

OPEN

# Predicting Effective Diffusivity of Porous Media from Images by Deep Learning

Haiyi Wu<sup>1</sup>, Wen-Zhen Fang<sup>1,2</sup>, Qinjun Kang<sup>3\*</sup>, Wen-Quan Tao<sup>2\*</sup> & Rui Qiao<sup>1\*</sup>

We report the application of machine learning methods for predicting the effective diffusivity ( $D_e$ ) of two-dimensional porous media from images of their structures. Pore structures are built using reconstruction methods and represented as images, and their effective diffusivity is computed by lattice Boltzmann (LBM) simulations. The datasets thus generated are used to train convolutional neural network (CNN) models and evaluate their performance. The trained model predicts the effective diffusivity of porous structures with computational cost orders of magnitude lower than LBM simulations. The optimized model performs well on porous media with realistic topology, large variation of porosity (0.28–0.98), and effective diffusivity spanning more than one order of magnitude ( $0.1 \lesssim D_e < 1$ ), e.g., >95% of predicted  $D_e$  have truncated relative error of <10% when the true  $D_e$  is larger than 0.2. The CNN model provides better prediction than the empirical Bruggeman equation, especially for porous structure with small diffusivity. The relative error of CNN predictions, however, is rather high for structures with  $D_e < 0.1$ . To address this issue, the porosity of porous structures is encoded directly into the neural network but the performance is enhanced marginally. Further improvement, i.e., 70% of the CNN predictions for structures with true  $D_e < 0.1$  have relative error <30%, is achieved by removing trapped regions and dead-end pathways using a simple algorithm. These results suggest that deep learning augmented by field knowledge can be a powerful technique for predicting the transport properties of porous media. Directions for future research of machine learning in porous media are discussed based on detailed analysis of the performance of CNN models in the present work.

Predicting the transport properties of porous materials from their structure is important in numerical simulations of a wide range of engineering problems, e.g., extraction of nature gas from shale reservoirs<sup>1–4</sup>, drying of building materials<sup>5</sup>, and charging/discharging of Li-air batteries. The transport properties of interest include effective diffusivity, permeability, thermal conductivity, among others<sup>1</sup>. A classical approach for calculating these effective transport properties is the pore scale simulations, in which the governing equations for the related transport phenomena are solved within the porous media<sup>2,6,7</sup>. While this approach can be highly accurate, its computational cost is significant for porous media with large dimension and/or small pore sizes. In fact, for porous media that undergo long operation during which their structure evolves (e.g., the pores in electrodes of Li-air batteries are gradually clogged during discharging), this approach can be prohibitively expensive because their effective transport properties may need to be evaluated for millions of times. As such, the effective transport properties of porous media are often computed using empirical correlations or effective medium theories with their structure information (e.g., porosity) as input. Such an approach needs little computational cost and can be very accurate for some specific (often idealized) classes of porous media. However, because typically only a few structure parameters of the porous media are used as input in this approach, its prediction for complex porous media often lacks specificity and can be inaccurate. Indeed, it remains a great challenge to develop methods for predicting the effective transport properties of porous media that require low computational cost but offer high accuracy for diverse porous structures.

Machine learning can potentially be an effective approach for tackling the above challenge. Deep neural networks have demonstrated good predictive power when their input and output have important correlation with each other. Furthermore, image-based learning has been shown to be able to extract important physical features

<sup>1</sup>Department of Mechanical Engineering, Virginia Tech, Blacksburg, VA, 24061, USA. <sup>2</sup>Key Laboratory of Thermo-Fluid Science and Engineering, MOE, Xi'an Jiaotong University, Xi'an, 710049, China. <sup>3</sup>Earth and Environmental Sciences Division, Los Alamos National Laboratory, Los Alamos, NM, 87545, USA. \*email: [qkang@lanl.gov](mailto:qkang@lanl.gov); [wqtao@mail.xjtu.edu.cn](mailto:wqtao@mail.xjtu.edu.cn); [ruiqiao@vt.edu](mailto:ruiqiao@vt.edu)

from images<sup>8,9</sup>. Because the effective transport properties (in particular, the effective diffusivity) of porous media is largely determined by their structure which can be conveniently represented using their binary images, conceivably, one can develop a surrogate deep learning model to extract key geometrical features from images of porous media and predict their transport properties. In terms of implementation, the application of deep neural network typically requires a training dataset, which can be generated numerically or experimentally for porous media. Next, a training model is constructed and trained using the dataset. Finally, the trained model can be used to predict the effective transport properties of new porous structures without repeating the training process. This general strategy resembles the investigation of image classification, where images are taken as the input and trained deep learning models predict the classification label (e.g. “dog” or “cat”) of images by identifying their key features<sup>10–12</sup>. As demonstrated in the studies of image classification, typically, a trained model can be used to make predictions with low computational cost. Therefore, a deep learning model, if well-constructed and trained, can potentially predict the effective transport properties of porous media both accurately and efficiently.

Of the many deep neural network models, convolutional neural network (CNN)<sup>13</sup> is commonly applied to analyze visual imagery and has achieved much success in image classification. Recently, CNN has also been adopted to study the effective properties of complex materials and showed much potential for efficient and accurate prediction of a material's effective properties from its structure (e.g., presented in the form of images). For example, researchers have used CNN to predict the effective permeability and stiffness of materials from their microstructures<sup>8,9,14–16</sup>. In particular, three dimensional CNN has shown to capture the nonlinear mapping between material microstructure and its effective stiffness<sup>8</sup>. Study of the prediction of permeability from images of porous media using CNN has provided useful insights in understanding the correlation between geometric features and transport properties<sup>14,16</sup>. The features of connectivity between nearby pixels in the image of a porous structure can be extracted by performing convolution with many possible cross shape templates. It has been shown that the features thus extracted can be used to make better prediction of permeability than using the geometric measurements (Minkowski functionals)<sup>16</sup>. Furthermore, it has been pointed out that deep learning approach can be further improved by incorporating physical parameters of porous media that are known to affect effective permeability<sup>14</sup>. Nevertheless, the porous structure in previous works is relatively well-defined and of limited geometrical variability, and some of the challenging topologies (i.e. trapped and dead-end pathways) that are commonly seen in porous media were not included. Consequently, it is still not clear whether the CNN model can accurately predict the effective transport properties of practical porous media with diverse geometries and challenging topologies. Delineating the impact of such diversity and complexity of porous media on the performance of CNN models and building more sophisticated deep learning models to deal with them are important for the practical application of CNN models in porous media research.

