

Predicting CO₂ storage pressure and saturation based on deep learning surrogate model

Jianchun Xu^{1,2*}, Qirun Fu^{1,2}

1 Key Laboratory of Unconventional Oil & Gas Development (China University of Petroleum (East China)), Ministry of Education, Qingdao 266580, P. R. China

2 School of Petroleum Engineering, China University of Petroleum (East China), Qingdao 266580, P. R. China

(*Corresponding Author: 20170048@upc.edu.cn)

ABSTRACT

CO₂ sequestration which is one of the important means to avoid global warming today provides an economically viable technical means to reduce greenhouse gas emissions. To maximize the sequestration efficiency while evaluating the effect of formation uncertainty, a numerical simulator of multiphase flow is required to simulate high-dimensional nonlinear multiphase flow within a non-homogeneous porous medium. Due to the inherent inhomogeneity of the formation of porous media and the nonlinear coupling of multiple complex physical processes, a significant amount of repetitive numerical simulation processes impose considerable computational costs and require prolonged computational time to obtain simulation results.

In this paper, we propose an efficient and fast flow surrogate modeling process for deep learning, proposing that the extended hyperparameter optimization process will incorporate the neural network architecture and the loss function as relevant parameters into the optimization process. Subsequently, we conducted experiments based on the workflow proposed in this study for the case of CO₂ storage in a homogeneous deep saltwater layer and achieved accurate predictions at 120-time steps with mean MSE of 5E-5 and 2E-5 for gas saturation and pressure, and MSSIMs of 0.9989 and 0.9998, respectively, under different production parameters and well placement settings.

Keywords: Numerical simulation, Surrogate model, Deep-learning, Geological sequestration

NOMENCLATURE

Abbreviations

MSE	mean square error
SSIM	Structural Similarity Index
MSSIM	mean Structural Similarity Index
CCUS	Carbon dioxide capture, utilization and storage
CNN	Convolutional Neural Networks
2D	Two-dimensional
TPE	Tree-structured Parzen Estimator algorithm
RMSE	root mean square error
CMG	Computer Modeling Group
BHP	bottom hole pressure
ADAM	adaptive moment estimation method

Symbols

C	mass fraction of components
ρ	density
v	flow Rate
S	fluid phase saturation
μ	viscosity
φ	rock porosity
q	mass flow rate
P	pressure
Z	depth of reservoir
K	absolute permeability
λ	fluidity
g	gravitational constant

y	output features of the simulator
x	input features of the simulator
\hat{y}	the output of the surrogate model
\hat{x}	the input of the surrogate model
θ	internal parameters of neural architecture
<i>Well</i>	well placement settings
t	feature map of time step
F	output feature map
W	weights
b	bias
p	pixel
N	total number of pixels
μ	average value
σ	variance/covariance
c	small constants
M	number of local images
u	MSSIM weighting factor
<i>Superscripts</i>	
m	mth time step
\circ	convolution operation
<i>Subscript</i>	
g	gas phase
w	aqueous phase
i	component
l	fluid phase
bh	bottom hole
inj	injection well
k	kth layer
p	pixel within the image signal
L	local image
y	feature Map

1. INTRODUCTION

Carbon sequestration, as one of the effective means to reduce carbon emissions, requires the guidance of a large number of numerical simulation processes. Due to the multi-scale inhomogeneity of porous media and complex coupled physical processes, the control equations describing multiphase fluid flow in porous media are highly nonlinear, which brings huge computational costs and running time.

Compared to complex numerical simulation models with high computational costs, surrogate models can significantly reduce computing time without sacrificing accuracy or detail[1,2]. Compared with surrogate models based on low-dimensional features, image-based surrogate models retain more original variables and are widely used for the prediction of fluid flow in

porous media[3–6]. However, redesigning the neural architecture and loss function to build the surrogate model is quite time-consuming.

We propose a novel surrogate model workflow that enables fast surrogate model construction by extending hyperparameter optimization. Then, we validate the performance of this workflow with the example of CO₂ sequestration in the saltwater layer.

