

Fast Prediction of Immiscible Two-Phase Displacements in Heterogeneous Porous Media with Convolutional Neural Network

Wei Feng and Haibo Huang*

Department of Modern Mechanics, University of Science and Technology of China, Hefei, Anhui 230026, China

Received 26 December 2019; Accepted (in revised version) 4 March 2020

Abstract. A convolutional neural network is developed for rapidly predicting multiphase flow in heterogeneous porous media. Some direct numerical methods can acquire accurate results of multiphase flow in porous media. However, once the geometry of the porous media changes, it takes much computational time to perform a new simulation. Here, a deep neural network model in the field of semantic segmentation is developed. It takes the two-dimensional microstructure of heterogeneous porous media as inputs and is able to predict corresponding multiphase flow fields (pressure and saturation fields). Compared to the direct lattice Boltzmann simulations, the inference time on new geometry of porous media can be reduced by several orders of magnitude. Our results show that the machine learning method is a good prediction tool in a wide range of porosity and heterogeneity. Besides, to better understand the inherent process, a visible explanation is presented on what our neural networks have learned.

AMS subject classifications: 76S05, 76T99, 68T99

Key words: Porous media, multiphase flow, convolutional neural network, porosity, sorting.

1 Introduction

Multiphase flow in porous media widely exists in industrial applications such as geological carbon capture and storage, oil recovery and treatment of contaminated groundwater and soil, etc [1–4]. Studying and understanding the process of immiscible displacements in porous media are important. Due to different underlying heterogeneous pore geometry, wettability, fluid properties and flow conditions, there are several typical displacements patterns such as capillary fingering, viscous fingering and stable displacement [5]. In different pattern, there are different features on the saturation, the finger length, and

*Corresponding author.

Email: huanghb@ustc.edu.cn (H. B. Huang)

the size of the trapped clusters. For the capillary fingering, the invading fingers grow in all directions and the defending fluid may be trapped by the invading fluid. For the viscous fingering, the fingers grow like a tree and no defending fluid would be trapped. For stable displacement, the invading front is almost flat and moving forward stably.

Early works on multiphase flow in porous media are mostly experimental works and numerical simulations. The experimental studies focused on internal fluid distribution inside realistic rocks. With the help of advanced computer-aided imaging system (such as X-ray computed microtomography) [6–8], the pore-scale two-phase displacement in real sample porous rock can be obtained in real-time. Holtzman et al. [9] defined the disorder coefficient λ to represent the heterogeneity of synthetic porous media. They investigated the impact of λ on two-phase displacements by experiments. In the study of Hu et al. [10], the competing effect of disorder and wettability on the fluid displacements in porous media was investigated through microfluidic experiments and pore-scale simulations.

Numerical simulations methods, e.g., the lattice Boltzmann method (LBM) can achieve detailed flow fields (e.g., pressure, velocity and saturation field) in porous media [11–15]. These studies mainly paid attention to impacts of flow conditions and fluid properties on immiscible displacements. Few studies focused on the effects of the pore-scale disorder of porous media. Although the LBM results are accurate, the LBM simulations are computationally intensive. Once the porous medium changes, it would take much time to carry out a new simulation.

Among these years, deep learning methods [16] have been proposed as a new solution to classic physics problems [17, 18]. Their prediction accuracy, generalization ability, and especially computational efficiency are very impressive compared with experimental methods and numerical simulations. Wu et al. [19] proposed a convolutional neural network embedded with physics information to predict the permeability from the images of porous media. Since permeability is indeed a function of the pore geometry, using neural networks to learn the implicit mapping from the pore geometry to permeability is successful. The usage of a convolutional encoder-decoder network for image-to-image regression task has been explored by Zhu et al. [20]. Taking the log-permeability field of porous media as inputs, the model is able to give an approximation on the single-phase flow in heterogeneous porous media. The spatial information was taken into account since its inputs and outputs are two-dimensional sliced tensors in a real physical scene. Mo et al. [21] extended the above neural network to predict the two-phase flow in heterogeneous porous media. In his work, in addition to the original regression task for the pressure fields, there is another pixel-wise classification task for the saturation fields. Moreover, Jin et al. [22] proposed variational U-net network to predict the CO_2 saturation and pressure fields under the variation of random well locations in porous media. Similarly, Wang et al. [23] constructed a network to approximate the mapping from the permeability fields to the velocity fields. Zhu et al. [24] embedded the governing equations into the loss function, and trained the neural network without labelled data to get the pressure and flux fields in a random field. It is more efficient and physical. Actually, not only deep neural networks methodology, but also fully Bayesian surrogate model

was introduced in this field [25, 26]. The Bayesian model takes the microstructure of random heterogeneous media as inputs instead of the inputs of the permeability field, to predict response fields such as pressure fields. Similar to the deep learning process, encoding and decoding components are still included in its framework. The encoding module builds the mapping from the high-dimensional microstructure of heterogeneous porous media to a much lower-dimensional quantity. The decoding module is responsible for reconstructing the coarse-grained maps back to the fine-grained outputs.

Building on previous studies, here we take a deep convolutional neural network to learn the mapping from the two-dimensional microstructure of heterogeneous porous media to their internal multiphase flow. It is noticed that in the works mentioned above the permeability fields are large-scale macroscopic maps that cannot be obtained directly. They were obtained through a series of experiments. However, the inputs of our models are images of real spatial microstructure and can be directly observed. Besides, in previous works, the intrinsic dimensionality of the permeability fields (both in training sets and test sets) are controlled by the identical specific value, e.g., the length scale of the Gaussian random fields is identical in Zhu et al. [20]. However, our models are trained and evaluated under two-variable dimensions: porosity and sorting degree [27, 28]. Porous media in our work are modelled as randomly distributed non-overlapping spherical grains with different sizes in a square domain (see Fig. 6). Due to the randomness of grains, they have a wide range of porosity ϕ and grain size distribution δ (also called sorting). ϕ and δ are used to control the sparsity and heterogeneity of our synthetic porous media. Here, we focus on the effect of pore-scale disorder geometry of porous media on the two-phase displacements. Although our synthetic porous media are not real geological geometry, they are able to represent the general characteristics. We would like to demonstrate the feasibility of our method in prediction of two-phase displacements in the synthetic porous media.

There are two prediction tasks for our models. One is the regression task for continuous pressure fields, the other is the pixel-wise classification task for discontinuous saturation fields. Despite using the same network, they are trained independently due to different type of tasks. To capture multi-scale spatial information caused by different grain sizes, convolutional kernels with different dilation rates are utilized to extract the features in parallel. We would demonstrate the accuracy and efficiency of our models in predicting the pressure and saturation fields in heterogeneous porous media. We also have an insight into our trained models and try to understand what is learned by the neural network.

The rest of the paper is organized as follows. In Section 2, the pipeline for the entire deep learning method is introduced, including the problem formulation, the hyperparameters of our neural network and the preparation of the datasets. In Section 3, models are trained and their generalization performance on test sets are evaluated, then some explanations are presented to further understand what the neural network has learned. Finally, a summary is presented in Section 4.

2 Methodology

The overall procedures of this deep learning method are shown in Fig. 1. This framework consists of the following steps: (a) Generating random porous media, (b) Direct simulations with the LBM, (c) Training models, and (d) Predicting flow for new porous media. In step (a), a large number of two-dimensional geometries of synthetic porous media are generated in the chosen range of the parameter space. The details are presented in Section 2.3.1. In step (b), multiphase flows in the porous media are simulated by the LBM, then the numerical results are used to make up the datasets. The details are presented in Section 2.3.2. In step (c), we use the prepared datasets to train models. The details are presented in Section 3.1. In step (d), the trained models would be able to predict cases with new inputs that have never been seen before. The details are presented in Section 3.2.

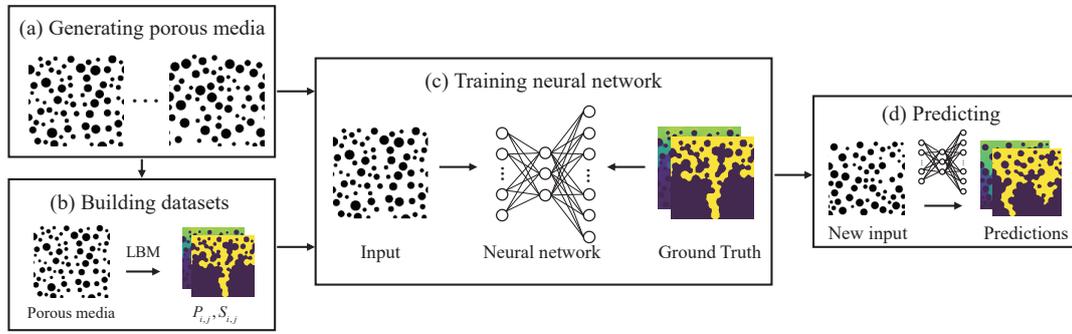


Figure 1: Overview of the computational framework.