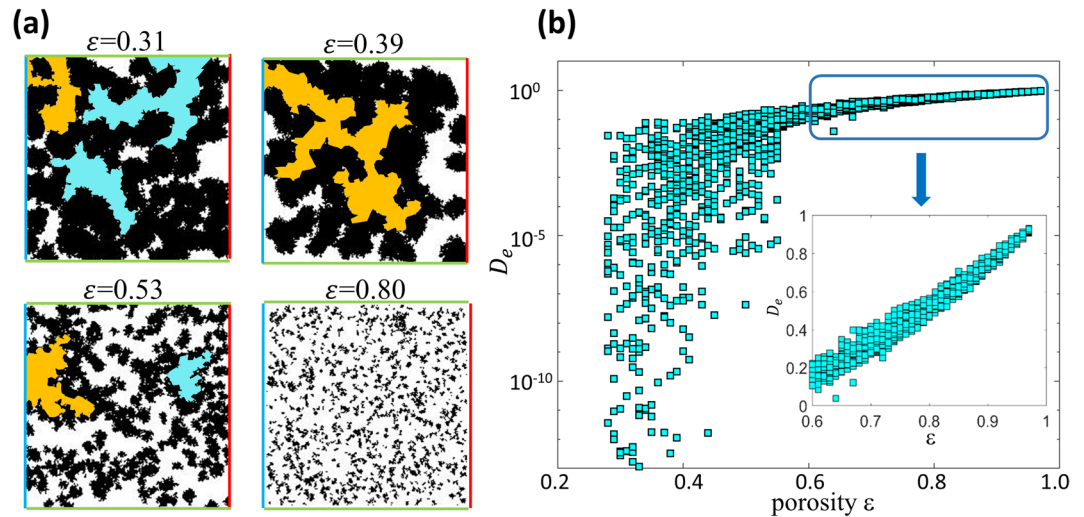
In this work, we establish a computational framework to predict the effective diffusivity of porous media from their images using CNN. We focus on the effective diffusivity because of its importance in practical applications of porous media. The proposed framework is demonstrated in porous media with a wide range of porosity (0.28–0.98) and diverse/complex structures (e.g., tortuous diffusion pathways and trapped and dead-end regions). Different filter sizes and convolutional layers are tested during the cross-validation step in order to select the best hyperparameters for CNN. Physical parameter (the porosity) is also combined with the extracted features by CNN with different weights to generate better predictions. The CNN can predict the effective diffusivity of most of the porous media with less than 10% truncated relative error. Nevertheless, the prediction error, especially the relative error, increases as the true diffusivity of the porous media becomes smaller (especially if  $D_e < 0.1$ ). The large error is attributed to the complex transport behavior in porous media with low diffusivity, where the porous structure can be highly heterogeneous with highly tortuous diffusion pathways and numerous trapped areas or dead-end paths.

The rest of the manuscript is structured as follows: In Section II, we detail the computational framework and methods used for predicting the effective diffusivity of porous media by deep learning models. In Section III, the performance of the developed model is examined and discussed. We first quantify the performance of CNN and compare the CNN prediction with the empirical Bruggeman equation predictions. Then, we attempt to improve the regular CNN prediction by leveraging field knowledge in the development and application of CNN models. Finally, conclusions and discussion on the future research directions of machine learning for porous media research are presented in Sec. IV.

## Computational Framework

In this section, we discuss the deep learning model for predicting the effective diffusivity of porous media with diverse and challenging structures. In part A, we present the methods for generating the dataset for deep learning. In part B, we present the architecture of our CNN model and summarize the computational framework of using CNN to predict the effective diffusivity of porous media from their images. Finally, the methods for training and testing of the CNN models are presented in part C.

**Generation of datasets.** The dataset for training, validating, and testing our deep learning models includes the structure of porous media (in the form of images) and their corresponding effective diffusivity. Without losing generality, we will focus on two-dimensional (2D) porous media with a square shape. The microstructures of 2D porous material are generated using the quartet structure generation set (QSGS) method, a popular method in the porous media field<sup>17</sup>. Detailed descriptions of this method can be found in the literature and we only outline its key steps<sup>17</sup>: (1) The computational domain is partitioned into square cells. (2) Solid “seeds” are randomly distributed in the domain based on a distribution probability,  $c_d$ , which is smaller than the target porosity of the porous media. This is accomplished by assigning a random number to each cell and the cells whose assigned random number is less than  $c_d$  are selected as the “seeds”. (3) Grow the “seeds” to their neighboring cells based on the directional growth probability,  $P_i$ . To this end, a random number is assigned to each of the neighboring cell of



**Figure 1.** (a) Representative images of the two-dimensional porous media generated for the deep learning model. The white and black colors denote the pore space and solid phase, respectively. The inlet (outlet) for the mass diffusion through the porous media are marked using blue (red) lines. The green lines at the top and bottom side of the porous media denote the periodic boundary in the LBM simulation. The orange areas represent the dead-end pathways (i.e., the pathways that are connected only to the inlet or outlet of the porous structure) and the cyan areas denote the trapped pore space (i.e., isolated pore space that is not connected to the porous structure's inlet and outlet). (b) The distribution of the effective diffusivity of porous media generated in this work.

a solid seed. If the random number of a neighboring cell is less than  $P_r$ , it will become part of the growing solids. (4) Repeat steps (2) and (3) until the target porosity is reached in the domain. Together, the above steps produce binary images of porous media. In these images, individual pores or grains are fully resolved, and each pixel is either a pore or solid node and is denoted with a binary value of 0 (pore space) or 1 (solid phase).

Using the above method, 2D porous media are generated within a  $200 \times 200$  (pixel) area. This size of the porous structure is chosen so that (1) the microstructures are large enough to capture the range of the topologies and transport behavior occurring in realistic porous media and (2) the porous structures are small enough so that a large number of porous structures and their effective diffusivity can be obtained at a reasonable computational cost. To ensure that the dataset includes a wide variety of porous structures, 1960 samples are generated with porosity ( $\varepsilon$ ) of 0.28, 0.29, ..., 0.98. For each porosity, 28 samples are generated. Figure 1a shows the representative images of the porous samples generated. A wide variety of structures featuring tortuous transport pathways, trapped regions, and dead-end pores are obtained in samples with porosity smaller than 0.5.

We next compute the effective diffusivity of the porous structures generated above. The molecular transport within the porous structure is assumed to follow Fick's Law with a constant diffusion coefficient  $\tilde{D}_0$ . Therefore, the molecular diffusion within the porous structure obeys the Laplace equation with a zero-flux boundary condition on pore surfaces. To compute the effective diffusivity of each porous structure, a uniform concentration difference ( $\Delta C$ ) is imposed between the left and right boundaries of each porous structure and the periodic boundary condition is imposed on the top and bottom boundaries, respectively. The Laplace equation is solved using the lattice Boltzmann method (LBM). Specifically, a two-dimensional, nine-velocity (D2Q9) LB model is adopted to simulate the diffusion process inside the porous structures. Different from traditional numerical methods by discretizing the Laplace equation in the pore space, LBM solves the evolution equation of the concentration distribution functions

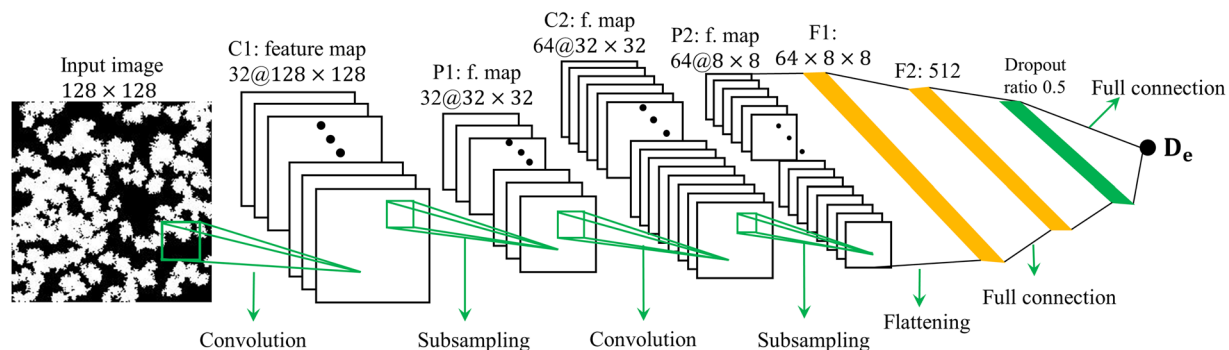
$$g_i(x + e_i \Delta t, t + \Delta t) - g_i(x, t) = -\frac{1}{\tau} (g_i(x, t) - g_i^{eq}(x, t)) \quad (1)$$

where  $g_i$  is the concentration distribution function at the space location  $x$  and time  $t$  along  $i$  direction;  $\tau$  is the relaxation time coefficient;  $g_i^{eq} = \omega_i C$  is the corresponding equilibrium concentration distribution function, where  $C = \sum g_i$  is the macroscopic local concentration,  $\omega_i$  is weight parameter, and  $\omega_0 = 4/9$ ,  $\omega_{1-4} = 1/9$ ,  $\omega_{5-8} = 1/36$ . In Eq. (1),  $e_i$  is the discrete velocity given by

$$e_i = \begin{bmatrix} 0 & 1 & 0 & -1 & 0 & 1 & -1 & -1 & 1 \\ 0 & 1 & 1 & 0 & -1 & 1 & 1 & -1 & -1 \end{bmatrix} \quad (2)$$

The relation between the intrinsic gas diffusion coefficient and the relaxation time coefficient is

$$\tilde{D}_0 = \frac{1}{3} (\tau - 0.5) \frac{\Delta x^2}{\Delta t} \quad (3)$$



**Figure 2.** The architecture of our regular convolutional neural network (CNN) for predicting the effective diffusivity of 2D porous media from their images.