2. METHODOLOGY

2.1 Formation flow control equation

In this work, we consider CO₂ injection into a 2D saltwater layer. The formation fluid phases are aqueous and gas phases, and the fluid components include water and CO₂. where the water component is present only in the aqueous phase. The CO₂ is mainly present in the gas phase, and part of the CO₂ is dissolved in the aqueous equivalent. Based on the law of mass conservation, the continuity equation of component i within a stratigraphic multiphase flow system can be described by the following equation:

$$-\nabla \cdot (C_{ig}\rho_g\mathbf{v}_g + C_{iw}\rho_w\mathbf{v}_w) + q_i = \frac{\partial}{\partial t} [\varphi(C_{ig}\rho_gS_g + C_{iw}\rho_wS_w)] \quad (1)$$

Further considering the effect of gravity and capillary pressure, the equation of motion of fluid seepage can be obtained by combining Eq. (1) and Darcy's law:

$$\nabla \cdot [K\lambda_i C_{il} \rho_l (\nabla P_l - \rho_l g \nabla Z)] + q_i = \frac{\partial}{\partial t} [\varphi(C_{il} \rho_l S_l)] \quad (2)$$

Note that the sum of the molar fractions of the components of the fluid phase is equal to 1, i.e. $\sum C_{il} = 1$. And the phase pressure is correlated

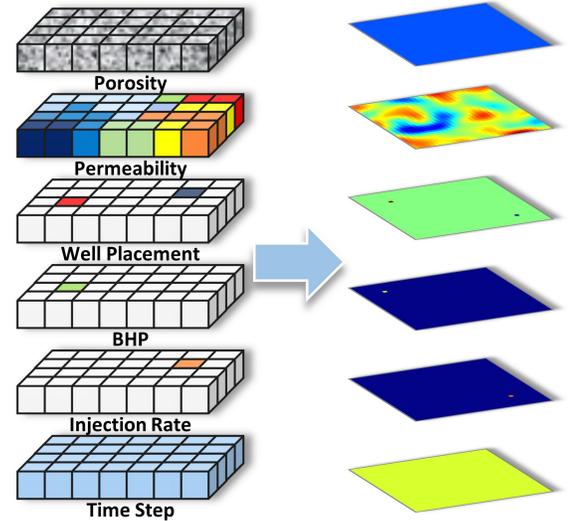


Fig. 1. 2D feature maps converted from the original input space.

according to the capillary pressure, i.e. $P_{cgw} = P_g - P_w$.