2.1 Problem formulation

For displacements of two immiscible fluids in porous media, the governing equations take the following form [29]:

$$\frac{\partial}{\partial t}(\phi \rho_i s_i) + \nabla \cdot (\rho_i v_i) = q_i, \quad (i=1,2), \quad (2.1a)$$

$$v_i = -\frac{\mathcal{K} k_{ri}}{\mu_i} (\nabla p_i - \rho_i \mathbf{G}), \quad (2.1b)$$

$$s_1 + s_2 = 1, \quad (2.1c)$$

where \mathcal{K} is the intrinsic permeability of the porous media, k_{ri} represents the relative permeability of phase i . In above models the following parameters are known, i.e., porosity ϕ , phase density ρ_i , velocity v_i , viscosity μ_i , source item q_i , and the gravity \mathbf{G} . The unknowns are the phase pressures p_i and saturations s_i . The permeability is solely gov-

erned by the geometry of porous media. There is a classical permeability-porosity relation named Kozeny-Carman equation:

$$\mathcal{K} = \frac{\phi^3}{c(1-\phi)^2 A^2}, \quad (2.2)$$

where c and A are Kozeny constant and specific surface area, respectively. If the initial condition and q_i are fixed, there is a black-box mapping from the input \mathcal{X} to the output \mathcal{Y} in the following form:

$$\xi: \mathcal{X}(\Omega) \rightarrow \mathcal{Y}(P, S), \quad (2.3)$$

where Ω represents the boundary condition and also the microstructure of porous media. P and S are our target pressure and saturation fields, respectively.

In this work, the neural network is utilized as a surrogate model with the regression function $\mathbf{y} = \mathcal{F}(\mathbf{x}, \omega)$. The models are trained with the datasets $\mathcal{D} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$ (see Section 2.3.2). N is the sample size of training datasets, ω is the parameters in the neural network. Goal of the training process is making $\mathbf{y} = \mathcal{F}(\mathbf{x}, \omega)$ close to the intrinsic physical relation $\mathbf{y} = \xi(\mathbf{x})$ as much as possible.

2.2 Convolutional neural network

In computer vision, deep convolutional neural networks are adopted for three main tasks: classification, object detection, and segmentation. Classification aims to conjecture the category of the object in the image, e.g., handwritten digit recognition. Object detection means detecting the category and position of semantic objects in the image. The typical task of segmentation is locating objects and boundaries, such as lines, curves, etc. in the image. For example, the segmentation model is able to locate tumours and other pathologies in medical imaging [30]. It is a regression task from image to image. Models for segmentation can be used in physical space, building the mapping from one physical field to another unknown physical field.

2.2.1 Network architecture

The U-shaped architecture is widely used in recent research [31,32]. It consists of encoder and decoder modules. The encoder module repeats the downsampling operation to obtain sufficient receptive field and rich semantic information, and also reduces the feature map size. In the decoder module, detailed object boundaries and feature map size are gradually recovered by upsampling operation, until the size is the same as the size of inputs. Here, we follow the Deeplab v3+ [33] architecture to build our neural network. The detailed architecture and parameters of our models are shown in Fig. 2 and Table 1, respectively. There are several submodules included in the model.

Modified Xception: The modified Xception module [33] (illustrated in Fig. 3) is adopted as a backbone to extract features. It has shown promising image classification results [34].

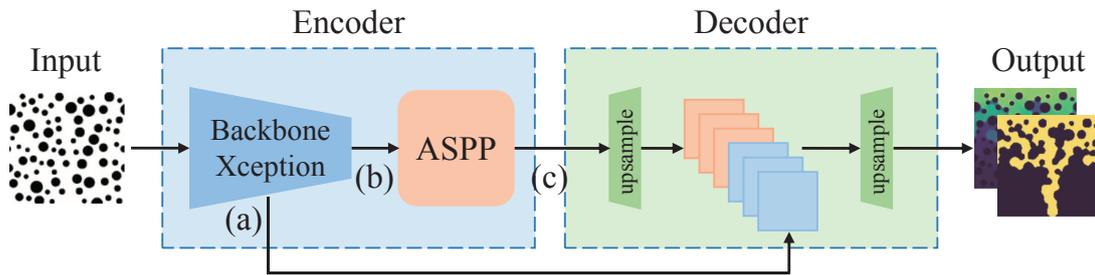


Figure 2: Architecture of our neural network. The designed neural network includes the encoder and decoder modules. Xception module (see Fig. 3) and the ASPP module (see Fig. 5) are adopted in the encoder module.

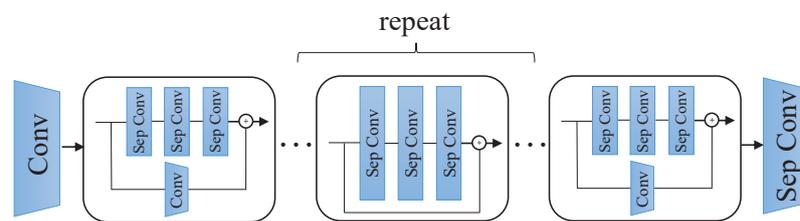


Figure 3: Schematic diagram of module Xception. The illustration of Separable Convolution (Sep Conv) in Xception is shown in Fig. 4. For simplicity, the common used Batch Normalization (BN) layers and ReLU activation layers after these convolution layers are not shown.

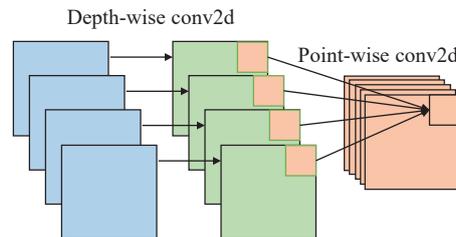


Figure 4: The composition of the separable convolutional layer, including a depth-wise convolution (applying a single filter for each channel in inputs) and a point-wise convolution (combining the outputs of depth-wise convolution across all channels).

The basic block of Xception (black frame in Fig. 3) is stacked by several separable convolution layers (see Fig. 4) with residual connections. In our model, we repeatedly stack the basic block six times to make the network deep and get the appropriate receptive field.

Separable Convolution: Instead of regular convolution layers, the separable convolution layers are used to make the network simple and efficient. It consists of a depth-wise convolution and a point-wise convolution. Depth-wise means each convolution filter extracts only one channel of features. Point-wise means each filter extracts only one pixel over all channels of features (see Fig. 4).

Atrous Spatial Pyramid Pooling: Unlike the standard convolution, atrous convolution

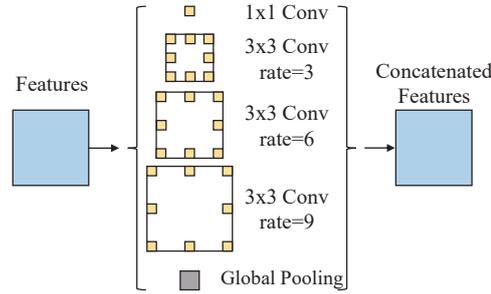


Figure 5: Parallel modules in Atrous Spatial Pyramid Pooling (ASPP). It has a convolution layer with 1×1 kernel, a global pooling layer and three atrous convolutional layers with dilation rates of [3, 6, 9].

can be applied on features with different resolutions by changing the rate value of r . For each location i on the input feature map Φ_{in} , output feature map Φ_{out} and a convolution filter ω , the atrous convolution is applied as follows:

$$\Phi_{out}[i] = \sum_k \Phi_{in}[i+r \cdot k] \omega[k]. \quad (2.4)$$

The module consisting of several parallel atrous convolutions with different rates (called Atrous Spatial Pyramid Pooling or ASPP) [35] is adopted in our model. Since the grain sizes in our porous media are different, the ASPP helps to capture the spatial information at multiple scales. The parallel modules in the ASPP are shown in Fig. 5. In the ASPP, there is a convolution layer with 1×1 kernel, a global pooling layer and three atrous convolutions with dilation rate of [3, 6, 9].

In the decoder module, if the features are upsampled back to the size of the inputs directly, much spatial information will be lost. For the problem, there is a simple and efficient solution [33]. The features with abundant high-level information (see Fig. 2(c)) are concatenated with the low-level outputs of the backbone (see Fig. 2(a)). Then, the concatenated features are upsampled back to the original size. With the combination of information on different levels, the result will be more accurate and have sharper object boundaries.