After the diffusion within the porous structure reaches a steady state, the cross-section averaged diffusive flux through the structure is obtained by<sup>18</sup>

$$J_x = \frac{\int_0^{L_y} \sum e_{i,x} g_i \frac{\tau - 0.5}{\tau} dy}{L_y} \quad (4)$$

where  $L_y$  is the domain size in the direction normal to the overall diffusion flux. The dimensional effective diffusivity of the porous structure is then determined using Fick's law, i.e.,  $\tilde{D}_e = J_x L_x / \Delta C$  ( $L_x$  is the domain size in the direction of the overall diffusion flux). Because  $\tilde{D}_e$  is linearly proportional to the molecule diffusion coefficient in the pore space ( $\tilde{D}_0$ ), the dimensionless effective diffusivity  $D_e = \tilde{D}_e / \tilde{D}_0$  is used hereafter. The effective diffusivity of the porous structures generated above spans  $\sim 10^{-10}$  to 1.0 and is shown as a function of the porosity of the porous structures in Fig. 1b.

The data generated above (porous structures and their  $D_e$ ) are randomly divided into the training dataset (60% of the whole dataset), validation dataset (11.4%) and testing dataset (28.6%). The training dataset is used to optimize the parameters of the CNN model so that the model can describe the training dataset as accurately as possible. The validation dataset is used to select hyperparameters and avoid overfitting the CNN model. The testing dataset is used to evaluate the predictive performance of the trained CNN model.

**Convolutional neural network for predicting effective diffusivity.** The basic concepts of classical and convolutional neural networks, along with terminologies including hyperparameters and learnable parameters, are reviewed in the Supporting Information. These models, especially the CNN, have been used for image classification with great success and are being explored for predicting the effective permeability of porous media. Inspired by these works, we adapt the CNN model for predicting the effective diffusivity of porous media from their images<sup>11,19,20</sup>, and the architecture of our model is shown in Fig. 2. The binary image of a porous structure, in which a pixel with a value of 1(0) corresponds to the solid phase (pore space), is the input of the CNN. We note that the binary nature of the input image is consistent with the format of the computationally generated porous media in this work (see Section II.A) and is not a result of the binarization of grayscale images as have been done in many image recognition studies. Because the desired pixel size of images in CNN is  $2^m$  ( $m$  is an integer), the images of porous structures made in Section A are downsampled to  $128 \times 128$  pixels using kernel<sup>21</sup> before feeding to the CNN. Below we outline the different layers in the CNN. We focus on identifying the data flow as well as learnable parameters and their dimensions, but omit numerical implementation details as they are widely available in the literature.

In this work, we adopt the CNN model that has similar architecture of the AlexNet<sup>11</sup>. Our CNN model has  $M$  pairs of convolutional and pooling layers and  $P$  fully-connected layers ( $M$  and  $P$  are both set to 2 in Fig. 2 for illustration purpose). When CNN is used in image-related studies as in this work, the output volume of a convolutional or pooling layer is typically termed *feature maps* as the purpose of these layers is to extract features from their input volume<sup>11,20</sup>. For simplicity, the width of any input/output volume of convolutional/pooling layer is always equal to its height in this work. The number of slices of an output volume of a convolutional layer ( $d_o$ ) is its depth. To obtain the  $\delta$ -th slice of the feature map,  $F_{\text{map}}^{\delta}_{\text{conv}}$ , a filter is slid over every width and height position of the input volume and the result is the output for neuron at each position. By writing  $F_{\text{map}}^{\delta}_{\text{conv}}$  as a  $a_o \times a_o$  order-2 tensor ( $a_o$  is the width/height of the feature map), this operation can be written as

$$F_{\text{map}}^{\delta}_{\text{conv}} = \text{ReLU}(W^{\delta} \otimes X_i + B(\delta) \cdot J) \quad (5)$$

where ReLU is the activation function adopted for the neurons in this work<sup>22</sup>.  $X_i$  is the input volume of the convolutional layer (for layers C1 and C2 in Fig. 2,  $X_i$  are the 2D binary image and the feature map generated by the first pooling layer, respectively).  $X_i$  is a  $a_i \times a_i \times d_i$  order-3 tensor, where  $a_i$  and  $d_i$  is the width and depth of the convolutional layer's input volume, respectively.  $W^{\delta}$  is the kernel of the  $\delta$ -th filter of the convolutional layer. It is a



$s_\delta \times s_\delta \times d_i$  order-3 tensor, where  $s_\delta$  is the filter's spatial extent in the width direction.  $W^\delta \otimes X_i$  denotes the 2D-convolution of the filter kernel with the input volume, and its expression can be found in numerous textbooks<sup>23</sup>.  $B(\delta)$  is the bias for the  $\delta$ -th filter and  $J$  is a  $a_o \times a_o$  order-2 tensor with all elements equal to 1.0. In Eq. (5),  $W^\delta$  and  $B(\delta)$  ( $\delta = 1, 2, \dots, d_o$ ) must be “learned” during the training of CNN and they include a total of  $(s_\delta^2 \times d_i + 1) \times d_o$  learnable parameters. On the other hand,  $a_o, s_\delta, d_o$ , and the number of convolutional layers  $M$  are hyperparameters to be specified when building the CNN. Generally, the width and depth of the feature map must be large enough to ensure both short- and long-range features can be extracted but small enough to lower computational cost and suppress overfitting. In this work, the width of the output feature maps of a convolutional layer is the same as its input feature map or image. The depth of feature maps generated by the first and second convolutional layers in our CNN is taken as 32 and 64, respectively.

The feature map generated by each convolutional layer is fed into a pooling layer (see Fig. 2) to obtain a new feature map with the same depth but reduced width. The spatial span, stride, and type of filter used in the pooling layers are also hyperparameters that can be optimized through cross-validation. Here, we use the max pooling with a filter size of  $4 \times 4$  and a stride of 4 to downsample the input feature maps.

The feature map generated by the last pooling layer is flattened into a vector ( $X_{\text{in,fc1}}$ ; length:  $n_{\text{in,fc1}}$ ). This vector is fed to the first fully connected layer with  $n_{\text{fc1}}$  neurons (see Fig. 2), which produces  $F_{\text{map,fc1}}$  through Eq. (S2) (supporting information) using a  $n_{\text{fc1}} \times n_{\text{in,fc1}}$  matrix  $W_{\text{fc1}}$  as learnable weights and a vector  $B_{\text{fc1}}$  (length:  $n_{\text{fc1}}$ ) as learnable biases.  $n_{\text{fc1}}$ , a hyperparameter, is taken as 512 in this work. To avoid overfitting, a dropout layer (dropout ratio = 0.5) is applied before the second fully connected layer. The output of the dropout layer is passed to the second fully connected layer to make the final prediction via

$$D_e^{\text{CNN}} = W_{\text{fc2}} X_{\text{o,drop}} + B_{\text{fc2}} \quad (6)$$

where  $D_e^{\text{CNN}}$  is the effective diffusivity predicted by CNN,  $X_{\text{o,drop}}$  is output of the dropout layer, and  $W_{\text{fc2}}$  and  $B_{\text{fc2}}$  are the learnable weight and bias of the second fully connected layer, respectively.