$$y \approx \hat{y} = \hat{S}(\hat{x}; \theta) \quad (4)$$

Therefore, the objective of this work is to

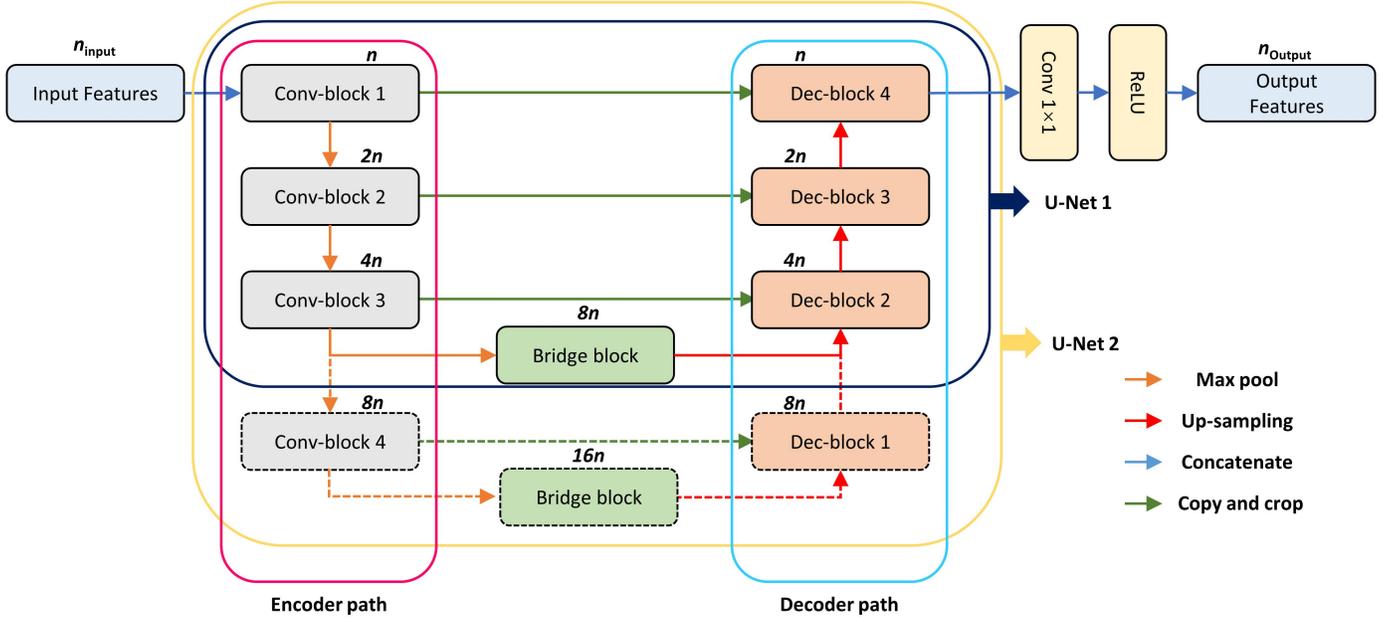


Fig. 2. Architecture of the U-Net for this study. The number of filters is labeled above the symmetric decoder and encoder blocks. There are two U-Net architectures (U-Net 1 and U-Net 2) included in this study, and the dotted lines refer to the additional blocks included in the deeper U-Net.

2.2 Surrogate models based on deep learning

The deep learning-based surrogate model is driven by simulated flow data to obtain mapping relationships between high-dimensional data, which in turn leads to a prediction of multiphase fluid flow states within non-homogeneous reservoirs. In the numerical simulation of the subsurface flow problem, the numerical simulator maps high-dimensional inputs to high-dimensional outputs by numerical computational methods. The relationship between the input and output of a single simulation can be considered as an approximate function of:

$$y = S(x) \quad (3)$$

The numerical simulation process maps multiple high-dimensional inputs to single (or multiple) high-dimensional outputs with the help of a numerical simulator. The surrogate model learns the mapping relationships from the original data space and thus makes predictions about the output state variables. After approximating the mapping relationships between inputs and outputs, the substitution process of the surrogate model is approximated as:

determine the neural architecture and loss function of the surrogate model based on the dataset in order to form a surrogate model for predicting the spatial and temporal evolution of the stratigraphic flow state variables. Considering the stratigraphic and production features that have a significant influence on the formation state variables during the numerical simulation, we determined the input features of the surrogate model from within the original input space. First, porosity, permeability, well placement, BHP of production well, and injection rate of injection wells are used as input features. Since the actual CO₂ sequestration involves continuous work over long periods, we discretize the input features into sequential time steps and incorporate them with the corresponding time steps accordingly. Further using formation pressure and saturation as output features, for the m th time step, Eq. (4) can be converted to:

$$y^m \approx \hat{y}^m = \hat{S}(\varphi, K, \text{Well}, p_{bh}^m, q_{inj}^m, t^m; \theta) \quad (5)$$

The main task of the surrogate model is to predict the output state variables (e.g., pressure and saturation) at each time step based on the input characteristics of the original input space. Fig. 2

illustrates treating both the input and output of the 2D reservoir model as 2D images. Accordingly, the task of this study is transformed into an image-to-image regression task, i.e., mapping the input feature maps to the output feature maps with the help of a neural architecture targeting image processing.

2.3 Variable U-Net architecture

The high-dimensional input features of the numerical simulation process (including permeability, porosity, etc.) have complex multi-scale spatial correlations, while these high-dimensional nonlinearities determine the spatiotemporal evolution of the output state map. For the complex spatiotemporal information contained in image signals (or image-like signals), CNNs designed to process image signals can capture them efficiently through filters inside the network architecture. CNNs usually include recursively connected convolutional layers, pooling layers, and fully connected layers. Among them, the lower-level convolutional layers will focus more on local features, while the higher-level convolutional layers focus on global information[7]. The sequentially connected layered structure forms a generic CNN architecture, where the relationship between adjacent layers can be characterized by the following function:

$$F_k = \sigma(W_k^o F_{k-1} + b_k) \quad (6)$$

To efficiently capture the spatial features inside the image signal and the complex nonlinear mapping relationships between the feature images, we chose the U-Net[8] architecture with a symmetric encoder-decoder structure in the CNN architecture. The width[9,10] and depth[11–13] of a neural network can significantly affect its performance. The width of a CNN is usually characterized by the number of filters inside the convolutional layers, while the number of layers inside the architecture determines the depth of the CNN. Moreover, redesigning the depth and width of the neural architecture requires a large number of manual

error-setting sessions.

