



## OPEN Deep learning for diffusion in porous media

Krzysztof M. Graczyk , Dawid Strzelczyk & Maciej Matyka

We adopt convolutional neural networks (CNN) to predict the basic properties of the porous media. Two different media types are considered: one mimics the sand packings, and the other mimics the systems derived from the extracellular space of biological tissues. The Lattice Boltzmann Method is used to obtain the labeled data necessary for performing supervised learning. We distinguish two tasks. In the first, networks based on the analysis of the system's geometry predict porosity and effective diffusion coefficient. In the second, networks reconstruct the concentration map. In the first task, we propose two types of CNN models: the C-Net and the encoder part of the U-Net. Both networks are modified by adding a self-normalization module [Graczyk et al. in *Sci Rep* 12, 10583 (2022)]. The models predict with reasonable accuracy but only within the data type, they are trained on. For instance, the model trained on sand packings-like samples overshoots or undershoots for biological-like samples. In the second task, we propose the usage of the U-Net architecture. It accurately reconstructs the concentration fields. In contrast to the first task, the network trained on one data type works well for the other. For instance, the model trained on sand packings-like samples works perfectly on biological-like samples. Eventually, for both types of the data, we fit exponents in the Archie's law to find tortuosity that is used to describe the dependence of the effective diffusion on porosity.

Diffusion transport in complex porous structures is ubiquitous in nature<sup>1–4</sup>. A prominent example is the brain, where diffusion is a dominant process for nutrient and signal transport<sup>5–7</sup>. The brain's extracellular space, filling the void between neuropil cells, can be treated as a porous medium of complex topology. Effective properties of diffusion in porous media are investigated with various techniques. The application type determines the choice of the method<sup>8</sup>. The fields of applications concern diverse domains such as assessment of the tortuosity of extracellular space in brain studies<sup>5,9</sup>, analysis of the diffusion in Li-ion batteries<sup>10</sup>, sandy sediments<sup>2</sup> and studies of hierarchical porous materials<sup>11</sup>.

The properties of transport in the void space of porous materials strongly depend on the vastly diverse geometry of its pores. Sandstone rocks, made of grains of the size on the order of tens to hundreds of  $\mu\text{m}$  might exhibit pores multiple times smaller while porosities, defined as a ratio of the pore volume to the overall volume:

$$\varphi = \frac{V_{\text{pore}}}{V}, \quad (1)$$

can be as low as several percent<sup>12</sup>. On the other hand, the extracellular space, which separates cells in tissues of living organisms, is composed of sheets and tunnels of width on the order of tens of nanometers with numerous dead-ends and cells winding around each other<sup>5,13,14</sup>. Therefore, an effort has been made to develop tools to predict diffusive transport properties in porous media of extremely different geometries. In this paper, we adapt convolutional neural networks to study two distinct types of porous media: isotropic granular, corresponding to, e.g., sand packings to rocks, and a system of channels resembling brain extracellular space geometry.

Diffusive transport is studied on micro and macro scales. In the first, the pore space in which the transport occurs is explicitly modeled. In the second, the effects of the porous medium on the transported molecules are spatially averaged and represented by effective coefficients. One distinguishes two general approaches to model diffusion transport. In the first, one traces individual particles (e.g. using random walk). On the other, one considers the concentration field obtained from solving the diffusion equation. This paper exploits the second approach: we solve the diffusion partial differential equation. We evaluate the concentration field,  $c(\mathbf{r}, t)$  ( $\mathbf{r}$  (the position vector,  $t$  refers to time), to study the phenomenon.

Faculty of Physics and Astronomy, Institute of Theoretical Physics, University of Wrocław, pl. M. Borna 9, 50-204 Wrocław, Poland. ✉email: krzysztof.graczyk@uwr.edu.pl

The diffusion equation describes spatial and temporal changes in concentration field of diffusing matter. It can be exploited to model diffusive transport in both scales. At the microscale, for bulk diffusion coefficient  $D_0(\mathbf{r})$  (variable in space), the equation reads:

$$\frac{\partial c}{\partial t} = \nabla \cdot (D_0(\vec{r}) \nabla c). \quad (2)$$

