon sequestration. *arXiv preprint arXiv:2303.04778*
- Kim YD, Durlafsky LJ (2021) A recurrent neural network-based proxy model for well-control optimization with nonlinear output constraints. *SPE J* 26(04):1837–1857
- Kuang T, Liu J, Yin Z, Jing H, Lan Y, Lan Z, Pan H (2023) Fast and robust prediction of multiphase flow in complex fractured reservoir using a Fourier neural operator. *Energies* 16(9):3765
- Li J, Zhang D, He T, Zheng Q (2023) Uncertainty quantification of two-phase flow in porous media via the coupled-TGNN surrogate model. *Geoenergy Sci Eng* 221:211368
- Li L, Jing H, Liu J, Pan H, Fang Z, Kuang T, Lan Y, Guo J (2024) The artificial neural network-based two-phase equilibrium calculation framework for fast compositional reservoir simulation of CO<sub>2</sub>. *Fluid Phase Equilibria* 2024:114151
- Li L, Lee SH (2008) Efficient field-scale simulation of black oil in a naturally fractured reservoir through discrete fracture networks and homogenized media. *SPE Reservoir Eval Eng* 11(04):750–758
- Li Z, Huang DZ, Liu B, Anandkumar A (2022) Fourier neural operator with learned deformations for PDES on general geometries. *arXiv preprint arXiv:2207.05209*
- Li Z, Kovachki N, Azizzadenesheli K, Liu B, Bhattacharya K, Stuart A, Anandkumar A (2020a) Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*
- Li Z, Kovachki N, Azizzadenesheli K, Liu B, Bhattacharya K, Stuart A, Anandkumar A (2020b) Neural operator: graph kernel network for partial differential equations. *arXiv preprint arXiv:2003.03485*
- Li Z, Kovachki N, Azizzadenesheli K, Liu B, Stuart A, Bhattacharya K, Anandkumar A (2020) Multipole graph neural operator for parametric partial differential equations. *Adv Neural Inf Process Syst* 33:6755–6766
- Li Z, Zheng H, Kovachki N, Jin D, Chen H, Liu B, Azizzadenesheli K, Anandkumar A (2021) Physics-informed neural operator for learning partial differential equations. *arXiv preprint arXiv:2111.03794*
- Liu P, Zhang K, Yao J (2023) Reservoir automatic history matching: methods, challenges, and future directions. *Adv Geo-Energy Res* 7(2):136–140
- Liu W, Liu W D, Gu J (2019) Petroleum production forecasting based on machine learning. In *Proceedings of the 2019 3rd international conference on advances in image processing. CAIP 2019*, November 8–10, 2019, Chengdu, China, pp 124–128. <https://doi.org/10.1145/3373419.3373421>