In addition to the CNN above, we also test the deep residual network, Resnet50<sup>24</sup>, a more recent scheme of CNN (see Supporting Information). Because Resnet50 only performs slightly better than the above CNN but its structure is considerably more involved, below we present only the results based on the CNN outlined in Fig. 2. The implementation, results, and comparison of the Resnet50 with the above CNN model are summarized in the Supporting Information.

**Using field knowledge to augment CNN model.** A great advantage of CNN is that physical properties can be encoded into the CNN architecture to improve its final prediction. Because the effective diffusivity of a porous structure is strongly correlated with its porosity, following previous work<sup>14</sup>, we also combine the porosity with the flattened feature map of the last pooling layer to form the first fully connected layer in CNN. Specifically, the porosity of each porous structure is added as input to the first fully connected layer of the conventional CNN model with a fixed weight (a variety of weights have been tested with similar results, and a value of 10/4096 is used in the final model here).

Another method that can potentially improve the performance of CNN model is to preprocess the image of porous structures. As shown in Fig. 1a, for structures with low porosity (e.g.,  $\epsilon < 0.5$ ), there exist many trapped areas and/or dead-end paths. These trapped regions are typically larger than the filter size used in the CNN model and thus can be difficult to extract directly using the CNN model. Moreover, the trapped regions are prone to provide faulty information to the feature map and induce noise during the learning process. In this work, before feeding to the CNN, images of porous structures are processed using the 8-connected component analysis<sup>25,26</sup> to eliminate all the trapped pore spaces (see Section III.C.2 for details).

**Summary.** The computation framework for predicting the effective diffusivity of porous structures from their images using CNN models can be summarized as four steps (see Fig. S2):

- Data generation:** Porous media with a wide variety of porosity and pore topologies are generated and their true effective diffusivity  $D_e$  (ground truth) is obtained using LBM simulations.
- Augmenting CNN by field knowledge:** Trapped pore spaces are removed in the preprocessing step and physical properties of porous structure (porosity) are combined with feature maps built by the convolutional and pooling layers to serve as the input to the fully-connected layers in CNN.
- Parameterizing CNN.** CNN's hyperparameters and learnable parameters are established through training and cross-validation using the training and validation datasets.
- Deploying CNN.** Using the trained model in step (c) to predict effective  $D_e$  of porous media using their images as input.

**Parameterization of convolutional neural networks.** After the datasets are generated and CNN model is set up, we first need to select all hyperparameters in the CNN model. A CNN model has many hyperparameters such as the choice of the activation function, size of the feature map produced by each convolutional/pooling layer, etc. Although it is desirable to select all parameters through cross-validation, doing so requires prohibitively high computational cost. Therefore, many of these parameters are selected empirically. While this is not a rigorous approach, researches in image classification show that the performance of CNN models is often insensitive to the selection of many hyperparameters. These researches suggest that the number of convolutional layers and the spatial extent (size) of the filters used in these layers usually play the most important role in CNN. Hence, in this

Number of convolutional layers ( $M$ )	Filter size ( $s \times s$ )		
	$3 \times 3$	$5 \times 5$	$7 \times 7$
2 layers	$1.38 \times 10^{-4}$	$8.76 \times 10^{-5}$	$7.51 \times 10^{-5}$
3 layers	$4.44 \times 10^{-4}$	$1.09 \times 10^{-4}$	$1.13 \times 10^{-4}$
4 layers	$2.14 \times 10^{-3}$	$6.17 \times 10^{-4}$	—

**Table 1.** The mean square error of CNN models with different hyperparameters obtained during cross-validation.

work, CNN models with different number of convolutional layers ( $M = 2, 3, 4$ ) and different filter sizes in the convolutional layer ( $s \times s = 3 \times 3, 5 \times 5, 7 \times 7$ ) are evaluated. Other hyperparameters used in the model can be found in Section II.B.2.

We next optimize the weights and biases for all filters in each convolutional and fully connected layer by “learning” the training dataset. The proposed CNN model is implemented in the machine learning framework Tensorflow<sup>27</sup>. The loss function  $L$ , which quantifies how well the CNN reproduces the training dataset, is defined as the average mean square error over the entire training dataset

$$L = 1/N \sum_{i=1}^N (D_{e,i}^{LBM} - D_{e,i}^{CNN})^2 \quad (7)$$

The weights of all filters are initialized using a truncated normal distribution with a standard deviation 0.1. All biases are set to 0.1 initially. The training model is converged through minimizing the loss function by the Adam Optimizer<sup>28</sup> with a learning rate  $\gamma = 10^{-4}$ . The training model stops at 1600 epochs. The optimized weights and biases for all learnable filters at the last epoch are saved in the trained model. The saved model will then be restored for evaluating the testing dataset without redoing the training process.

## Results and Discussions

In this section, we first examine how key hyperparameters affect the performance of CNN models using cross-validation. We next present the performance of the regular CNN and compare it with that of a classical empirical model. Then, the performance of field knowledge-informed CNN on a wide range of porous structures is evaluated.

**Cross-validation for hyperparameters selection.** A trained CNN model often can fit the training dataset used to parameterize it well, but may not perform well on other datasets due to overfitting. To reduce overfitting of the trained model, one usually optimizes hyperparameters of the CNN model through cross-validation. Here, we systematically vary the number of convolutional layers and the filter size in the CNN and cross-validate the trained model using the validation dataset generated in Section II.A. Table 1 summarizes the mean square error (MSE) of the effective diffusivity predicted by the CNN model with relative to the true value. We observe that, for a fixed filter size, MSE is smaller for CNN models with fewer convolutional layers. For CNN models with the same number of convolutional layers, the MSE is highest when the filter size is  $3 \times 3$ , likely because the narrow filters cannot capture some important features spanning moderate to large number of pixels. The MSE is comparable for models with a filter size of  $5 \times 5$  and  $7 \times 7$ . Because the computational cost is higher when filters with  $7 \times 7$  size are used, we adopt the CNN models with 2 convolutional layers and a filter size of  $5 \times 5$  (see Fig. 2).

**Performance of the regular CNN model.** The performance of the regular CNN model, in which field knowledge is not encoded, is evaluated using the testing dataset. For each porous structure described by a  $128 \times 128$  binary image, its effective diffusivity is computed by the trained model using  $\sim 1.5 \times 10^8$  FLOPs of calculations. This cost is close to the cost of multiplying two  $400 \times 400$  matrices and requires  $\sim 4$  ms on a laptop (Intel CPU 6600, 3.3 GHz, 8 GB memory, single core). As a comparison, computing the effective diffusivity using LBM requires  $\sim 1$  hour on the same computer. Therefore, the computational cost of the CNN model is roughly six orders of magnitude smaller than the LBM simulations.