For a constant isotropic diffusivity inside the pore volumes, this simplifies to  $\partial c / \partial t = D_0 \nabla^2 c$  and the diffusive flux value depends only on the concentration gradient in the direction of the flux. However, in porous media, the diffusion process is hindered<sup>15</sup> or accelerated<sup>16,17</sup> due to phenomena like interactions of matter with solid obstructions (steric effects) or binding of species at certain locations in pores. In practice we represent the complex porous system by a continuous medium with effective diffusion coefficient  $D(t)$ , obtained from the Fick's law

$$\langle D_0 \nabla c(\vec{r}, t) \rangle = D(t) \frac{c_{\text{out}} - c_{\text{in}}}{L}, \quad (3)$$

where  $\langle \dots \rangle$  denotes averaging over the pore space,  $L$  is the distance between the sample inlet and outlet, while  $c_{\text{in}}$  and  $c_{\text{out}}$  are concentrations at the sample inlet and outlet, respectively. For sufficiently long diffusion times  $D(t)$  approaches a steady value  $D$ , which usually differs from the bulk diffusion coefficient  $D_0$ . The ratio between the bulk diffusion coefficient  $D_0$  and the effective diffusion coefficient  $D$  is used to define the diffusive tortuosity of a medium:

$$\lambda = \frac{D_0}{D}. \quad (4)$$

The above quantity appears in the scalar transport equation applicable to porous media at the macroscale (see, e.g., Eq. 9 in Sykova and Nicholson<sup>5</sup>). The relation between the diffusive tortuosity and porosity of a porous sample is a subject of ongoing debate with a gap to be bridged between experimental and theoretical studies<sup>2</sup>.

Due to the complexity of the boundary conditions solving analytically Eq. (2) is, in most cases, impossible. Numerical solutions have several drawbacks, such as quadratic scaling of timestep or computationally expensive matrix manipulations. The above issues become especially prohibitive when one needs to solve the diffusion equation for a large number of data to perform statistical analysis of a given type of porous media, as in Kasthuri et al.<sup>18</sup>.

In this paper, we adopt convolutional neural network (CNN) systems to model the diffusive transport in the porous media. Recently, deep learning-based techniques have been used to compute effective properties of porous media, including permeability,<sup>19,20</sup>. An example of the usage of CNNs to calculate both permeability and tortuosity is given by Graczyk and Matyka<sup>21</sup>. In Cawte and Bazylak<sup>22</sup>, diffusion and permeability are studied within machine learning tools, including gradient boosting regression, neural network, and support vector regression. Image-based prediction of effective diffusion,  $D$ , was performed for scanning electron microscope data<sup>23</sup> and random field-based porous media<sup>24</sup>. Kamrava et al.<sup>25</sup> considered the U-Net model, also exploited in this paper, to predict the pressure and velocity field's spatial distribution in porous membranes. Permeability of porous samples generated with Voronoi tessellation was predicted within physics informed CNN by Wu et al.<sup>26</sup>. Kamrava et al.<sup>27</sup> simulated the fluid flow in a porous medium within the deep learning model that incorporated mass conservation and the Navier-Stokes equations in its learning process. Eventually, a comprehensive review of the usage of machine learning methods in studies of porous media and geosciences is provided by Tahmasebi et al.<sup>28</sup>.

Our goal is to develop a deep learning (DL) system, a deep neural network (DNN), that, based on the system's geometry analysis, predicts either the porosity and effective diffusion coefficient  $D$  or concentration field  $c(\mathbf{r})$  in the steady state. We treat both goals separately. In the first case, we consider two types of CNNs architectures. One of them is adapted from our previous work<sup>21</sup> on predicting porosity, permeability, and tortuosity in porous media. Additionally, both architectures are modified by adding a self-normalized (SN) module, proposed by Graczyk et al.<sup>29</sup>. The modified architectures work with higher precision than vanilla ones. Moreover, the networks without the SN module tend to overfit the data. Eventually, we propose to consider the U-Net architecture to predict the concentration distribution<sup>30</sup>, and it works with better accuracy and in a broader range than networks that only predict porosity and diffusion coefficient.

The DL systems are obtained from supervised learning. In such an approach, the critical is to work with the data which represent the studied feature comprehensively. We generate two types of samples: systems of randomly deposited isotropic grains (mimicking the sand packings), and random systems derived from Voronoi tessellation of a plane (mimicking extracellular space of tissues).

In practice, we consider the 2D pictures of the systems with geometry configuration. Then, exploiting the Lattice Boltzmann Method (LBM), we simulate the diffusion process and obtain a steady-state concentration field. From this, using Eq. (3), we calculate the effective diffusion coefficient. The image representing the porous medium's geometry is the input for the network that predicts either the porosity, diffusion coefficient or concentration field.

The network models are obtained from the statistical analysis of the data, and hence, the networks predict the porosity and diffusion coefficient with some uncertainty. Therefore, we exploit the Monte Carlo dropout technique<sup>31</sup> to estimate how certain in predictions our models are.