To simplify the surrogate modeling process, the variable U-Net as shown in Fig. 1 is designed in this study. The dashed convolutional blocks in Figure 1 are the additional architecture of U-Net 2 compared to U-Net 1. The symmetric contracting and expansive paths that U-Net has to enable it to capture the high-dimensional nonlinear relationships between complex image signals more efficiently after discarding the fully connected layer.

Fig. 3(a) shows the composition of the convolutional block within the architecture, where repeating convolutional layers, a batch normalization layer, and a nonlinear activation (ReLU) layer are sequentially connected to form the basic convolutional block. The convolutional block Conv-block 1, which is directly connected to the input features, contains n filters. The input feature map passes through several successive convolution blocks in the encoder path in sequence, with the number of channels incrementing from n to $4n$ ($8n$), and then connects to the decoder path via the bridge block. Fig. 3(b) shows the upsampling operation on the U-Net decoder path, where the batch normalization and nonlinear activation functions are sequentially connected after the upper convolutional layer. The output feature map of the bridge block is upsampled and concatenated with the output of the convolution block of the corresponding layer of the encoder path. The feature channels are decremented from $8n$ ($16n$) at the bridge block to n via the decoder path and then mapped to the specified output channel n_{out} by a 1×1 convolution operation and a nonlinear activation function.

U-Net 1, U-Net 2, and the initial number of convolutional block filters n are included as relevant hyperparameters in the hyperparameter optimization process of the surrogate model. Accordingly, the architectural design aspects of the surrogate model are simplified and further incorporated into the hyperparameter optimization process.

2.4 Variable loss function

For the image-to-image regression task, the performance of the neural architecture is mainly characterized by the fidelity between the original image signal and the predicted image signal. This study characterizes the performance by comparing the simulation features with the output features of the surrogate model. MSE is widely used as a measure of signal fidelity because it is simple to calculate and has a clear physical meaning[14]. The MSE gives the degree of

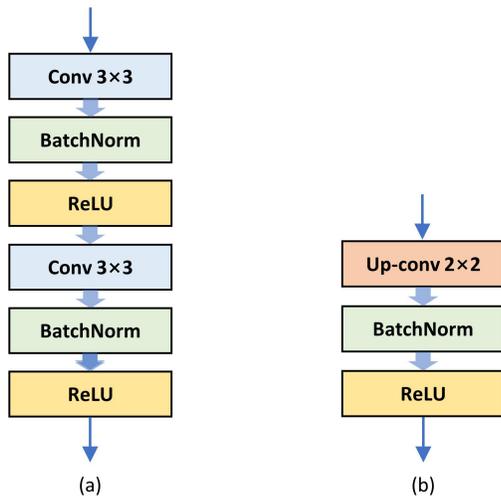


Fig. 3. Composition of the convolutional blocks inside the U-Net architecture.

fidelity between signals by averaging the square of the difference between the intensity of the predicted image pixels and the labelled image pixels. For both the output feature maps of the simulator and the surrogate model, the computation of MSE can be characterized by the following equation:

$$MSE(y, \hat{y}) = \frac{1}{N} \sum_{p=1}^N (y_p - \hat{y}_p)^2 \quad (7)$$

The formation flow state variables predicted by the surrogate model have complex spatial structure information, especially in the pressure and saturated plume leading edge and in the region close to the well. However, the spatial correlation between pixels within the image signal is ignored when using MSE alone as a fidelity measure. To more accurately measure the fidelity of the surrogate model, we introduced SSIM[15] with reference to previous studies[5,16,17]. The SSIM proposed by Wang et al.[15,18] forms a local SSIM by comparing local image blocks at the same location of two image signals. The local SSIM metric measures the similarity of brightness, contrast, and structure of the local images. For a particular two local image blocks, the SSIM is calculated as shown below:

$$SSIM(y_L, \hat{y}_L) = \frac{(2\mu_{y_L} \mu_{\hat{y}_L} + c_1)(2\sigma_{y_L \hat{y}_L} + c_2)}{(\mu_{y_L}^2 + \mu_{\hat{y}_L}^2 + c_1)(\sigma_{y_L}^2 + \sigma_{\hat{y}_L}^2 + c_2)} \quad (8)$$