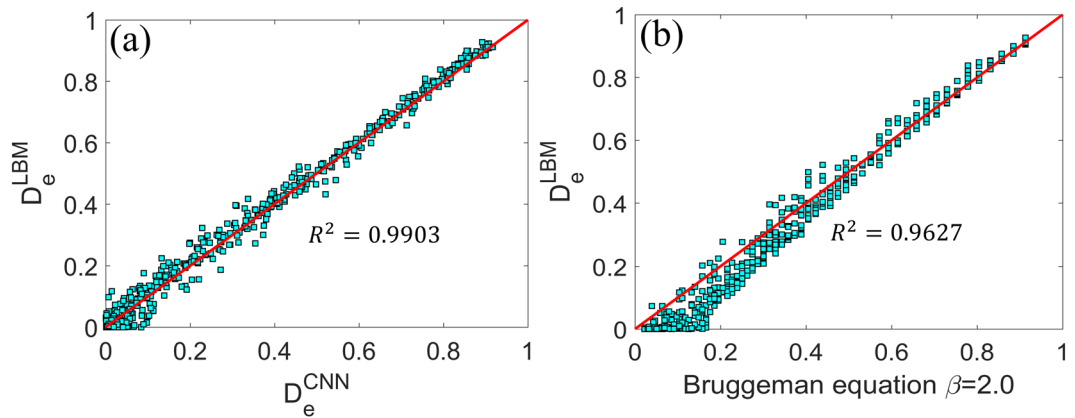
The mean square error (MSE) of the effective diffusivity of porous structures in the entire testing dataset is found to be  $8.64 \times 10^{-4}$  (see Table 2). Since the porous structures feature diverse pore shape/topology and their effective diffusivity spans  $\sim 10^{-10}$  to 1.0, such a small MSE suggests that the CNN model performs well for the complex porous structure examined here. Figure 3a further compares the effective diffusivity predicted by the CNN model and computed by LBM codes, and we observe that the CNN model's overall performance is again very good. It is worthwhile to compare the performance of the CNN model with that of empirical correlations. The Bruggeman equation has been widely adopted in predicting the effective diffusivity of porous structure and is given by  $D_e = \varepsilon/\tau$  ( $\varepsilon$ : porosity,  $\tau$ : tortuosity). The tortuosity of porous media is commonly modeled using  $\tau = \varepsilon^{-\beta}$ . Hence, the Bruggeman equation predicts

$$D_e = \varepsilon^{\beta+1} \quad (8)$$

The parameter  $\beta$  depends on the structure of porous media, e.g., the connectivity of pores.  $\beta$  is equal to 0.5 for porous media made of packed spheres and is otherwise an empirical parameter that is strongly correlated with the tortuosity of the porous media<sup>1,29</sup>. Here,  $\beta$  is taken as 2.0 because it describes the scaling of  $D_e$  at large porosity

	Regular CNN trained using the loss function Equ. 7	Regular CNN trained using the loss function Equ. 10	Porosity- informed CNN	Regular CNN with preprocessed image as input
Mean square error	$8.64 \times 10^{-4}$	$7.01 \times 10^{-4}$	$7.64 \times 10^{-4}$	$6.92 \times 10^{-4}$
Mean truncated relative error	68.8%	41.7%	59.2%	29.7%
$R^2$	0.9903	0.9921	0.9914	0.9912

**Table 2.** The mean square error and mean truncated relative error of the various CNN models.



**Figure 3.** Effective diffusivity predicted by the CNN model (a) and the empirical Bruggeman equation (b).

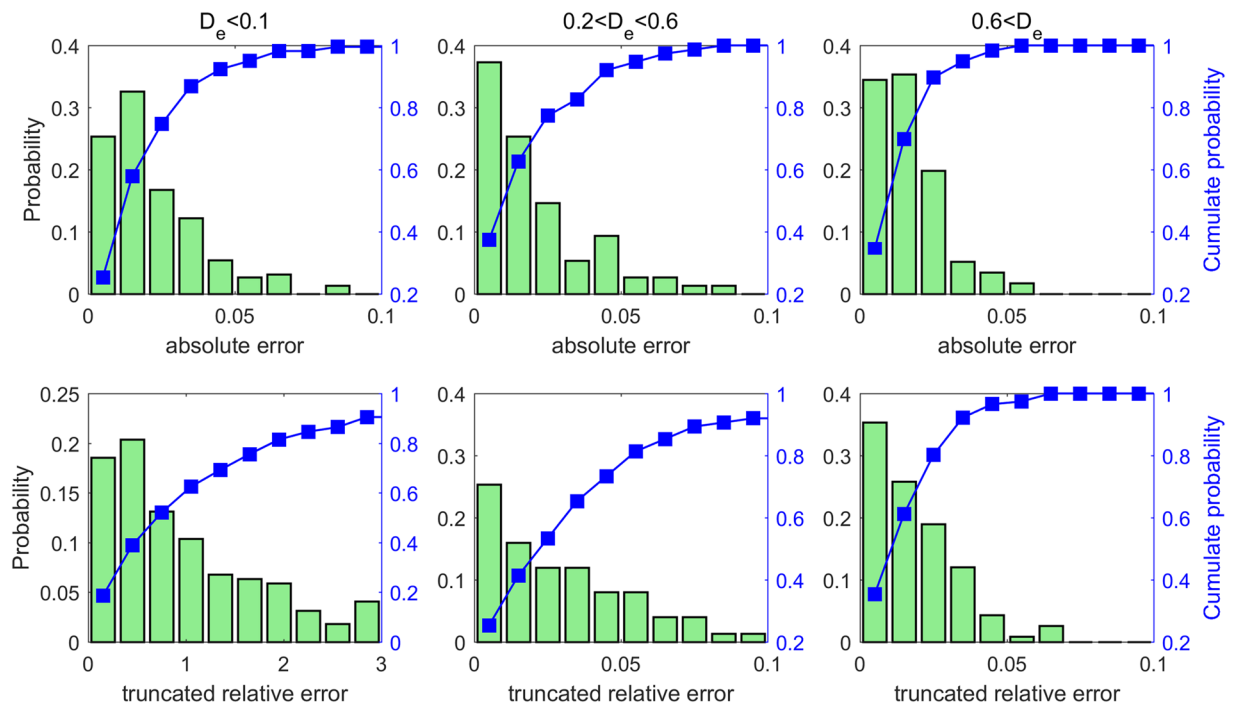
relative well. Figure 3a compares the prediction of the Bruggeman equation with the LBM prediction. When the true  $D_e$  is larger than  $\sim 0.6$ , the Bruggeman equation performs as well as the CNN model. However, as the true  $D_e$  becomes smaller, which is mostly caused by the emergence of more tortuous transport pathway in the porous media, the Bruggeman equation systematically overpredicts the effective diffusivity and performs worse than the CNN model. Therefore, the CNN model can better capture the transport properties of porous media with complex structure than the Bruggeman equation.

We next quantify the predictive power of the CNN model *posteriori* systematically. Because the performance of the CNN model depends on the effective diffusivity of porous structure (see Fig. 4a), the porous structures in the testing dataset are divided into three groups: those with  $D_e < 0.1$ , those with  $0.2 < D_e < 0.6$  and those with  $D_e > 0.6$ . The top row of Fig. 4 shows the distribution of the absolute error of  $D_e$  predicted by the CNN model. In each group, over 95% of predictions have absolute error smaller than 0.1. Furthermore, the distribution of the absolute error is similar for all groups, which suggests that, during the training of the CNN model, the characteristics of all three groups are “learned” by the model.

Another way to characterize the CNN model’s predictive power is to study the relative error of its predictions. The standard calculation of relative errors, however, is not necessarily a good approach for the porous media studied here. Specifically, because many of our porous structures have extremely small  $D_e$  (e.g.,  $10^{-10}$ , see Fig. 1), a minute error in their  $D_e$  can correspond to an enormous relative error even though such minute error has little practical impact on the prediction of transport in the porous media. While there is no unique way to address this issue, we define a truncated relative error, TRE, as

$$\text{TRE} = \begin{cases} \frac{|D_e^{\text{LBM}} - D_e^{\text{CNN}}|}{D_e^{\text{LBM}}} & (\text{if } D_e^{\text{LBM}} > D_e^{\text{thsd}}) \\ \frac{|D_e^{\text{LBM}} - D_e^{\text{CNN}}|}{D_e^{\text{thsd}}} & (\text{if } D_e^{\text{LBM}} < D_e^{\text{thsd}}) \end{cases} \quad (9)$$

where  $D_e^{\text{LBM}}$  and  $D_e^{\text{CNN}}$  are the predictions by the LBM simulations (taken as ground truth here) and the CNN model, respectively. In this definition, only inaccuracy comparable to or larger than a threshold  $D_e^{\text{thsd}}$  (taken as 0.01 hereafter to be specific) is thought to affect the practical application of the  $D_e$  computed for the porous structure (e.g., in the prediction of pore clogging during the discharging of Li-air electrodes). Analysis of the predicted  $D_e$  shows that the mean truncated relative error is 68.8% for the entire testing dataset (see Table 2). The bottom row of Fig. 4 further shows the distribution of the truncated relative error for the three groups of porous structure in the testing dataset. We observe that the truncated relative error is the larger in samples with smaller  $D_e$ : for samples with  $0.2 < D_e < 0.6$ ,  $\sim 95\%$  of the CNN predictions have a truncated relative error less than 10%; for samples with  $D_e < 0.1$ ,  $\sim 60\%$  of the CNN predictions have a truncated relative error less than 100%. These observations are consistent with the above observation that the magnitude of the absolute error similar among samples with different  $D_e$ .