Similar studies were performed by Wu et al.<sup>32</sup>. They adopted AlexNet and ResNet50 CNN models to predict the diffusion coefficient, and they considered one type of data in a broader range of porosity than in our analysis. In contrast, we exploited two other types of porous media. Moreover, our network models are of different architectures. One of the architectures was considered in our previous work<sup>21</sup>. The other is the encoder part of the U-Net architecture. All our architectures contain batch normalization layers, which was essential for getting

successful fits<sup>21</sup>. Eventually, we modify the vanilla models by adding the self-normalization module<sup>29</sup>. Additionally, in this paper, we study the adaptation of the U-Net architecture for reconstructing the concentration field.

The structure of the paper is the following: in Sect. “Diffusion from LBM–data preparation”, the method of preparation of the data is shortly presented; in Sect. “DNN models”, we introduce the network models. The numerical results for both types of the tasks are presented in Sects. “Porosity and diffusion coefficient from deep learning” and “Reconstruction of the concentrations by U-Net”. We conclude in the Sect. “Summary”. The paper contains Appendix 5, that summarizes the LBM method.

### Diffusion from LBM–data preparation

We shall obtain the DL systems that can make predictions about the diffusion phenomenon. To accomplish this task, we need labeled data which will be used to train the DNNs. We consider geometries of randomly generated porous domains of two kinds:

- Type-A: systems of randomly deposited isotropic grains;
- Type-B: samples based on Voronoi tessellation.

Each sample has a size  $L \times L$ ,  $L = 128$  nodes. Each node can be either fluid or solid. All fluid nodes belong to the  $\mathcal{D}_f$  set, while all solid nodes belong to the  $\mathcal{D}_s$  set. The nodes outside the  $L \times L$  square belong to the  $\mathcal{D}'$  set.

Both types of systems have controllable porosity ranging from devoid of obstacles ( $\varphi = 1$ ) down to the percolation threshold. Samples of type-A are idealizations of granular media such as sand packings, clay, or fluidized granular beds. We generate them by placing  $3 \times 3$  square obstacles in an initially all-fluid domain; see Fig. 1 (two left panels). The coordinates of obstacle positions are drawn from the uniform distributions in the range  $[0, L]$ , and obstacles can overlap. Filling the sample configuration with blocks stops when the assumed porosity is achieved. Then the percolation test between the left and right boundary is performed. Samples that do not pass the test are rejected from the dataset.

Type-B domains’ geometries are derived from biological systems such as the extracellular space of tissues; see Fig. 1 (two right panels). We generate them by placing 50 points in  $L \times L$  square. The points’ position coordinates are drawn from the real uniform distribution on the interval  $[0, L]$ . Then the set is duplicated to eight neighboring domains to minimize the boundary effects. Then, Voronoi tessellation of  $\mathbb{R}^2$  space based on the drawn points is performed to obtain the edges of the polygons, which are the centerlines of the pore channels. We start discretization from all-solid nodes, which corresponded to zero-width channels. Then the width of the channels is iteratively and uniformly increased, and nodes laying inside any channel are marked as fluid, see Fig. 1. The iteration procedure stops when the assumed porosity is achieved. Due to the nature of the edges network, the samples are guaranteed to percolate from left to right, and no additional tests are performed.

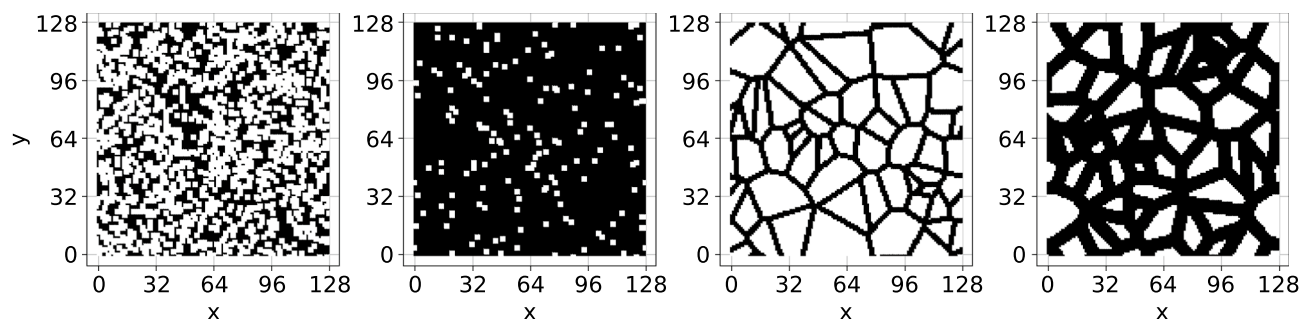
We assume reflecting (zero normal flux) boundary conditions on obstacles and top/bottom walls. The Dirichlet boundary conditions are imposed on the left ( $c([0, y], t) = 0 = c_{in}$ ) and right ( $c([L, y], t) = 1 = c_{out}$ ) boundary.