2.2.2 Loss function and training hyperparameters

When training and optimizing the weights in a neural network, the loss function indicates the direction of the gradient descent. Because the pressure is a continuous type of value, the smooth $L1$ loss (also called Huber loss) [36] is chosen to evaluate the error between predicted pressure fields and the LBM results. It is less sensitive to outliers than the regular $L2$ Loss, and in some cases, it prevents the explosion of gradients. The formula of Huber loss is

$$Loss_{Huber}(\hat{\mathbf{y}}, \mathbf{y}) = \frac{1}{M} \sum_{i=1}^M z_i + \alpha \Omega(\omega), \quad (2.5)$$

Table 1: Architectures for the network. The number following the bracket represents how many times we repeat the expression in the bracket. For simplicity, the commonly used Batch Normalization (BN) layers and ReLU activation layers are not counted in this table.

layer name		feature size	kernel size	stride	dilation	repeat
input		$1 \times 256 \times 256$	-	-	-	-
resize		$1 \times 128 \times 128$	-	-	-	-
Xception	Conv1	$32 \times 64 \times 64$	3×3	2	1	1
	Sep Conv1	$64 \times 32 \times 32$	$1 \times 1, [3 \times 3] \times 3$	$2, [1, 1, 2]$	1	1
	Sep Conv2	$64 \times 32 \times 32$	$[3 \times 3] \times 3$	$[1] \times 3$	1	6
	Sep Conv3	$80 \times 32 \times 32$	$1 \times 1, [3 \times 3] \times 3$	$[1] \times 4$	1	1
	Sep Conv4	$128 \times 32 \times 32$	$[3 \times 3] \times 3$	$[1] \times 3$	$[2] \times 3$	1
ASPP	Sep Conv5	$64 \times 32 \times 32$	1×1	1	1	1
			$[3 \times 3] \times 3$	$[1] \times 3$	$[3, 6, 9]$	
		average pool				
Decoder	Upsample	$64 \times 64 \times 64$	-	-	-	-
	Concatenate	$96 \times 64 \times 64$	-	-	-	-
	Sep Conv7	$1 \times 64 \times 64$	$[3 \times 3] \times 2, 1 \times 1$	$[1] \times 3$	1	1
	Upsample	$1 \times 128 \times 128$	-	-	-	-

in which

$$z_i = \begin{cases} 0.5(\hat{y}_i - y_i)^2, & |\hat{y}_i - y_i| < 1, \\ |\hat{y}_i - y_i| - 0.5, & |\hat{y}_i - y_i| \geq 1, \end{cases} \quad (2.6)$$

where $\hat{\mathbf{y}}$ is a predicted field and \mathbf{y} is a target field with a total of M elements, i denotes the i th element. ω denotes all of the parameters in the network. $\Omega(\omega) = \frac{1}{2}\omega^T\omega$ represents the L2 regularization. It is used to avoid over-fitting. The α is the weight decay coefficient.

For the prediction of saturation fields, we use a binary map to represent the distribution of two immiscible fluid phases. If a pixel in the image is occupied by the invading phase, the local saturation $S_{i,j}$ is set to be $S_{i,j}=1$, otherwise $S_{i,j}=0$ for the defending phase. This is an image segmentation task and the Dice loss (DL) [37] is selected as the loss function. The physical meaning of the Dice coefficient is the ratio between the intersection ($\hat{\mathbf{y}} \cap \mathbf{y}$) and the union ($\hat{\mathbf{y}} \cup \mathbf{y}$) of two regions. The DL can be written as,

$$DL = 1 - 2 \times \frac{\hat{\mathbf{y}} \cap \mathbf{y}}{\hat{\mathbf{y}} \cup \mathbf{y}}. \quad (2.7)$$

Suppose the predicted value in every pixel is p_i while the corresponding target value is t_i , the formula of Dice loss is

$$Loss_{Dice}(\hat{\mathbf{y}}, \mathbf{y}) = 1 - \frac{\sum_{i=1}^M 2p_i t_i + \epsilon}{\sum_{i=1}^M p_i + t_i + \epsilon} + \alpha \Omega(\omega), \quad (2.8)$$

where ϵ is very small and used to avoid the case that the denominator equals to zero.

There are several typical steps during the training of models. First, the input data are fed into the network. They will propagate through the entire network and in the

end, there is the output. Second, the deviation between the output and the LBM result is calculated according to the loss function. Third, the deviation is propagated reversely through the entire network and the gradient of each parameter along the propagating way is calculated. Finally, based on the calculated gradients, each parameter ω can be updated as

$$\omega' = \omega - \eta \frac{\partial \text{Loss}(\hat{\mathbf{y}}, \mathbf{y})}{\partial \omega}, \quad (2.9)$$

where η is the learning rate, $\frac{\partial \text{Loss}(\hat{\mathbf{y}}, \mathbf{y})}{\partial \omega}$ is the gradient of the parameter ω . Repeating the above process until the deviation is converged. Each repetition of the above process is one step. There are many steps until running out of the dataset, and we call the whole procedure one epoch. The whole iteration is accomplished by Pytorch [38] optimizer. The Adam [39] algorithm, which is one variant of the stochastic gradient descent (SGD) algorithms [40], is adopted for training in this work. Based on experiences, the initial learning rate is set to be 1×10^{-5} , and it decreases with the number of epochs linearly. The weight decay is set to be 1×10^{-3} . Total epochs of training iteration is 1,000.

2.3 Datasets

2.3.1 Porosity and grain size sorting of porous media

Since the porous media are heterogeneous, the radius r of each spherical grain is determined by

$$r(\beta) = \begin{cases} r_{\min}, & 0 \leq \beta < \delta, \\ r_{\min} + \frac{\beta - \delta}{1 - 2\delta} (r_{\max} - r_{\min}), & \delta \leq \beta \leq 1 - \delta, \\ r_{\max}, & 1 - \delta < \beta \leq 1, \end{cases} \quad (2.10)$$

where β is a uniformly distributed random number within the interval $[0, 1]$. The minimum radius r_{\min} and the maximum radius r_{\max} are selected as 5 lattice units ($l.u.$) and $20l.u.$, respectively. δ is the sorting parameter to control the sorting degree of grain sizes [14] and its range is $[0, 0.5]$. For random seeds β fall into the interval $[\delta, 1 - \delta]$, the radius is evenly distributed between r_{\max} and r_{\min} . Otherwise, it can only be r_{\min} or r_{\max} . When the porous medium is constructed, we ensure that any two grains would not overlap. Besides, to capture the surface tension precisely in the pore throat domain, their minimum gap between each other is not less than $5l.u.$ [41]. Seeds of locations and grain radius are generated randomly in a $256l.u. \times 256l.u.$ domain until the target porosity ϕ is achieved. The typical synthetic porous media with $\phi = 0.7, 0.8$ and $\delta = 0, 0.5$ are shown in Fig. 6. Porosity ϕ and sorting δ represent the sparsity and heterogeneity of porous media, respectively. When ϕ is large, there is more void space in the porous medium and fluid invasion becomes easier. When δ is large, the grains become easier to separate. For example, if $\delta = 0.5$, there are only two kinds of grains. One is the grains with $r = 5l.u.$, the other is the grains with $r = 20l.u.$. In this work, the porosity ϕ is chosen in the range of $0.70 - 0.80$, and the sorting degree δ ranges from 0 to 0.5.

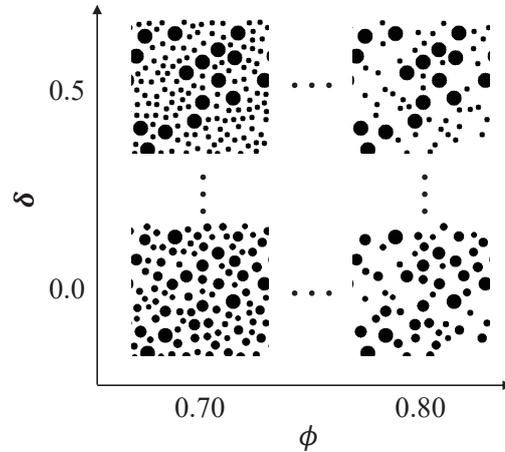


Figure 6: Samples of porous media in the parameter space. ϕ represents the porosity, δ represents the sorting degree of porous media.

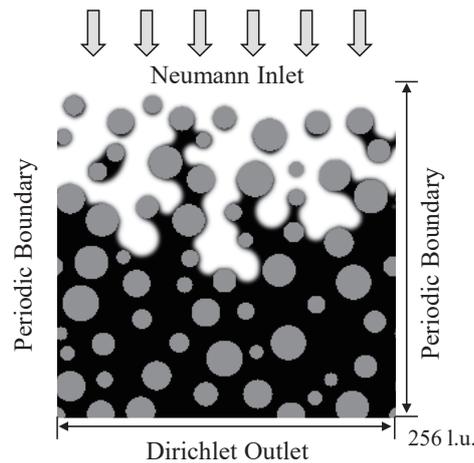


Figure 7: Snapshot of the two-phase displacement and set up for the boundary conditions. The white, black, and grey represent the invading, defending fluids, and solid, respectively. The invading fluid is injected from the upper boundary as indicated by the arrows.