**Figure 4.** Distribution of the absolute error (top panels) and truncated relative error (lower panels) of the predictions of the CNN model for the porous structures in the testing dataset with  $D_e < 0.1$ ,  $0.2 < D_e < 0.6$  and  $D_e > 0.6$ . The CNN model is trained using the loss function based on the mean square error.

The larger relative error in samples with small  $D_e$  is expected because the regular CNN model is trained to minimize the MSE (i.e., a measure of the absolute rather than relative error) over the training set. To explore whether the CNN model can be tailored to give smaller relative errors (especially in samples with  $D_e < 0.1$ ), we define a new loss function for training the CNN model as

$$L = 1/N \sum_{i=1}^N (TRE_i) \quad (10)$$

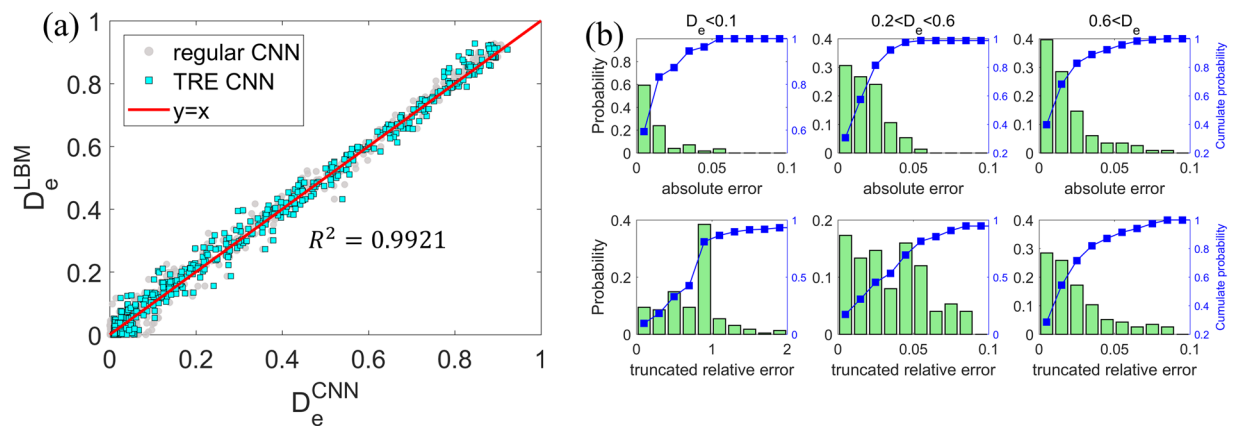
where  $N$  is the number of porous samples in the training dataset,  $TRE_i$  is the truncated relative error for each training data, which is defined in Eq. (9). Training of the CNN model based on this loss function is able to converge. The CNN thus trained gives a truncated relative error of 41.7% when applied to the samples in the testing dataset, which is moderately better than that of the regular CNN model (68.6% see Table 2). Interestingly, the MSE of the new model ( $7.01 \times 10^{-4}$ ) is also slightly smaller than that of the regular CNN model ( $8.64 \times 10^{-4}$ ). This result indicates that the choice of the loss function during the training of a CNN model also affects how effectively the trained model performs, presumably because the tradeoff between how well the model is fitted for porous structures with low and high  $D_e$  is shifted when different loss functions are used. The latter is evident when predictions by the newly trained CNN model is compared against the LBM predictions. As shown in Fig. 5a, compared to the regular CNN model, the new model performs better for samples with  $D_e < 0.2$  but worse for samples with  $D_e > 0.8$ . These observations are corroborated by the distribution of the absolute errors in the three groups of samples in the testing dataset (see Fig. 5b, upper panels). The improved performance of the CNN model for samples with low  $D_e$  is also evident in the distribution of the truncated relative error (see Fig. 5b, lower panels), e.g., for samples with  $D_e < 0.1$ , 85% of the predictions by the new CNN model is within a truncated relative error of 100%.

## Improving CNN Models Using Field Knowledge

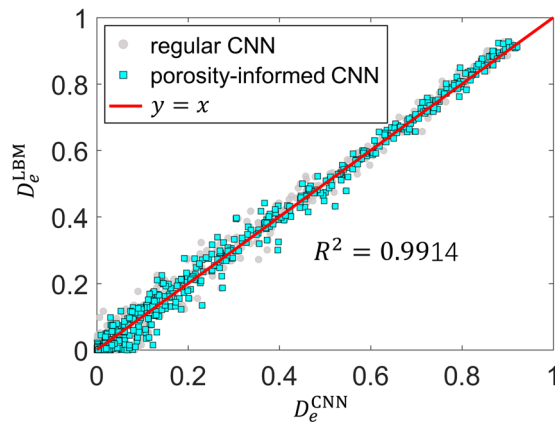
**Porosity-informed CNN model.** In the previous section, we demonstrate that the CNN model performs well for porous structures with a wide range of porosity. Note that the key features determining the effective diffusivity of a porous structure are extracted through the convolutional layers and these features are mostly connected with the input images locally. Therefore, global features or features spanning large scale may not be effectively extracted using the CNN, which may compromise the predictive power of the CNN. Therefore, it may be useful to directly introduce physical parameters describing these features into the CNN model to improve its performance. As described in Section II, here we introduce the overall porosity of the porous media as an input to the first fully connected layer in the CNN. Although the complex porous structures spanning large scales are not easy to identify using CNN models, they can be easily identified based on knowledge of porous media.

The predictions of the porosity-informed CNN model are only marginally improved compared to the regular CNN, e.g., both the MSE and truncated relative error for the testing dataset are smaller than those of the regular





**Figure 5.** (a) The effective diffusivity predicted by the CNN model trained using a loss function based on the truncated relative error (Eqs. (9 and 10)). (b) Distribution of the absolute error (top panels) and truncated relative error (lower panels) of the predictions of the CNN model for the porous structures in the testing dataset with different  $D_e$ .