We numerically solve the diffusion equation with the LBM to obtain the effective diffusion coefficient and concentrations. The details of the approach are given in Appendix 5. Each LBM simulation is iterated until the maximum change of the concentration field between two consecutive iterations  $\Delta c_n$  defined as:

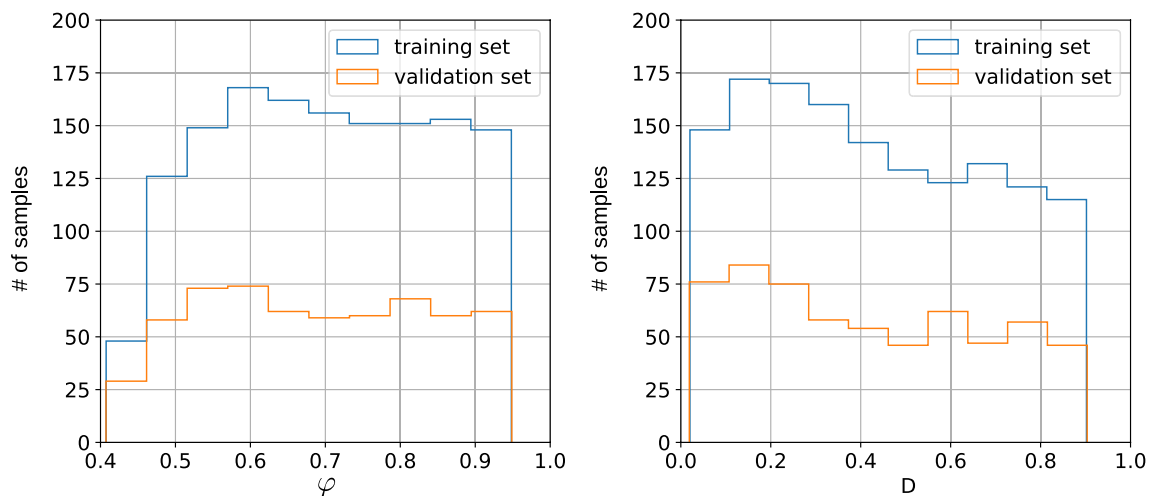
$$\Delta c_n = \max_{i: \vec{r}_i \in \mathcal{D}_f} \left( |c_i^n - c_i^{n-1}| \right) \quad (5)$$

is smaller than  $10^{-13}$  or up to  $n = 10^6$  iterations. Then the effective diffusion coefficient  $D$  is calculated using discrete implementation of Eq. (3), namely,

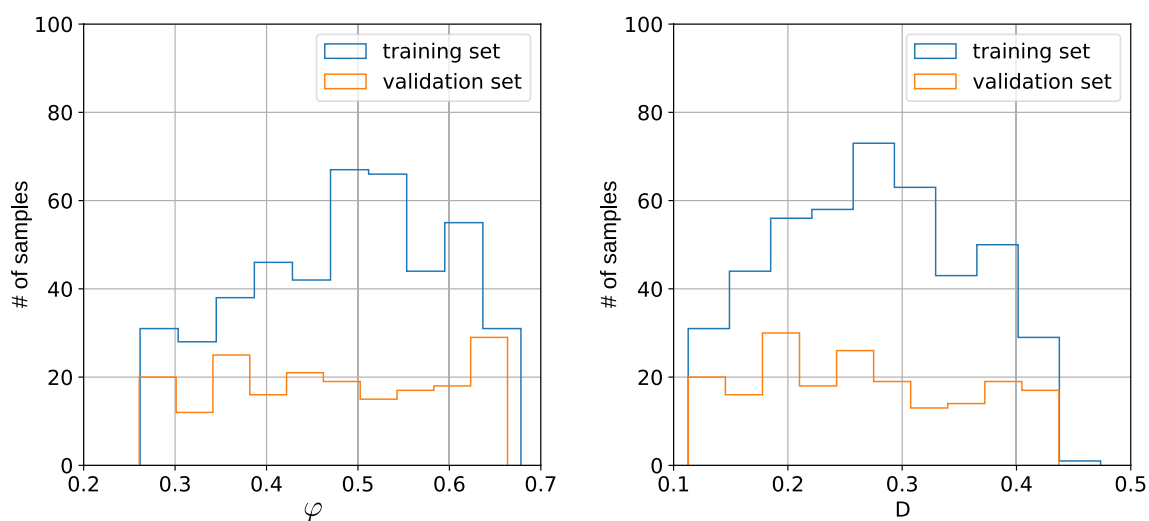
$$D = \frac{L}{N(c_{out} - c_{in})} \sum_{\vec{r}_i \in \mathcal{D}_f} (\nabla c)|_{\vec{r}_i}, \quad (6)$$



**Figure 1.** Type-A samples with  $\varphi = 0.42$  and  $\varphi = 0.92$  (from left: the first and the second, respectively) and type-B samples with  $\varphi = 0.32$  and  $\varphi = 0.66$  (from left: the third and the fourth, respectively). Fluid nodes are denoted by black, while solid nodes are denoted by white. We show samples with low and high porosity values for both data types.