2.3.2 Generation of datasets

The success of deep learning is partially attributed to high quality and large amounts of training data. In order to establish the datasets for training our models, a multiple-relaxation-time Rothman and Keller lattice Boltzmann method [13] is utilized to simulate the multiphase flow in porous media. From Fig. 7, we can see that the left and right sides of the computing domain are both periodic boundary. At the upper and lower sides of the computational domain, the inlet velocity and outlet pressure are specified, respectively. The domain is originally saturated by the wetting fluid (defending phase), then the non-

Table 2: Parameters for the LBM numerical simulations ($m.u.$ and $t.s.$ are mass unit and time step, respectively).

Items	Settings
Density (non-wetting fluid) ρ_n	$1.0m.u./l.u.^3$
Density (wetting fluid) ρ_w	$1.0m.u./l.u.^3$
Kinematic Viscosity (non-wetting fluid) ν_n	$0.333l.u.^2/t.s.$
Kinematic Viscosity (wetting fluid) ν_w	$0.033l.u.^2/t.s.$
Surface tension	$0.0272m.u./t.s.^2$
Contact angle (non-wetting fluid)	180°
Inject velocity	$0.01l.u./t.s.$

wetting fluid (invading phase) is injected into the domain to displace the wetting fluid. The breakthrough occurs when the invading fluid reaches the lower boundary. Then our simulation would stop and the flow fields at this time would be taken as a uniform standard result. Detailed simulation settings are shown in Table 2. Typically, each numerical simulation takes about 4 minutes on a NVIDIA RTX 2080Ti GPU.

We simulated multiphase flow in porous media with discrete porosity ϕ values in $[0.70, 0.72, 0.74, 0.76, 0.78]$, and δ is set to be zero. After the numerical simulations are accomplished, 80% of the data is chosen for training, to calibrate the weights in neural network. The other 20% would be used for validation, to monitor the training process. In order to avoid uneven distribution in datasets, stratified sampling strategy is utilized to divide datasets. The datasets are divided into training sets and validation sets in every porosity cases individually, then merge them together. Here, we generate a series of datasets with different amount ($N = 4,000, 8,000, 12,000, 16,000, 20,000$), aiming to see the convergence level and generalization performance with respect to the training sample sizes.

During training, the data normalization is always helpful for reaching a converged state. On the preprocessing of the datasets, the pressure fields are normalized in the following way,

$$P' = \frac{P - P_{\min}}{P_{\max} - P_{\min}}, \quad (2.11a)$$

$$P_{\min} = \operatorname{argmax}(\mathcal{PDF}\{P_{\min}^1, P_{\min}^2, \dots, P_{\min}^N\}), \quad (2.11b)$$

$$P_{\max} = \operatorname{argmax}(\mathcal{PDF}\{P_{\max}^1, P_{\max}^2, \dots, P_{\max}^N\}). \quad (2.11c)$$

The minimum and maximum pressure values over all training pressure field samples, i.e., P_{\min} and P_{\max} are calculated, respectively. Then the values with the highest probability density are chosen as the reference values. These two reference pressure values are saved for transforming our physical pressure fields to the normalized pressure fields.

2.3.3 Displacement patterns in datasets

In the immiscible two-phase displacements in porous media, we mainly focus on the flow pattern in saturation fields. Fig. 8 shows samples of data from the training sets. Though

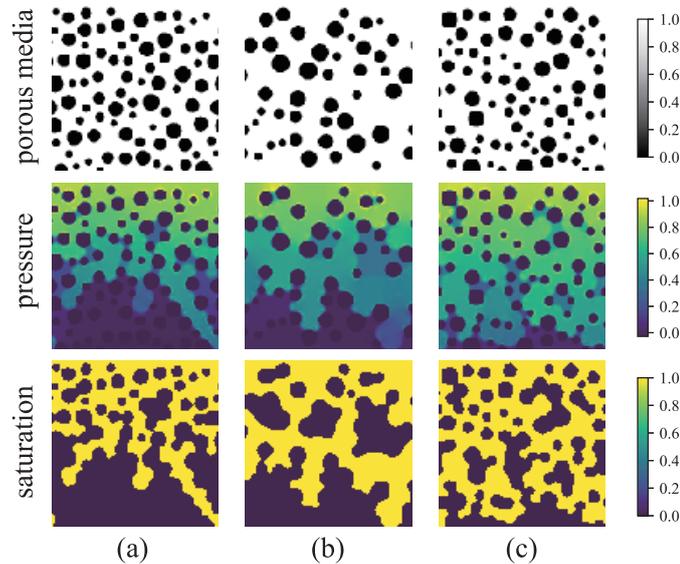


Figure 8: Samples of data from the training datasets with porous media of (a) $\phi = 0.70$, $\delta = 0$, (b) $\phi = 0.78$, $\delta = 0$, and (c) $\phi = 0.76$, $\delta = 0$. In the saturation maps, pores filled with invading fluid are marked in yellow but the solid and defending fluid are not distinguished.

the fluid properties and flow conditions in our simulations are fixed, there are different displacement patterns in the saturation fields due to the random pore geometry. From Fig. 8(a), we can see that the flow pattern is viscous fingering. The fingers grow towards the outlet like a tree. In Fig. 8(b), the flow pattern is capillary fingering. Some clusters of defending fluids are trapped in loops formed by fingers which grow in many directions. In Fig. 8(c), it is a stable displacement. The invading fluids present a flat front with only some irregularities at the pore scale. In addition, there are many transition flow patterns in our datasets which are not shown. Our models are expected to have a good prediction on these varied flow patterns.

3 Numerical experiments and discussion

3.1 Model training

Two models for predicting pressure and saturation fields are trained individually under different sizes of datasets. It takes about 5.0 to 20.5 hours on NVIDIA RTX 2080Ti GPU to train one model. The monitored loss as a function of the epoch during the training is shown in Fig. 9. As the number of epochs increases, the loss values on the training and validation datasets both decrease and reach a plateau. It shows that the neural network is gradually adjusting its weights to better fit the training datasets when the number of epochs increases. It also shows the neural network has a better prediction when it faced

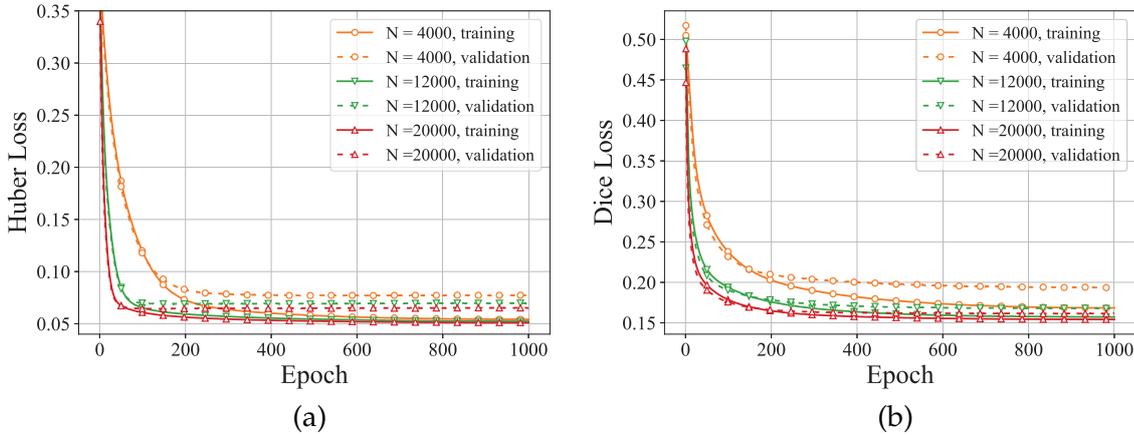


Figure 9: The loss as a function of the number of epochs under different sizes of datasets ($N=4,000, 12,000, 20,000$). For each dataset, 80% of the data is chosen for training and the other 20% would be used for validation. (a) Huber loss for pressure fields. (b) Dice loss for saturation fields. The solid and dotted lines represent the loss on the training sets and the corresponding validation sets, respectively.

with new data that have never been seen before. When more training data is used, the minimum loss (plateau value) would be smaller. For example, the minimum Huber loss on the validation datasets are 0.077 and 0.065 for the cases with training data $N=4,000$ and $N=20,000$, respectively. It can be concluded that more data is helpful for training and can get better convergence. But it tends to be saturated even though trained with more data. For example, the minimum loss on the training datasets for $N=12,000$ and $N=20,000$ are almost identical. Therefore, the fitting and predicting capability would not be further improved after they become saturated due to the capacity limitation of the neural network.