Based on a fixed-size sliding window, the SSIM score of each local part of the image is obtained by moving the window pixel by pixel. For the overall fidelity measure of the complete image signal, we use the MSSIM metric to evaluate the overall image quality, it can be calculated from the following equation:

$$MSSIM(y, \hat{y}) = \frac{1}{M} \sum_{j=1}^M SSIM(y_L, \hat{y}_L) \quad (9)$$

We apply MSE and MSSIM to the loss function to measure the performance of the surrogate model in two directions: pixel-level intensity error and the degree of spatial structural distortion. In order to balance the contributions of MSE and MSSIM in a particular stratigraphic flow problem, we introduce the MSSIM weighting factor u . The detailed loss function is calculated as shown below:

$$\ell(y, \hat{y}) = MSE(y, \hat{y}) + u \cdot MSSIM(y, \hat{y}) \quad (10)$$

The weighting factor u is used as the relevant hyperparameter in the hyperparameter optimization process to obtain the best value based on the simulated flow data set.

2.5 Extended hyperparameter optimization

To avoid redesigning neural architectures and loss functions, we designed variable neural architectures and loss functions to support extended hyperparameter optimization. When the weighing factor u takes different values, the performance of the surrogate model cannot be accurately measured by directly comparing the value of loss functions. To objectively measure the performance of surrogate models composed of different neural architectures and loss functions, we define a joint performance metric based on RMSE and MSSIM, which is calculated as shown below:

$$score(y, \hat{y}) = \left(\sqrt{\frac{1}{N} \sum_{p=1}^N (y_p - \hat{y}_p)^2} + (1 - MSSIM(y, \hat{y})) \right)^{-1} \quad (11)$$

MSSIM takes values from 0 to 1, 1 when identical, and 0 when completely irrelevant. For the pre-processed feature maps (data between 0 and 1), the maximum value of RMSE is 1 and the minimum value is 0. Accordingly, the joint performance metric has a minimum of 0.5 and a maximum tends to infinity. Based on the joint performance metrics, we evaluate the performance of different surrogate models.

In this work, we implemented a neural architecture based on Pytorch[19] and chose the ADAM[20] as the optimizer. The extended hyperparameter optimization includes learning rate, batch size, neural architecture (U-Net 1 or U-Net 2), the number of initial convolutional layer filters n , and weighing factor u . Based on the hyperparameter optimization framework OPTUNA[21], we use the TPE algorithm[22] for hyperparameter optimization with the objective of maximizing the joint performance metrics. The TPE algorithm is a powerful tool for optimizing multivariate complex functions, and the details of the method are described by please refer to Bergstra et al. [22] and are not presented here

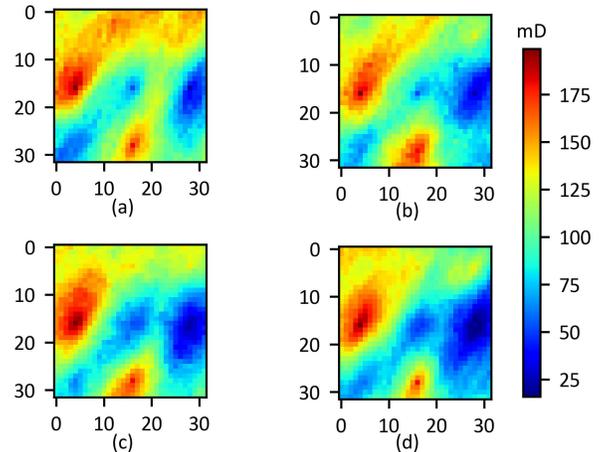


Fig. 4. Four homogeneous permeability fields..

additionally.