**Figure 2.** Distributions of porosities  $\varphi$  and effective diffusion coefficients  $D$  obtained with LBM for 2017 type-A samples (1412 and 605 for training and validation datasets, respectively).



**Figure 3.** Distributions of porosities  $\varphi$  and effective diffusion coefficients  $D$  obtained with LBM for 640 type-B samples (448 and 192 for training and validation datasets, respectively).

where  $c_i^n$  denotes concentration value at node  $i$  at iteration  $n$ . Note that we choose the LBM parameters in a way that  $D_0 = 1$ .

We generate the data for two tasks: predicting  $\varphi$  and  $D$  from geometry and predicting the concentrations from the geometry. For both of the tasks, we produce type-A and type-B data. Each generated dataset is split into training and validation datasets. In Figs. 2 and 3, we show the distributions of porosity values  $\varphi$ , and effective diffusion coefficients  $D$ , for 2017 samples of type-A and 640 samples of type-B.

### Diffusion phenomenon from deep learning

In the present paper, we exploit the deep neural network to construct the computational system that predicts:

- I Porosity,  $\varphi$ , and effective diffusion coefficient  $D$ , see Eq. (4);
- II The distribution of the concentrations  $c(\vec{r}, t)$  in the system.

In both tasks, the network takes a picture of the system's geometry for input. In the first, the network predicts  $\varphi$ ,  $D$ , and in the other, the final concentration map.

In the following sections, we shall discuss tasks I and II separately. The procedure for obtaining a DNN for task I can be summarized as follows:

- (i) Consider the samples of the given type of data;
- (ii) Each sample has assigned porosity and diffusion coefficient;
- (iii) Train the DNN on training dataset, and validate on validation dataset

- (iv) Test the obtained DNN on the other data type.

For task II the network reconstructs the concentration field. Therefore, the data contains the pairs of figures: the input picture with the system's geometry and the output picture with the concentration field.

The full analysis is conducted in PyTorch<sup>33</sup>.

**DNN models.** For task I, we consider two types of network architecture, namely:

- C-Net: a network of similar structure as the one used in our previous paper<sup>21</sup> designed to predict the porous medium's porosity, permeability, and (hydraulic) tortuosity.
- U-Net-Half: it is the decoder part of the U-Net network<sup>30</sup> exploited by us to reconstruct the maps of concentrations in the medium.

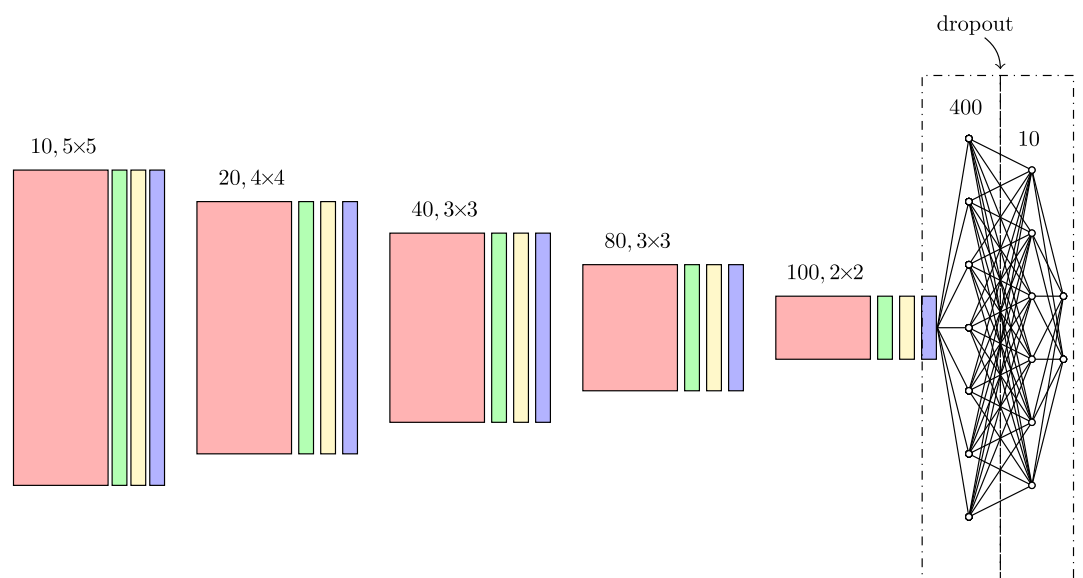
The C-Net is the convolutional neural network containing five blocks; see Fig. 4. On the top of the network are two subsequent fully connected layers with 400 and 10 hidden units and tanh and linear activations functions, respectively. The network takes the input picture of size  $128 \times 128$  with geometry's configuration. Its output is a two-dimensional vector  $(\varphi, D)$ . Each convolutional block contains a convolutional layer (with ReLU activation function), followed by batch normalization, and dropout layers, and the max pooling (with  $2 \times 2$  kernel) layer. After the section of the convolutional blocks, there are two fully connected layers. After the first fully connected layer, there is a dropout layer. The convolutional layers consists of 10, 20, 40, 80 and 100 kernels, of size  $5 \times 5$ ,  $4 \times 4$ ,  $3 \times 3$ ,  $3 \times 3$ , and  $2 \times 2$ , respectively. More details about the structure of the C-Net can be found in the caption of Fig. 4.