3.2 Model evaluation

Once the models get well trained, they are stored as files. Using the packaged model files, we can get the predictions of pressure and saturation fields for new porous media. The prediction is very fast. According to calculations, on the input with 256×256 pixels, our models have an average inference time on the order of seconds. Specifically, it takes an average of 6.7 milliseconds on a piece of NVIDIA RTX 2080Ti GPU, and 0.13 second on a piece of Intel Core i5-4460 CPU. In contrast, even adopting the parallel numerical algorithm on GPU, it takes the LBM about 4 minutes to finish one simulation. The computational costs of our models is one of tens of thousands than that of numerical methods. In this section, three evaluation experiments are conducted to investigate the prediction performance of our trained models. It is noticed that the data in the test sets are not included in the training set.

3.2.1 Test sets and evaluation metrics

Our test datasets have two dimensions ϕ and δ . The porous medium with different ϕ and δ has different characteristic. The porosity ϕ denotes the sparsity degree and may partially quantify the difficulty of fluid invasion. Generally speaking, the smaller the ϕ is, the more difficultly the fluid can enter. The sorting parameter δ indicates the different size distribution of grains. Both ϕ and δ would influence the hydraulic conductivity of porous media and permeability. Here, three evaluation experiments are carried on three typical porous media with different characteristics. The two dimensions ϕ and δ of the porous media are presented in Table 3. In Case 1, the test set has the same δ as the training sets but a different range of ϕ . It can test the performance of the neural network under the change of porosity. In Case 2, the test set has the same ϕ as the training sets but a different range of δ . We make a random uniform sample of δ within $\{0, 0.5\}$, to see whether the generalization performance is still good in variable sorting degrees. Further, in Case 3, there are totally different ϕ and δ between training sets and test sets. By controlling ϕ and δ , we want to illustrate the generalization performance of our models for porous media with a wide range of scenario. There are 500 samples in every test set. The stratified sampling strategy is not only used in the training set, but also in the test set for different porosity cases.

Coefficient of determination (R^2 score) and mean intersection over union ($mIoU$) are used to quantify the performance of the trained models on test sets. They are defined as

$$R^2 = 1 - \frac{\sum_{j=1}^{N_{test}} \|\hat{\mathbf{y}}^j - \mathbf{y}^j\|_2^2}{\sum_{j=1}^{N_{test}} \|\mathbf{y}^j - \bar{\mathbf{y}}\|_2^2}, \tag{3.1a}$$

$$mIoU = \frac{1}{c} \frac{1}{N_{test}} \sum_{k=1}^c \sum_{j=1}^{N_{test}} \frac{\sum_{i=1}^M p_{i,k}^j t_{i,k}^j + \epsilon}{\sum_{i=1}^M p_{i,k}^j + t_{i,k}^j - p_{i,k}^j t_{i,k}^j + \epsilon} \times 100\%, \tag{3.1b}$$

where $\bar{\mathbf{y}} = \frac{1}{N_{test}} \sum_{j=1}^{N_{test}} \mathbf{y}^j$, N_{test} is the sample size of the test datasets, k is the index of category, c is number of category. R^2 score is used to evaluate the approximation of the continuous pressure fields. A higher R^2 score indicates a higher degree of approximation for the target map. $mIoU$ is used to assess the approximation of the binary saturation fields. The

Table 3: Parameters ϕ and δ in three cases. The expressions in square bracket stand for [start:step:end]. For example, [0.70:0.02:0.78] is equivalent to the sequence of discrete value in [0.70, 0.72, 0.74, 0.76, 0.78]. The braces represent uniform sampling within {start, end}. The bold numbers highlight the differences between the training and test sets.

Case No.	Training set		Test set	
	ϕ	δ	ϕ	δ
1	[0.70:0.02:0.78]	0	[0.71:0.02:0.79]	0
2	[0.70:0.02:0.78]	0	[0.70:0.02:0.78]	{0, 0.5 }
3	[0.70:0.02:0.78]	0	[0.71:0.02:0.79]	{0, 0.5 }

higher the $mIoU$, the larger the overlap area between the prediction map and the target map.

3.2.2 Accuracy of predictions

On quantitative data, the evaluation scores in test sets are shown in Fig. 10. The overall evaluation scores in Case 1 are higher than those of other cases. A possible explanation is that the structure characteristic of porous media in Case 1 is the most similar to that in the training set among the three cases. Case 1 achieves a high score with $R^2 = 0.943$ and $mIoU = 92.21\%$ with 20,000 training samples. However, in Cases 2 and 3, due to the discrepancy of ϕ and δ between training and test sets, the scores drop a little bit. But even that with only 4,000 training samples, the models still achieves $R^2 = 0.871$ and $mIoU = 85.85\%$ in tough Case 2. Increasing the sample size to 20,000, the above scores reach to relatively high $R^2 = 0.916$ and $mIoU = 89.56\%$. It is also seen that the scores in test sets increase as the number of training samples N increases. It shows that lots of data can promote the generalization ability of models. However, the models are getting saturated gradually while the growth rate of the evaluation scores is gradually slowing down as N increases. Unless we make the models deeper, the generalization ability would not be further improved. Unfortunately, deeper neural networks with more capacity require more training data and time. There is a trade-off between the generalization ability and the computational costs of models.

The improvement of the accuracy of the network is usually attributed to the ASPP module and the skip connection (SC) between low-level and high-level features. To verify the point, a network without both the SC and ASPP, i.e., a conventional encoder-decoder

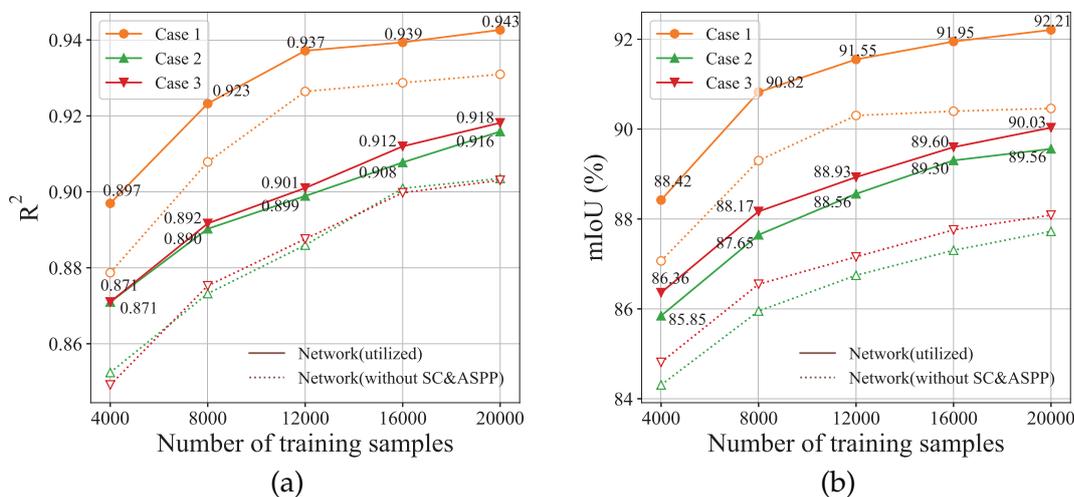


Figure 10: Evaluation scores on test sets as a function of training sample numbers. The solid and dashed lines denote the results of the utilized network and the network without the SC and ASPP, respectively. (a) the R^2 scores for the prediction of pressure fields. (b) the $mIoU$ scores for the prediction of saturation fields.