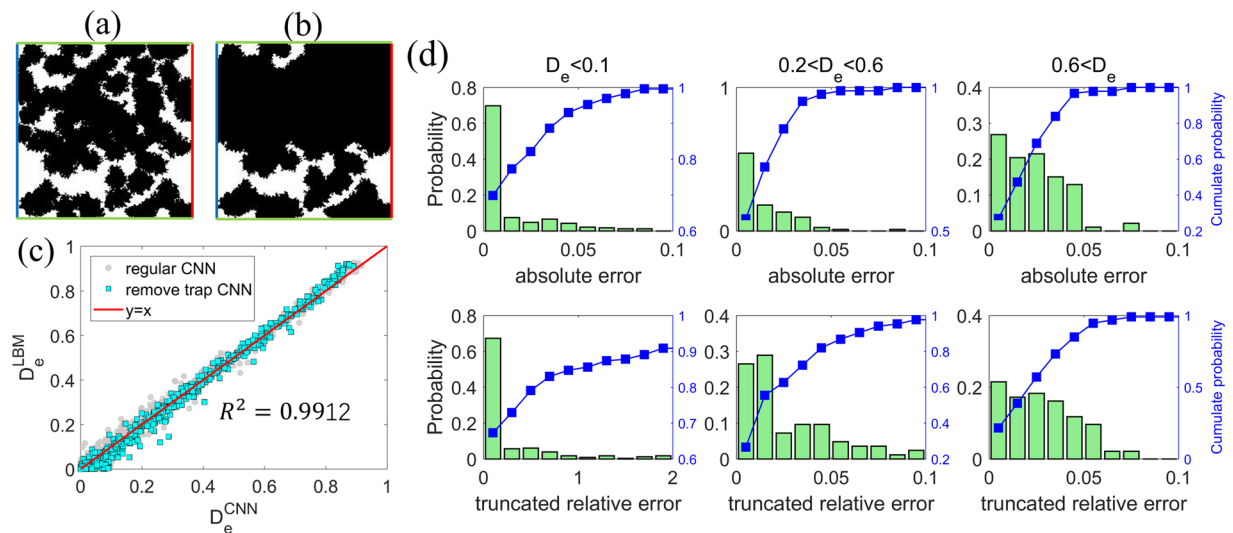


**Figure 6.** The effective diffusivity predicted by the porosity-informed CNN model.

CNN by about 12% (see Table 2). This is also evident in Fig. 6, where the CNN predictions for the entire testing dataset are shown. In particular, we observe that the limitations of the regular CNN model for porous structures with  $D_e \lesssim 0.1$  are not greatly alleviated in the new model. The limited improvement of the new model may originate from several sources. First, porosity is not a reliable indicator of  $D_e$ , especially for porous structures with low  $D_e$  (see Fig. 3b), because the transport is hindered also by complex features such as dead-end transport pathways and trapped space beyond the overall amount of pore space. Second, though porosity is added as additional neurons in the CNN, their impact may have already been captured in the feature maps generated by the filters in the convolutional layer. In this case, the input due to porosity is redundant and thus does not improve the performance of the CNN.

**The CNN model using preprocessed porous structure as input.** The CNN model illustrated in the previous sections works well for porous media with moderate to large diffusivity ( $D_e > 0.2$ ) but exhibits relatively large error for porous samples with very small diffusivity (e.g.,  $D_e < 0.1$ ). This is closely related to the more complex transport behavior in porous media with very small diffusivities: in these media, the diffusion pathways are tortuous and there exist many trapped pores and dead-end paths. Since CNN models may not effectively extract features of these complicated structure spanning relatively large length scales using filters with small spatial extent, they do not perform well for such porous media. Nevertheless, these complex features are easily discerned based on field knowledge of porous media. Hence, here we explore the possibility of improving CNN prediction by processing the images of porous structures to remove dead-end and trapped pore spaces.

To improve the CNN prediction for porous media with small  $D_e$ , all datasets are optimized by removing the trapped and dead-end pore spaces. First, each binary image of the porous structures is labeled using 8-connected component labeling method<sup>25,26</sup> with periodic conditions on the top and bottom side (see Fig. 7a). This labeling method is consistent with LBM simulations based on the D2Q9 scheme, where each target position is associated with the nearby 8 directions. Next, in each labeled image, all components with labels across the inlet and outlet of the porous structure are kept, while other components (trapped pores and dead-end pores) are eliminated.



**Figure 7.** Improving CNN prediction by preprocessing images of porous structures. (a,b) Images of the regular (a) and processed (b) porous structure by removing the trapped pore space and dead-end transport pathways. The blue (red) lines on denote the inlet (outlet) of the molecule transport through the porous structure. The green lines denote the periodic boundaries. (c,d) The effective diffusivity predicted by the CNN model using the preprocessed image of porous structures (c) and the distribution of their absolute and truncated relative errors (d).

Afterward, the porous structure is reconstructed with the remaining pore space as the effective pore space and the rest as the solid space. Figures 7a,b show a porous structure before and after the preprocessing described above. Clearly, although there exists large pore space in the regular porous structure, only part of it is available for molecular transport. Finally, the CNN model is trained using the preprocessed binary images and their corresponding  $D_e$ .

Using the CNN model trained above, we again calculated the effective diffusivity of the porous samples in the testing dataset from their preprocessed images (see Fig. 7c). Overall, the new predictions are in good agreement with the LBM predictions. As shown in Table 2, the MSE of the CNN predictions is  $6.92 \times 10^{-4}$ , which is 20% smaller than that of the regular CNN model. The mean truncated relative error is reduced from 68.8% in the regular CNN predictions to 29.7%. Examination of the distribution of the absolute and truncated errors of the predictions (see Fig. 7d) shows that the predictions are improved considerably for porous samples with  $D_e < 0.6$ , e.g., for samples with  $D_e < 0.1$ , over 70% of the predictions are within a truncated relative error of 20%. Therefore, leveraging field knowledge can help improve the predictions of the CNN model.

Even with the above improvement, for porous samples with  $D_e < 0.1$ , ~10% of the CNN predictions still have a truncated relative error >200%. Examination of these data points revealed that the true  $D_e$  of the corresponding samples is very small ( $\lesssim 10^{-4}$ ) but the CNN prediction are much higher. In these porous samples, only a few diffusion pathways contribute to the net diffusion flux. A representative case is shown in Fig. 7a,b. Although there is a large amount of pore space in the original porous sample (Fig. 7a), only a single diffusion path is found after the trapped pore spaces are identified (Fig. 7b). The net transport through these porous samples is thus dictated by a few narrowest throats along the long, tortuous pathway. Therefore, to accurately predict the effective transport property, both global feature (tortuous pathway) and local feature (critical throats) need to be captured by the CNN model. This need can be difficult to meet by the convolutional layers, which retrieves relatively *local* characteristics of the porous structure. Indeed, similar situations are also frequently encountered in direct simulations of transport phenomena exhibiting multiple length and time scales (e.g., turbulent flows). Second, the number of samples having the above challenging geometrical features is limited and outweighed by other samples in the training dataset. Therefore, these features are likely not extracted accurately in the CNN model.

## Conclusions

In summary, inspired by recent works on application of machine learning in porous media research, we develop deep learning models based on CNN for predicting the effective diffusivity of 2D porous media from their binary images. The computational cost of the model is six orders of magnitude lower than direct pore-scale calculations. The performance of the deep learning model is evaluated in detail by computing the mean square error, mean truncated relative error, and distribution of absolute and truncated relative errors for samples with different effective diffusivity. The effective diffusivity of realistic porous media spanning more than one order of magnitude ( $0.1 \lesssim D_e < 1$ ) can be predicted well even when the porous media contain complex and diverse topologies and have large variation of porosity (0.28–0.98). The performance of the deep learning model, especially when applied to porous media with  $D_e \lesssim 0.1$ , can be improved by training the model using a loss function based on the truncated relative error or preprocessing images of porous media to remove the trapped pore space and dead-end transport pathways. Improvement due to incorporation of porosity of porous media into the CNN, however, is marginal.

Our results demonstrated that the transport properties of realistic porous media with diverse/complex topologies can be predicted with good accuracy and little computational cost. While only 2D porous structure is considered here, extension of the model to 3D situations is possible given that CNN can handle high dimensional data effectively. Indeed, the recent success of extending 2D CNN model to 3D for human pose and gesture recognition<sup>30,31</sup> and medical image process problems<sup>32</sup> suggests that CNN can be highly effective in tackling 3D problems. Nevertheless, there may exist caveats that makes CNN less effective in 3D than 2D for the transport in complex media, and extensive tests must be performed to confirm the feasibility of CNN in predicting the transport properties of 3D porous media.