The U-Net-half is the encoder part of the U-Net (described below), followed by two fully connected layers of the same structure as in the C-Net.

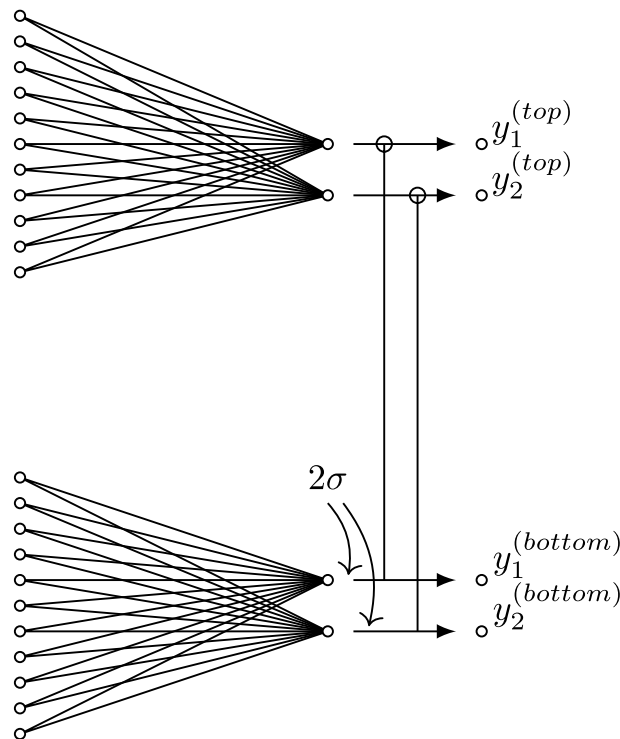
Eventually, for task I, we consider two more network architectures obtained from the C-Net and the U-Net-Half, by replacing the last hidden layer with the self-normalized (SN) module. It is the structure proposed by Graczyk et al.<sup>29</sup> to correct the network's output. The SN mechanism is motivated by the observation that getting the deep neural network with good qualitative predictions is usually simple. But on the quantitative level, the system systematically overshoots or undershoots. The self-normalization module corrects the network response and increases accuracy in the predictions. This module is a part of the system, and its parameters are a subject of optimization too.

The SN module is shown in Fig. 5. In practice, there are two similar-size, parallel, disconnected (from each other) fully-connected layers instead of one fully-connected layer. The top fully-connected layer has two-dimensional output multiplied by the corresponding output of the bottom layer, namely,

$$(y_1^{(top)}, y_2^{(top)}) \rightarrow (y_1^{(top)} y_1^{(bottom)}, y_2^{(top)} y_2^{(bottom)}), \quad (7)$$



**Figure 4.** The C-Net architecture. It contains five convolutional sections. Each of them consists of the convolutional layer (red) with ReLU, the stride one and the padding zero, batch normalization layer (green), max pooling layer with kernel  $2 \times 2$  (violet), and dropout layer (yellow). Over each convolutional layer, there is a number and a size of kernels specified. There is a dropout layer between the last two fully connected layers with 400 and 10 hidden units. For the fully connected layer with 400 units, tanh is the activation function, while for the layer with 10 units, the linear activation function is chosen.