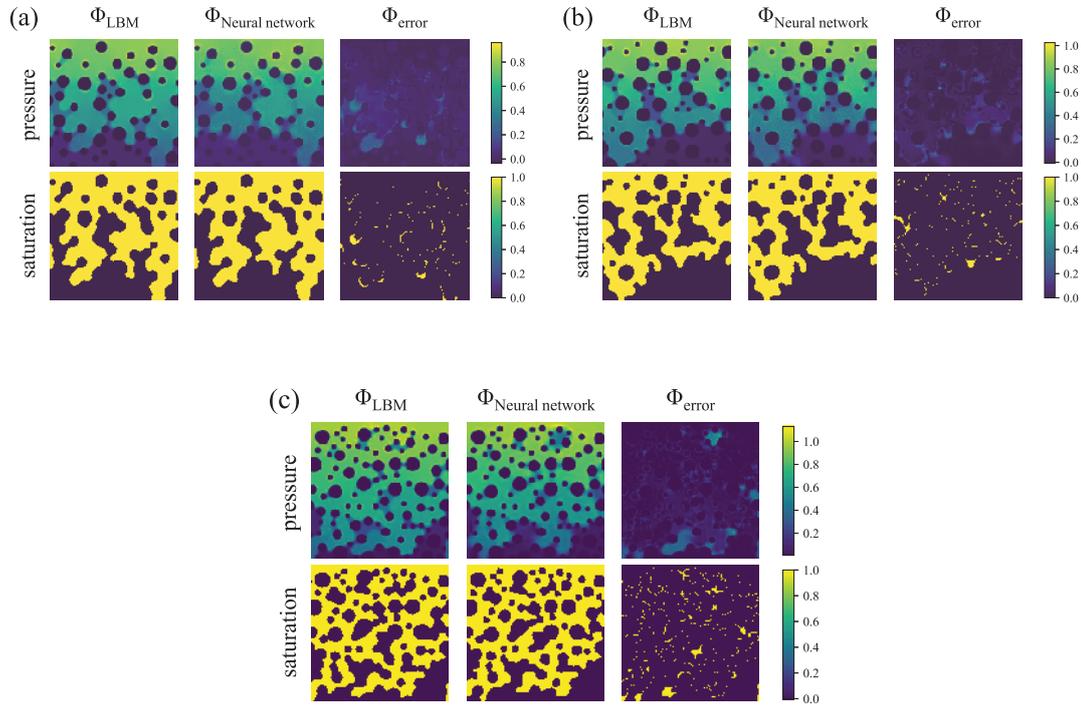


Figure 11: Samples of the pressure fields and saturation fields in (a) Case 1 (porous media with $\phi=0.79$, $\delta=0$), (b) Case 2 ($\phi=0.74$, $\delta=0.45$) and (c) Case 3 ($\phi=0.71$, $\delta=0.12$). The left, middle and right columns represent the LBM results, the neural network predictions and their errors, respectively. In saturation maps, pores filled with invading fluid are marked in yellow, the rest are solid and defending fluid.

based model, is used for comparison. The performances of our utilized network and the network without the SC & ASPP are shown in Fig. 10. It is seen that the utilized network achieves higher scores than the network without the SC & ASPP in all three cases. For example, the network without the SC and ASPP achieves $mIoU = 88.08\%$ with 20,000 training samples in Case 3, which is lower than the utilized network (90.03%). Hence, our network with the SC and ASPP is more accurate.

The qualitative comparison is shown in Fig. 11, samples of the LBM results, the neural network predictions and their errors are shown in each column, respectively. The error Φ_{error} is defined as follow,

$$\Phi_{error} = \|\Phi_{LBM} - \Phi_{predicted}\|_1, \quad (3.2)$$

where Φ_{LBM} and $\Phi_{predicted}$ represent the LBM results and the neural network predictions, respectively. It is seen that the predicted pressure field agrees well with that of the LBM. On the prediction of saturation fields, the models can predict well on the general shape and right location of invading fingers. Also, the region of trapped defending fluid can

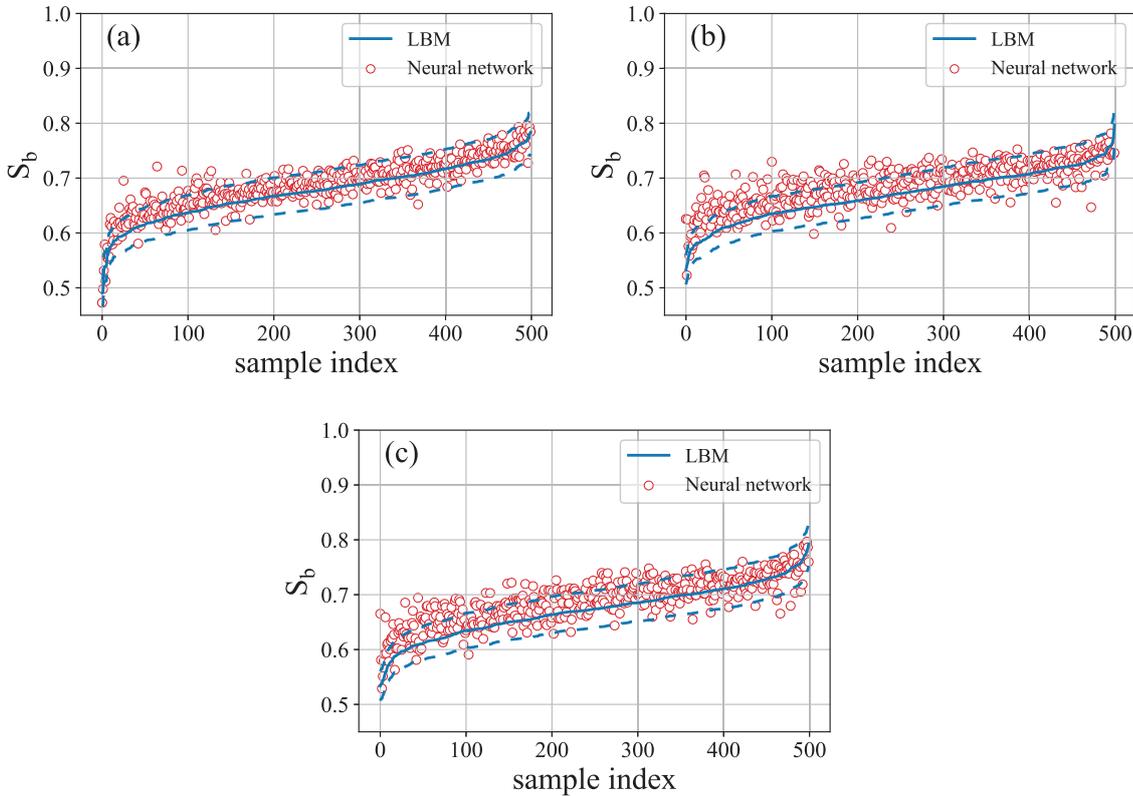


Figure 12: The comparison of breakthrough saturation S_b between results of the LBM and predictions of our models in (a) Case 1, (b) Case 2 and (c) Case 3. The solid line stands for the ground truth simulated by the LBM. The scattered points denote the prediction of our models. The area between the two dashed lines shows the region within 5% relative error.

be identified. More importantly, the models are adaptable for different flow patterns. For example, there are three different typical flow patterns: viscous fingering (Fig. 11(a)), capillary fingering (Fig. 11 (b)) and stable displacement (Fig. 11(c)). In different flow patterns, the invading fingers and trapped clusters have different behaviors. This remains a challenge for our models. Especially, when the δ of the porous media in Case 2 and Case 3 is variable (see Figs. 11(b) and (c)), there are many small grains surround the big grains. Although the models have never been trained by data like this, they are still able to predict well. It shows that our models have learned the essence of this physical phenomenon, not just remembering and interpolating.

The predicted error of saturation fields is mostly at the front of the saturation profile. It is likely that our loss function (Huber loss) mainly penalizes the large error on the general positions of invading fingerings and trapped clusters. The small errors at the front of the saturation profile are not apparent enough to lead the training of the neural networks. In addition, sometimes it is failed to predict the connection or disconnection between ad-

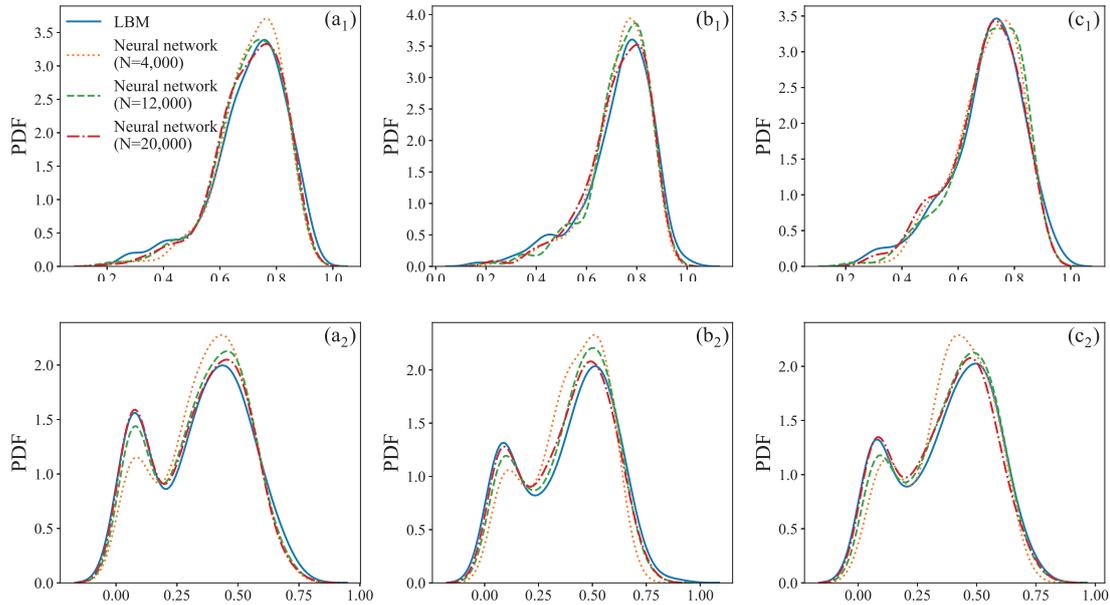


Figure 13: Distribution estimate for the pressure at two locations: (0.5, 0.33) (upper row) and (0.5, 0.75) (lower row) in Case 1 (left column), Case 2 (middle column) and Case 3 (right column). Our neural networks are trained with datasets of sizes $N=4,000, 12,000, 20,000$.

adjacent interfaces in the throat. It means that there is a high degree of uncertainty in the predictions of these particular regions.