3. RESULTS AND DISCUSSION

3.1 The homogeneous 2D reservoir model

In this section, we construct a numerical simulation model to simulate the storage of CO₂ in the saltwater layer. The flow data obtained based on the numerical

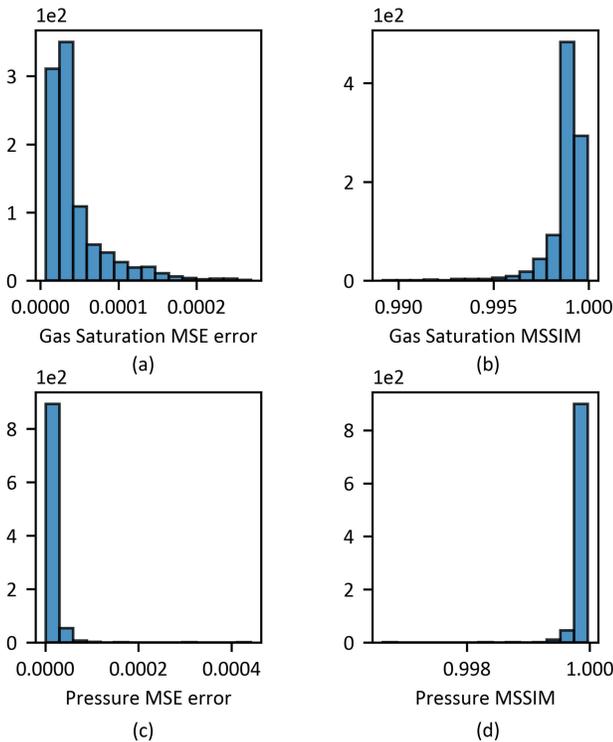


Fig. 5. MSE and MSSIM distributions of the surrogate model on the test set.

simulation model will be used to form a dataset to train and evaluate the surrogate model according to the proposed workflow. Based on the GEM commercial simulator developed by CMG, the 2D reservoir model is discretized using a uniform Cartesian grid divided into 32×32×1 grid cells in the x , y , and z directions, respectively. The homogeneous nature of the reservoir was mainly considered for the permeability field, and the porosity was all set to 0.2. In this work, we constructed four stochastic permeability fields using the Gaussian covariance model and Kriging geostatistical simulation method in the open-source Python package GSTOOLS[23]. Fig. 4 shows the specific distribution form of these permeability fields.

Further, we set 4 different well control parameters and 5 different well settings. The simulation time was set to 10 years, for a total of 120 output time steps. We divided the acquired flow data into 56 training cases (70%), 16 validation cases (20%), and 8 test cases (10%).

After the extended hyperparameter optimization is completed, the model is trained based on the optimized hyperparameter combinations. Test set cases will not be present in the training and optimization process to objectively characterize the performance and error distribution of the surrogate model.

3.2 Performance of the surrogate flow model

Based on the proposed workflow, we implemented the agent models for pressure and gas saturation, respectively. Comparing the simulated flow data, we validate the performance of the surrogate models on a test set.

On the test set, the mean MSE for gas saturation and pressure were 5E-5 and 2E-5, and the MSSIMs were 0.9989 and 0.9998, with the specific error distributions given in Fig. 5.

4. CONCLUSION

In this study, we propose a new deep learning-based surrogate model workflow to rapidly form models with high fidelity for CO₂ sequestration cases. Redesign sessions are avoided by constructing variable neural architectures and loss functions that can be used as relevant hyperparameters. Extended hyperparameter optimization replaces the original manual trial-and-error session to form high-performance surrogate models based on flow data. We validate the performance of the proposed workflow using a case study of CO₂ saltwater sequestration. The resulting surrogate model can accurately predict state variables of stratigraphic flow under different production parameters and well placement settings over a decade-long prediction time. Thus, the workflow based on extended hyperparameter optimization is expected to expand the application of the surrogate model and provide more timeous technical guidance for the practical work of carbon storage.

DECLARATION OF INTEREST STATEMENT

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. All authors read and approved the final manuscript.

REFERENCE

- [1] Jaber AK, Al-Jawad SN, Alhuraishawy AK. A review of proxy modeling applications in numerical reservoir simulation. Arab J Geosci 2019;12:701. <https://doi.org/10.1007/s12517-019-4891-1>.