**Figure 5.** The self-normalized module for the C-Net (or U-Net-Half networks). The last hidden layer of the C-Net (or U-Net-Half), outlined part of the Fig. 4, is replaced by the SN module.

where  $(y_1^{(bottom)}, y_2^{(bottom)})$  is the bottom output, where

$$y_i^{(bottom)} = 2\sigma(\cdot), \quad (8)$$

$\sigma$  is the sigmoid function. With such a definition,  $y_i^{(bottom)}$  plays a role of normalization factor ranging from 0 to 2. In principle, these factors should be around one. To enforce the network to have proper normalization, we consider the loss function with an additional penalty term, namely,

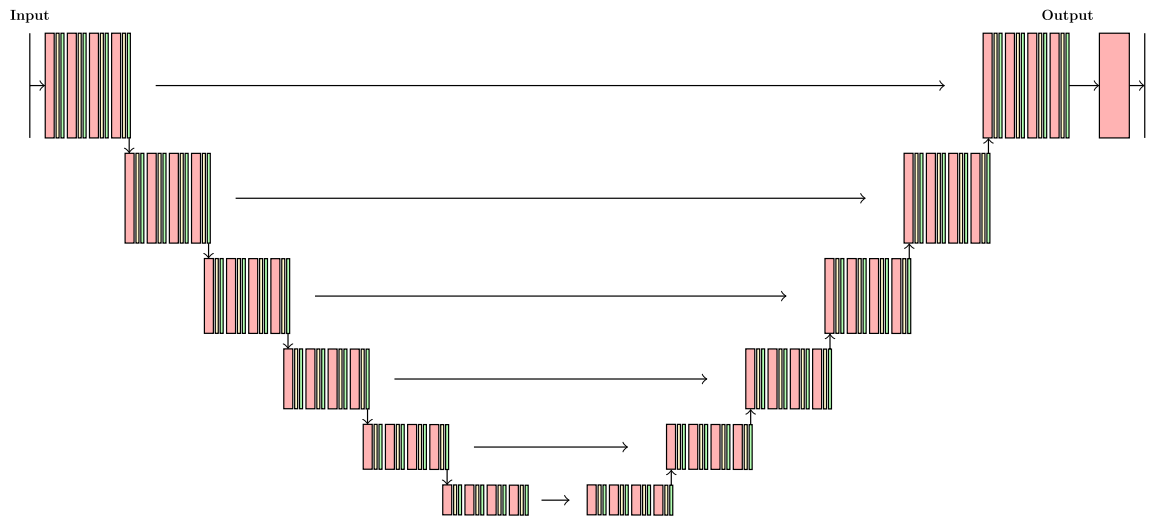
$$L(s_i) \rightarrow L(s_i) + (1 - y_1^{(bottom)})^2 + (1 - y_2^{(bottom)})^2, \quad (9)$$

where  $L(s_i)$  is the loss for the  $s_i$  sample. For  $L(\dots)$ , we consider the mean-square error (MSE) loss in our analysis.

For task II, we propose considering the U-Net architecture, designed to face the biomedical image segmentation problem<sup>29,30</sup>. This type of network architecture was utilized, i.e., to predict the membrane's flow properties, taking its morphology as input<sup>25</sup> and simulating a fluid flow in the porous materials<sup>27</sup>. It has a more complicated structure than the C-Net network; see Fig. 6. In its structure, we can distinguish an encoder and a decoder part. The encoder consists of a sequence of six containers. Each of them downsamples the input by two. Each container has four blocks containing a convolutional layer (with ReLU activation), dropout, and batch normalization layers. After four blocks, the max pooling layer downsamples the input. The decoder includes six corresponding upsampling containers, and each upscales the input size by a factor of two. The upsampling container consists of four convolutional blocks (the same structure as in the encoder), followed by the upsampling layer. The encoder and decoder containers of the same output-input size are connected. Eventually, there is a block of convolutional layers (with batch normalization and dropout layers) after the decoder section. Fig. 6 contains a detailed description of the U-Net structure. Note that each downsampling or upsampling container has  $N = 4$  convolutional blocks. In the pre-analyses, we started with  $N = 1$ , and after several tests, we obtained an optimal value of  $N = 4$ .

To estimate how uncertain the network predictions are, we exploit the Monte Carlo (MC) dropout technique<sup>31,34</sup>. Graczyk et al.<sup>29</sup> adopt this method to estimate uncertainties for network predictions in the analysis of microbiological data. In short, to get  $1\sigma$  uncertainty for the network response, we run network  $M = 20$  times, keeping the dropout layers active. Then the model response is given by the average overall network responses, while from the standard deviation, the  $1\sigma$  uncertainty is obtained.