Although all datasets are generated computationally here for generic porous materials, these data can also be generated experimentally for specific porous materials, e.g., porous structures can be built from CT-scan and X-ray microtomography<sup>33–35</sup>, and effective transport properties such as diffusivity and thermal conductivity can be measured experimentally. The experimentally generated data can be used to inform the computational reconstruction of porous structures for specific classes of porous materials (e.g., shales or ceramic matrix composites) and validate predictions of pore-scale simulations, thereby enabling the creation of high-fidelity computational models of many classes of materials. The creation of these models and pore-scale simulations of them by high-performance computing will produce large, high-quality datasets tailored to different classes of porous materials. Deep learning models based on these datasets will enable fast and accurate prediction of the properties of these materials and benefit their applications. These potential extensions of the present work, along with the predictive power of the CNN models demonstrated here, suggest that deep learning can be a powerful new tool in the future research of porous media.

The analysis of the performance of our current model points to the origins of its current limitations and directions for future development. Our present model does not perform very well when the porous media has a very low effective diffusivity. Such a limitation may be addressed through two possible approaches: *multiscale feature extraction* and *encoding of advanced geometrical properties*. Because the transport in low effective diffusivity porous media is often dictated by a few narrow throats along a few (or even just one) tortuous pathway, the accurate prediction of transport by deep learning models requires the geometrical features at both global and local scale to be extracted effectively and properly weighed during training. Research on multiscale feature extraction can likely benefit from work on feature extractions from complex systems such as turbulent flows. Another way to simultaneously consider both the local and global structure of porous media is to directly encode information of such structure into the deep learning model, thereby by passing the need to extract them using the convolutional layers. For example, given the importance of pore connectivity and pore size distribution in the transport in porous media, quantitative measures of these properties may be encoded into the deep learning model in the future. Identifying the best information to encode into deep learning models will benefit from the immense field knowledge on transport in porous media accumulated by the community over the past decades.

## Data availability

The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

Received: 11 April 2019; Accepted: 30 November 2019;

Published online: 31 December 2019

## References

- Chen, L. *et al.* Nanoscale simulation of shale transport properties using the lattice Boltzmann method: permeability and diffusivity. *Sci. Rep.* **5**, 8089 (2015).
- Chen, L., Kang, Q., Dai, Z., Viswanathan, H. S. & Tao, W. Permeability prediction of shale matrix reconstructed using the elementary building block model. *Fuel* **160**, 346–356 (2015).
- Chen, L. *et al.* Generalized lattice Boltzmann model for flow through tight porous media with Klinkenberg's effect. *Physical Review E* **91**, 033004 (2015).
- Lunati, I. & Lee, S. A dual-tube model for gas dynamics in fractured nanoporous shale formations. *J. Fluid Mech.* **757**, 943–971 (2014).
- Shokri, N., Sahimi, M. & Or, D. Morphology, propagation dynamics and scaling characteristics of drying fronts in porous media. *Geophysical Research Letters* **39** (2012).
- Jackson, G. W. & James, D. F. The permeability of fibrous porous media. *The Canadian Journal of Chemical Engineering* **64**, 364–374 (1986).
- Millington, R. Gas diffusion in porous media. *Science* **130**, 100–102 (1959).
- Yang, Z. *et al.* Deep learning approaches for mining structure-property linkages in high contrast composites from simulation datasets. *Computational Materials Science* **151**, 278–287 (2018).
- Cecen, A., Dai, H., Yabansu, Y. C., Kalidindi, S. R. & Song, L. Material structure-property linkages using three-dimensional convolutional neural networks. *Acta Materialia* **146**, 76–84 (2018).
- He, K., Zhang, X., Ren, S. & Sun, J. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–78 (2016).
- Krizhevsky, A., Sutskever, I. & Hinton, G. E. In *Advances in neural information processing systems*. 1097–1105 (2012).
- Cireřan, D., Meier, U. & Schmidhuber, J. Multi-column deep neural networks for image classification. *arXiv preprint arXiv:1202.2745* (2012).
- LeCun, Y., Bottou, L., Bengio, Y. & Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* **86**, 2278–2324 (1998).
- Wu, J., Yin, X. & Xiao, H. Seeing permeability from images: fast prediction with convolutional neural networks. *Science Bulletin* **63**, 1215–1222 (2018).
- Cang, R., Li, H., Yao, H., Jiao, Y. & Ren, Y. Improving direct physical properties prediction of heterogeneous materials from imaging data via convolutional neural network and a morphology-aware generative model. *Computational Materials Science* **150**, 212–221 (2018).
- Srisuthiyakorn, N. Deep-learning methods for predicting permeability from 2D/3D binary-segmented images. *SEG technical program expanded abstracts* **2016**, 3042–3046 (2016).

17. Wang, M., Wang, J., Pan, N. & Chen, S. Mesoscopic predictions of the effective thermal conductivity for microscale random porous media. *Physical Review E* **75**, 036702 (2007).
18. Fang, W.-Z., Gou, J.-J., Chen, L. & Tao, W.-Q. A multi-block lattice Boltzmann method for the thermal contact resistance at the interface of two solids. *Applied Thermal Engineering* **138**, 122–132 (2018).
19. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436 (2015).
20. LeCun, Y. & Bengio, Y. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks* **3361**, 1995 (1995).
21. Dollár, P., Appel, R., Belongie, S. & Perona, P. Fast feature pyramids for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **36**, 1532–1545 (2014).
22. Ramachandran, P., Zoph, B. & Le, Q. V. Searching for activation functions. *arXiv preprint arXiv:1710.05941* (2017).
23. Aghdam, H. H. & Heravi, E. J. Guide to Convolutional Neural Networks. New York, NY: Springer. **10**, 978–973 (2017).
24. He, K., Zhang, X., Ren, S. & Sun, J. In *European conference on computer vision*. 630–645 (Springer) (2016).
25. He, L., Chao, Y., Suzuki, K. & Wu, K. Fast connected-component labeling. *Pattern Recognition* **42**, 1977–1987 (2009).
26. Fisher, R., Perkins, S., Walker, A. & Wolfart, E. In *IEEE International Conference on Intelligent Systems*. 300–350 (2003).
27. Abadi, M. *et al.* In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*. 265–283.
28. Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
29. Lange, K. J., Sui, P.-C. & Djilali, N. Pore scale simulation of transport and electrochemical reactions in reconstructed PEMFC catalyst layers. *Journal of The Electrochemical Society* **157**, B1434–B1442 (2010).
30. Molchanov, P., Gupta, S., Kim, K. & Kautz, J. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 1–7 (2015).
31. Ji, S., Xu, W., Yang, M. & Yu, K. 3D convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence* **35**, 221–231 (2012).
32. Dou, Q. *et al.* Automatic detection of cerebral microbleeds from MR images via 3D convolutional neural networks. *IEEE transactions on medical imaging* **35**, 1182–1195 (2016).
33. King, H. E. Jr. *et al.* Pore architecture and connectivity in gas shale. *Energy & Fuels* **29**, 1375–1390 (2015).
34. Wildenschild, D. & Sheppard, A. P. X-ray imaging and analysis techniques for quantifying pore-scale structure and processes in subsurface porous medium systems. *Advances in Water Resources* **51**, 217–246 (2013).
35. Sok, R. M. *et al.* Pore scale characterization of carbonates at multiple scales: Integration of Micro-CT, BSEM, and FIBSEM. *Petrophysics* **51** (2010).

## Acknowledgements

We thank the ARC at Virginia Tech for generous allocations of computer time. Q.K. acknowledges the support from LANL's LDRD Program.

## Author contributions

R.Q., Q.K. and W.Q.T. conceived the project, H.W. and W.Z.F. designed the project and performed all simulations; H.W. and W.Z.F. analyzed the simulation results with input from everyone. H.W. led the writing of the manuscript and everyone participated in editing the manuscript.

## Competing interests

The authors declare no competing interests.

## Additional information

**Supplementary information** is available for this paper at <https://doi.org/10.1038/s41598-019-56309-x>.

**Correspondence** and requests for materials should be addressed to Q.K., W.-Q.T. or R.Q.

**Reprints and permissions information** is available at [www.nature.com/reprints](http://www.nature.com/reprints).

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2019