- Lu L, Jin P, Karniadakis GE (2019) Deeponet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators. arXiv preprint [arXiv:1910.03193](https://arxiv.org/abs/1910.03193)
- Mo S, Zhu Y, Zabarar N, Shi X, Wu J (2019) Deep convolutional encoder-decoder networks for uncertainty quantification of dynamic multiphase flow in heterogeneous media. *Water Resour Res* 55(1):703–728
- Moinfar A, Varavei A, Sepehrmoori K, Johns RT (2014) Development of an efficient embedded discrete fracture model for 3D compositional reservoir simulation in fractured reservoirs. *SPE J* 19(02):289–303
- Moosavi N, Bagheri M, Nabi-Bidhendi M, Heidari R (2022) Fuzzy support vector regression for permeability estimation of petroleum reservoir using well logs. *Acta Geophysica* 70(1):161–172
- Nasir Y, Durlafsky LJ (2023) Deep reinforcement learning for optimizing well settings in subsurface systems with uncertain geology. *J Comput Phys* 477:111945
- Pang G, Lu L, Karniadakis GE (2019) FPINNS: fractional physics-informed neural networks. *SIAM J Sci Comput* 41(4):A2603–A2626
- Pecha M, Horák D (2018) Analyzing  $L_1$ -loss and  $L_2$ -loss support vector machines implemented in permon toolbox. In: International conference on advanced engineering theory and applications. Springer, pp 13–23
- Rahmanifard H, Plaksina T (2019) Application of artificial intelligence techniques in the petroleum industry: a review. *Artif Intell Rev* 52(4):2295–2318
- Raissi M, Perdikaris P, Karniadakis GE (2019) Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J Comput Phys* 378:686–707
- Robinson DB, Peng DY (1978) The characterization of the heptanes and heavier fractions for the GPA Peng-Robinson programs. Gas Processors Association
- Shi Z, Yao W, Li Z, Zeng L, Zhao Y, Zhang R, Tang Y, Wen J (2020) Artificial intelligence techniques for stability analysis and control in smart grids: Methodologies, applications, challenges and future directions. *Appl Energy* 278:115733
- Shokouhi P, Kumar V, Prathipati S, Kifer D (2020) A physics-informed deep learning method for prediction of CO<sub>2</sub> storage site response. In: AGU fall meeting abstracts, vol 2020, GC110-03
- Tang M, Liu Y, Durlafsky LJ (2020) A deep-learning-based surrogate model for data assimilation in dynamic subsurface flow problems. *J Comput Phys* 413:109456
- Tang M, Liu Y, Durlafsky LJ (2021) Deep-learning-based surrogate flow modeling and geological parameterization for data assimilation in 3D subsurface flow. *Comput Methods Appl Mech Eng* 376:113636
- Ṭene M, Bosma SB, Al Kobaisi MS, Hajibeygi H (2017) Projection-based embedded discrete fracture model (PEDFM). *Adv Water Resour* 105:205–216
- Wang N, Chang H, Zhang D (2021) Deep-learning-based inverse modeling approaches: a subsurface flow example. *J Geophys Res Solid Earth* 126(2):e2020JB020549
- Wen G, Benson SM (2021) CO<sub>2</sub> plume migration and dissolution in layered reservoirs. *Int J Greenhouse Gas Control* 87:66–79
- Wen G, Hay C, Benson SM (2021) Ccsnet: a deep learning modeling suite for CO<sub>2</sub> storage. *Adv Water Resour* 155:104009
- Wen G, Li Z, Azizzadenesheli K, Anandkumar A, Benson SM (2022) U-FNO-An enhanced Fourier neural operator-based deep-learning model for multiphase flow. *Adv Water Resour* 163:104180
- Wen G, Li Z, Long Q, Azizzadenesheli K, Anandkumar A, Benson SM (2023) Real-time high-resolution co<sub>2</sub> geological storage prediction using nested Fourier neural operators. *Energy Environ Sci* 16(4):1732–1741
- Wu C, Zhu M, Tan Q, Kartha Y, Lu L (2023) A comprehensive study of non-adaptive and residual-based adaptive sampling for physics-informed neural networks. *Comput Methods Appl Mech Eng* 403:115671
- Wu H, Qiao R (2021) Physics-constrained deep learning for data assimilation of subsurface transport. *Energy AI* 3:100044
- Yan B, Chen B, Harp DR, Jia W, Pawar RJ (2022) A robust deep learning workflow to predict multiphase flow behavior during geological co<sub>2</sub> sequestration injection and post-injection periods. *J Hydrol* 607:127542
- Yan B, Harp DR, Chen B, Pawar RJ (2022) Improving deep learning performance for predicting large-scale geological co<sub>2</sub> sequestration modeling through feature coarsening. *Sci Rep* 12(1):20667
- Younis RM (2011) Modern advances in software and solution algorithms for reservoir simulation. Ph.D. thesis. Stanford University



- Yu J, Lu L, Meng X, Karniadakis GE (2022) Gradient-enhanced physics-informed neural networks for forward and inverse PDE problems. *Comput Methods Appl Mech Eng* 393:114823
- Zare A, Bagheri M, Ebadi M (2020) Reservoir facies and porosity modeling using seismic data and well logs by geostatistical simulation in an oil field. *Carbonat Evaporit* 35:1–10
- Zeng QD, Yao J, Shao J (2019) Study of hydraulic fracturing in an anisotropic poroelastic medium via a hybrid EDFM-XFEM approach. *Comput Geotech* 105:51–68
- Zhang K, Fu W, Zhang J, Zhou W, Liu C, Liu P, Zhang L, Yan X, Yang Y, Sun H et al (2023) Inversion framework of reservoir parameters based on deep autoregressive surrogate and continual learning strategy. *SPE J* 28:2223–2239
- Zhang K, Zuo Y, Zhao H, Ma X, Gu J, Wang J, Yang Y, Yao C, Yao J (2022) Fourier neural operator for solving subsurface oil/water two-phase flow partial differential equation. *SPE J* 27(03):1–15
- Zhong Z, Sun AY, Jeong H (2019) Predicting CO<sub>2</sub> plume migration in heterogeneous formations using conditional deep convolutional generative adversarial network. *Water Resour Res* 55(7):5830–5851
- Zhou Y (2012) Parallel general-purpose reservoir simulation with coupled reservoir models and multisegment wells. Ph.D. thesis, Stanford University
- Zhu Y, Zabarar N (2018) Bayesian deep convolutional encoder-decoder networks for surrogate modeling and uncertainty quantification. *J Comput Phys* 366:415–447
- Zhu Y, Zabarar N, Koutsourelakis PS, Perdikaris P (2019) Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data. *J Comput Phys* 394:56–81

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.