The breakthrough saturation S_b is defined as follow,

$$S_b = \frac{A_i}{A - A_s}, \tag{3.3}$$

where A is the total area of computing domain, A_s is the area of solids and A_i is the area of domain occupied with invading phase. We calculate the S_b according to the saturation fields, then compare the predicted S_b with that of the LBM results in Fig. 12. The solid line represents the LBM result. Because of the randomness of the porous media geometry, every case of the simulations has different breakthrough saturation. The curves of S_b are shaped like ‘S’. The scattered points are the predictions of our models. The area between the two dashed lines shows the region within 5% relative error. It is seen that most scattered points are close to the solid line, and inside the error region.

Fig. 13 shows the probability density function (PDF) of pressure at locations (0.5, 0.33) and (0.5, 0.75) over all test sets. The PDF of pressure is fitted by a univariate or bivariate kernel density estimate. Since the (0.5, 0.33) is located at upstream, it has low uncertainty. In Figs. 13(a₁)-(c₁), the predictions of density estimate by neural networks are close to the results of the LBM. As we increase the sizes of training datasets, they can coincide better. From Figs. 13(a₂)-(c₂), we can see that at the location (0.5, 0.75) in downstream,

the uncertainty increases, and the deviation of PDF between the neural networks and the LBM also increases. However, the deviation is reduced when the size of training datasets increase.

3.3 Visualization of features in models

The neural network has always been treated as a black box system with uninterpretable characteristics. It can be used to fit implicit nonlinear mapping, but can not produce explicit function formulas. The mapping relationship is interpreted as weights and bias in neurons and stored in model file invisibly. However, there are still some efforts dedicated to interpreting the function of deep neural network [19, 42–44]. In order to have a glance at how the neural network works, we feed the trained neural network with the three samples of porous media in Section 2.3.3. There are many layers of features along the way that inputs passing through the whole network. Each layer of feature corresponds to one convolution filter kernel. Every kernel and its input interact to produce a piece of feature. Here, some features from different depths of the neural network are chosen to show in Fig. 14. It is helpful to take an inside look and understand what our neural network saw from the geometry of porous media.

The features in different depths of model (position (a), (b) and (c) in Fig. 2) are extracted as low-level, middle-level and high-level features, respectively. Fig. 14(b) show

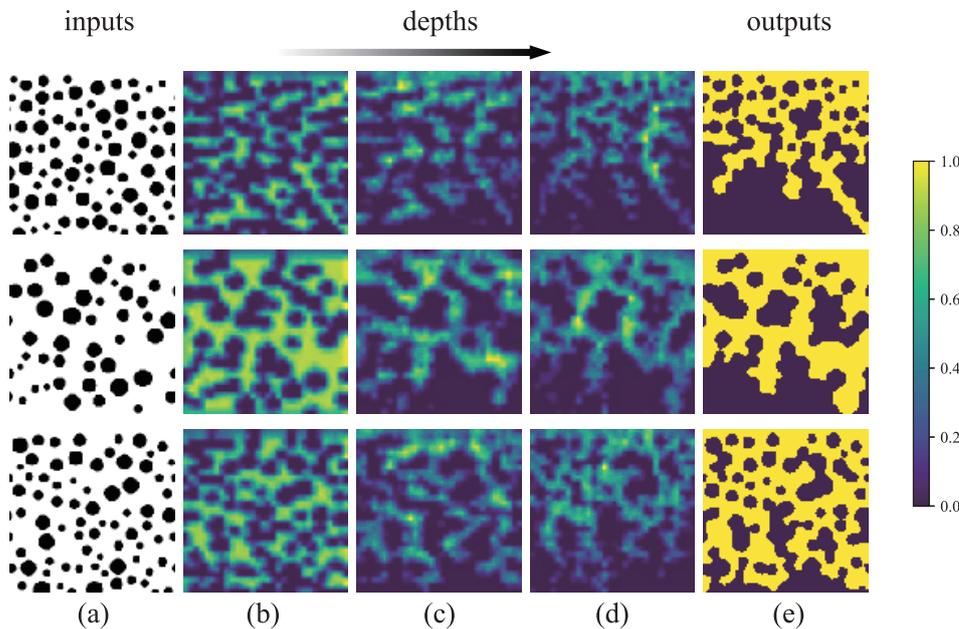


Figure 14: Visualization of features sampled in different depths of the trained models for three typical inputs. (a) the geometry of input porous media, (b) the low-level features, (c) the middle-level features, (d) the high-level features, and (e) the saturation field of outputs.

the low-level feature maps. The low-level feature is correspond to the void domain of the inputs (see Fig. 14(a)). The chosen convolution filter is able to identify the void space in the porous medium. Next, as these low level features passing forward and goes deeper, features are concatenated to be higher level features. In the middle-level features maps (Fig. 14(c)), we can see that the isolated highlights in Fig. 14(b) gradually degenerate and even disappear. Meanwhile, the highlights with relatively large area can be reserved. Finally, in the high-level feature maps (Fig. 14(d)), these highlights evolve into a connected network structure. The connection between highlights are also strengthened. They have the same profile with the corresponding saturation fields of outputs (Fig. 14(d)). After concatenated with the low-level features, they can be directly restored to the desired output by upsampling. This is the whole procedure that how our trained neural network predicts on the geometry of input. The above visualizations and explanations may help us to understand what is inside the neural network, and how our inputs are transformed into our desired outputs.

4 Conclusions

In this work, a deep semantic segmentation convolutional neural network is utilized to predict the multiphase flow in heterogeneous porous media. The random two-dimensional geometries of porous media are the inputs of the neural networks. It is an intrinsic microstructure. The predictions for porous media with a wide range of porosity ϕ and sorting δ are evaluated. Our models achieve accurate predictions for both the pressure and saturation fields. Especially, they are adaptable for different displacement patterns. On quantitative statistics, not only the breakthrough saturation S_b but also the PDFs of pressure agree well with the LBM result. It is also noticed that although the ϕ and δ of porous media in test sets may deviate from that in training sets, our models are still able to predict the pressure and saturation fields satisfactorily.

The fast prediction is computational efficient since it is orders of magnitude faster than numerical simulations. By visualizing internal features, we also partially explained the learning process of the neural networks.

Our current models deal with the one moment that the 'breakthrough' occurs. However, the displacement process is a time-dependent dynamic system. In the near future, we would extend our one-frame output to multi-frame outputs at many time instances with recurrent neural network.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Nos. 11572312 and 11621202).