- [2] Asher MJ, Croke BFW, Jakeman AJ, Peeters LJM. A review of surrogate models and their application to groundwater modeling: SURROGATES OF GROUNDWATER MODELS. *Water Resour Res* 2015;51:5957–73. <https://doi.org/10.1002/2015WR016967>.
- [3] Tang M, Liu Y, Durlofsky LJ. A deep-learning-based surrogate model for data assimilation in dynamic subsurface flow problems. *Journal of Computational Physics* 2020;413:109456. <https://doi.org/10.1016/j.jcp.2020.109456>.
- [4] Yan B, Harp DR, Chen B, Pawar R. A physics-constrained deep learning model for simulating multiphase flow in 3D heterogeneous porous media. *Fuel* 2022;313:122693. <https://doi.org/10.1016/j.fuel.2021.122693>.
- [5] Zhong Z, Sun AY, Jeong H. Predicting CO₂ Plume Migration in Heterogeneous Formations Using Conditional Deep Convolutional Generative Adversarial Network. *Water Resour Res* 2019;55:5830–51. <https://doi.org/10.1029/2018WR024592>.
- [6] Shokouhi P, Kumar V, Prathipati S, Hosseini SA, Giles CL, Kifer D. Physics-informed deep learning for prediction of CO₂ storage site response. *Journal of Contaminant Hydrology* 2021;241:103835. <https://doi.org/10.1016/j.jconhyd.2021.103835>.
- [7] Dumoulin V, Visin F. A guide to convolution arithmetic for deep learning 2018.
- [8] Ronneberger O, Fischer P, Brox T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Navab N, Hornegger J, Wells WM, Frangi AF, editors. *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, vol. 9351, Cham: Springer International Publishing; 2015, p. 234–41. https://doi.org/10.1007/978-3-319-24574-4_28.
- [9] Lu Z, Pu H, Wang F, Hu Z, Wang L. The Expressive Power of Neural Networks: A View from the Width 2017.
- [10] Hanin B, Sellke M. Approximating Continuous Functions by ReLU Nets of Minimal Width 2018.
- [11] Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition 2015.
- [12] Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, et al. Going Deeper with Convolutions 2014.
- [13] He K, Zhang X, Ren S, Sun J. Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA: IEEE; 2016, p. 770–8. <https://doi.org/10.1109/CVPR.2016.90>.
- [14] Zhou Wang, Bovik AC. Mean squared error: Love it or leave it? A new look at Signal Fidelity Measures. *IEEE Signal Process Mag* 2009;26:98–117. <https://doi.org/10.1109/MSP.2008.930649>.
- [15] Wang Z, Bovik AC, Sheikh HR, Simoncelli EP. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Trans on Image Process* 2004;13:600–12. <https://doi.org/10.1109/TIP.2003.819861>.
- [16] Maldonado-Cruz E, Pyrcz MJ. Fast evaluation of pressure and saturation predictions with a deep learning surrogate flow model. *Journal of Petroleum Science and Engineering* 2022;212:110244. <https://doi.org/10.1016/j.petrol.2022.110244>.
- [17] Zhong Z, Sun AY, Wang Y, Ren B. Predicting field production rates for waterflooding using a machine learning-based proxy model. *Journal of Petroleum Science and Engineering* 2020;194:107574. <https://doi.org/10.1016/j.petrol.2020.107574>.
- [18] Zhou Wang, Bovik AC. A universal image quality index. *IEEE Signal Process Lett* 2002;9:81–4. <https://doi.org/10.1109/97.995823>.
- [19] Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library 2019.
- [20] Kingma DP, Ba J. Adam: A Method for Stochastic Optimization 2017.
- [21] Akiba T, Sano S, Yanase T, Ohta T, Koyama M. Optuna: A Next-generation Hyperparameter Optimization Framework. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Anchorage AK USA: ACM; 2019, p. 2623–31. <https://doi.org/10.1145/3292500.3330701>.
- [22] Bergstra JS, Bardenet R, Bengio Y, Kégl B. Algorithms for Hyper-Parameter Optimization n.d.:9.
- [23] Müller S, Schüler L, Zech A, Heße F. GSTools v1.3: a toolbox for geostatistical modelling in Python. *Geosci Model Dev* 2022;15:3161–82. <https://doi.org/10.5194/gmd-15-3161-2022>.