References

- [1] J. D. JANSEN, *Adjoint-based optimization of multi-phase flow through porous media—a review*, *Comput. Fluids*, 46(1) (2011), pp. 40–51.
- [2] M. S. VALAVANIDES, *Review of steady-state two-phase flow in porous media: independent variables, universal energy efficiency map, critical flow conditions, effective characterization of flow and pore network*, *Transp. Porous Media*, 123(1) (2018), pp. 45–99.
- [3] A. Z. ZINCHENKO AND R. H. DAVIS, *Motion of deformable drops through porous media*, *Annu. Rev. Fluid Mech.*, 49 (2017), pp. 71–90.
- [4] J. J. L. HIGDON, *Multiphase flow in porous media*, *J. Fluid Mech.*, 730 (2013), pp. 1–4.
- [5] R. LENORMAND, E. TOUBOUL, AND C. ZARCONE, *Numerical models and experiments on immiscible displacements in porous media*, *J. Fluid Mech.*, 189 (1988), pp. 165–187.
- [6] S. BERG, H. OTT, S. A. KLAPP, A. SCHWING, R. NEITELER, N. BRUSSEE, A. MAKURAT, L. LEU, F. ENZMANN, AND J.-O. SCHWARZ ET AL., *Real-time 3D imaging of haines jumps in porous media flow*, *Proc. Natl. Acad. Sci. USA*, 110(10) (2013), pp. 3755–3759.
- [7] A. GEORGIADIS, S. BERG, A. MAKURAT, G. MAITLAND, AND H. OTT, *Pore-scale micro-computed-tomography imaging: Nonwetting-phase cluster-size distribution during drainage and imbibition*, *Phys. Rev. E*, 88(3) (2013), 033002.
- [8] J.-Q. SHI, Z. XUE, AND S. DURUCAN, *Supercritical CO₂ core flooding and imbibition in taso sandstone—influence of sub-core scale heterogeneity*, *Int. J. Greenhouse Gas Control*, 5(1) (2011), pp. 75–87.
- [9] R. HOLTZMAN, *Effects of pore-scale disorder on fluid displacement in partially-wettable porous media*, *Sci. Rep.*, 6 (2016), pp. 36221.
- [10] R. HU, T. LAN, G.-J. WEI, AND Y.-F. CHEN, *Phase diagram of quasi-static immiscible displacement in disordered porous media*, *J. Fluid Mech.*, 875 (2019), pp. 448–475.
- [11] Q. KANG, P. C. LICHTNER, AND D. R. JANECKY, *Lattice Boltzmann method for reacting flows in porous media*, *Adv. Appl. Math. Mech.*, 2(5) (2010), pp. 545–563.
- [12] H. HUANG AND X.-Y. LU, *Relative permeabilities and coupling effects in steady-state gas-liquid flow in porous media: A lattice Boltzmann study*, *Phys. Fluids*, 21(9) (2009), 092104.
- [13] H. HUANG, J.-J. HUANG, AND X.-Y. LU, *Study of immiscible displacements in porous media using a color-gradient-based multiphase lattice Boltzmann method*, *Comput. Fluids*, 93 (2014), pp. 164–172.
- [14] H. LIU, Y. ZHANG, AND A. J. VALOCCHI, *Lattice Boltzmann simulation of immiscible fluid displacement in porous media: Homogeneous versus heterogeneous pore network*, *Phys. Fluids*, 27(5) (2015), 052103.
- [15] C. HUANG, B. SHI, N. HE, AND Z. CHAI, *Implementation of multi-GPU based lattice Boltzmann method for flow through porous media*, *Adv. Appl. Math. Mech.*, 7(1) (2015), pp. 1–12.
- [16] Y. LECUN, Y. BENGIO, AND G. HINTON, *Deep learning*, *Nature*, 521(7553) (2015), pp. 436.
- [17] K. DURAISAMY, G. IACCARINO, AND H. XIAO, *Turbulence modeling in the age of data*, *Annu. Rev. Fluid Mech.*, 51 (2019), pp. 357–377.
- [18] C. SHEN, *A transdisciplinary review of deep learning research and its relevance for water resources scientists*, *Water Resour. Res.*, 54(11) (2018), pp. 8558–8593.
- [19] J. WU, X. YIN, AND H. XIAO, *Seeing permeability from images: fast prediction with convolutional neural networks*, *Science Bulletin*, 63(18) (2018), pp. 1215–1222.
- [20] Y. ZHU AND N. ZABARAS, *Bayesian deep convolutional encoder–decoder networks for surrogate modeling and uncertainty quantification*, *J. Comput. Phys.*, 366 (2018), pp. 415–447.
- [21] S. MO, Y. ZHU, N. ZABARAS, X. SHI, AND J. WU, *Deep convolutional encoder-decoder networks*

- for uncertainty quantification of dynamic multiphase flow in heterogeneous media, *Water Resour. Res.*, 55(1) (2019), pp. 703–728.
- [22] L. JIN, H. LU, AND G. WEN, *Fast uncertainty quantification of reservoir simulation with variational U-Net*, arXiv preprint, arXiv:1907.00718, (2019).
- [23] Y. WANG AND G. LIN, *Efficient deep learning techniques for multiphase flow simulation in heterogeneous porous media*, arXiv preprint, arXiv:1907.09571, (2019).
- [24] Y. ZHU, N. ZABARAS, P.-S. KOUTSOURELAKIS, AND P. PERDIKARIS, *Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data*, *J. Comput. Phys.*, 394 (2019), pp. 56–81.
- [25] C. GRIGO AND P.-S. KOUTSOURELAKIS, *A physics-aware, probabilistic machine learning framework for coarse-graining high-dimensional systems in the Small Data regime*, arXiv preprint, arXiv:1902.03968, (2019).
- [26] C. GRIGO AND P.-S. KOUTSOURELAKIS, *Bayesian model and dimension reduction for uncertainty propagation: applications in random media*, *SIAM/ASA J. Uncertain. Quantif.*, 7(1) (2019), pp. 292–323.
- [27] J. DVORKIN, AND M. A. GUTIERREZ ET AL., *Grain sorting, porosity, and elasticity*, *Petrophysics*, 43(03) (2002).
- [28] J. GUNNING AND M. E. GLINSKY, *Detection of reservoir quality using Bayesian seismic inversion*, *Geophysics*, 72(3) (2007), pp. R37–R49.
- [29] J. E. AARNES, V. KIPPE, K.-A. LIE, AND A. B. RUSTAD, *Modelling of multiscale structures in flow simulations for petroleum reservoirs*, in *Geometric Modelling, Numerical Simulation, and Optimization*, pages 307–360, Springer, (2007).
- [30] G. LITJENS, T. KOOL, B. E. BEJNORDI, A. A. A. SETIO, F. CIOMPI, M. GHAFORIAN, J. A. VAN DER LAAK, B. VAN GINNEKEN, AND C. I. SÁNCHEZ, *A survey on deep learning in medical image analysis*, *Med. Image Anal.*, 42 (2017), pp. 60–88.
- [31] O. RONNEBERGER, P. FISCHER, AND T. BROX, *U-net: Convolutional networks for biomedical image segmentation*, in *Proc. Med. Image Comput. Comput. Assist. Intervent.*, pages 234–241, Springer, (2015).
- [32] V. BADRINARAYANAN, A. KENDALL, AND R. CIPOLLA, *Segnet: A deep convolutional encoder-decoder architecture for image segmentation*, *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(12) (2017), pp. 2481–2495.
- [33] L.-C. CHEN, Y. ZHU, G. PAPANDREOU, F. SCHROFF, AND H. ADAM, *Encoder-decoder with atrous separable convolution for semantic image segmentation*, in *Proc. 15th Eur. Conf. Comput. Vis. (ECCV)*, pages 801–818, (2018).
- [34] F. CHOLLET, *Xception: Deep learning with depthwise separable convolutions*, in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 1251–1258, (2017).
- [35] L.-C. CHEN, G. PAPANDREOU, F. SCHROFF, AND H. ADAM, *Rethinking atrous convolution for semantic image segmentation*, arXiv preprint, arXiv:1706.05587, (2017).
- [36] R. GIRSHICK, *Fast r-cnn*, in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, pages 1440–1448, (2015).
- [37] C. H. SUDRE, W. LI, T. VERCAUTEREN, S. OURSELIN, AND M. J. CARDOSO, *Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations*, in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pages 240–248, Springer, (2017).
- [38] A. PASZKE, S. GROSS, S. CHINTALA, G. CHANAN, E. YANG, Z. DEVITO, Z. LIN, A. DESMAISON, L. ANTIGA, AND A. LERER, *Automatic differentiation in PyTorch*, in *NIPS Autodiff Workshop*, (2017).

- [39] D. P. KINGMA AND J. BA, *Adam: A method for stochastic optimization*, arXiv preprint, arXiv:1412.6980, (2014).
- [40] I. SUTSKEVER, J. MARTENS, G. DAHL, AND G. HINTON, *On the importance of initialization and momentum in deep learning*, in Proc. Int. Conf. Mach. Learn. (ICML), pages 1139–1147, (2013).
- [41] H. LIU, A. J. VALOCCHI, C. WERTH, Q. KANG, AND M. OOSTROM, *Pore-scale simulation of liquid CO₂ displacement of water using a two-phase lattice boltzmann model*, Adv. Water Resour., 73 (2014), pp. 144–158.
- [42] J. T. SPRINGENBERG, A. DOSOVITSKIY, T. BROX, AND M. RIEDMILLER, *Striving for simplicity: The all convolutional net*, arXiv preprint, arXiv:1412.6806, (2014).
- [43] H. NOH, S. HONG, AND B. HAN, *Learning deconvolution network for semantic segmentation*, in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), pages 1520–1528, (2015).
- [44] R. R. SELVARAJU, M. COGSWELL, A. DAS, R. VEDANTAM, D. PARIKH, AND D. BATRA, *Grad-cam: Visual explanations from deep networks via gradient-based localization*, in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), pages 618–626, (2017).