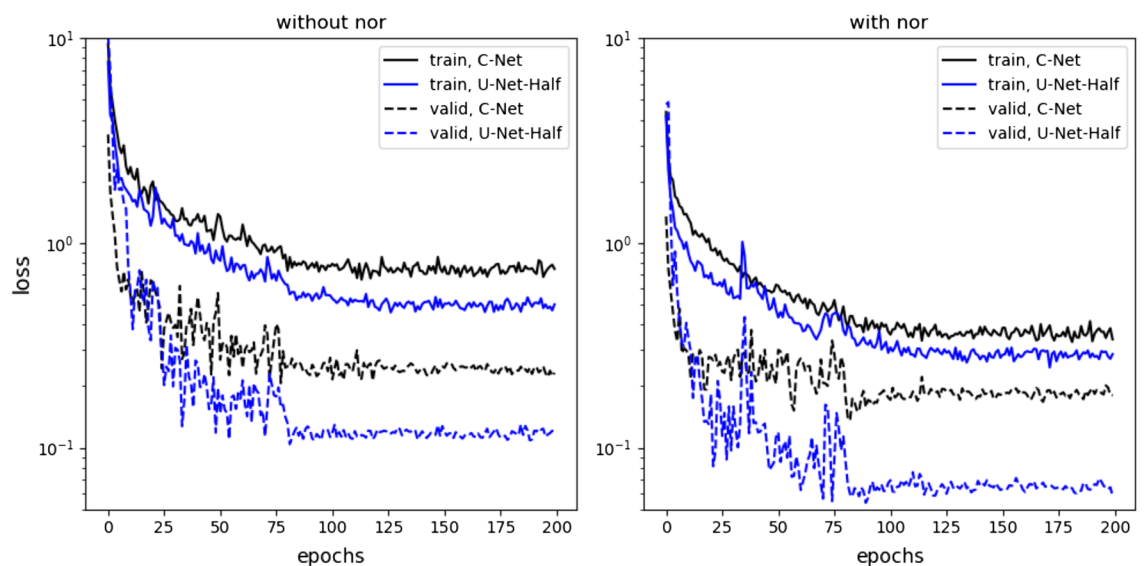
**Porosity and diffusion coefficient from deep learning.** Our first task is to obtain the system that predicts the porosity and diffusion coefficient based on the input picture of the system's geometry. Here we distinguish two scenarios for model development. In the first scenario, the networks are trained on type-A data, and the validation dataset is of the same kind. Then we verify how the model works on the type-B data. The second scenario refers to the models trained and validated on the type-B dataset and tested on the type-A dataset.



**Figure 6.** The U-Net architecture. Each downsampling container consists of ( $N = 4$ ) blocks containing a convolutional layer (with ReLU activation, 64 kernels of  $3 \times 3$  size), dropout (yellow), and batch normalization layers (green). The padding in the convolutional layer is one. After each downsampling block, there is a max pooling (with kernel  $2 \times 2$ ) visualized by vertical arrow. Similarly, each upsampling container comprises four blocks of the same structure as the downsampling block. After four such blocks, there is upsampling (by two) layer (vertical arrow). Note that the upsampling container connects with the previous container's output and the corresponding block's output in the downsampling section. In the final section of the network, there is the downsampling container (without the max pooling) and convolutional layer, which reproduces a one-channel density map of the same size as the input map.

Note that the loss is given by the mean-square error (MSE). In the data augmentation, we exploit the property that a horizontal flip of the input picture does not change the porosity and diffusion coefficient of the system. In Fig. 7, we show how the value of the loss varies during the training. The plots are obtained for all discussed network models for the first scenario - training and validation datasets are of type-A. In this analysis, we set  $p = 0.1$  in the dropout layers. It turned out to be the most optimal value for the dropout mechanism.

The detailed analysis of Fig. 7 leads to the conclusion that adding the self-normalization module improves the performance of the networks. Indeed the errors computed on the validation dataset for the models with SN-module are systematically lower than those obtained for the vanilla networks. On the other hand, the errors computed for the C-Net networks are systematically larger than for the U-Net-half models, which can be interpreted that the U-Net-Half works better than C-Net. To study this problem more carefully, we introduce



**Figure 7.** The MSE loss during networks training. The dashed and solid curves correspond to an error on the validation and training dataset, respectively. The blue and black plots correspond to the results for the U-Net-Half and the C-Net, respectively. All networks contain dropout layers with  $p = 0.1$ . The results for the models without/with the self-normalization module are shown in the left/right figure.

$$\bar{\chi}^2 = \frac{1}{N} \sum_{i=1}^N \left[ \frac{(\varphi_{pred}^i - \varphi_{true}^i)^2}{\Delta\varphi^i{}^2} + \frac{(D_{pred}^i - D_{true}^i)^2}{\Delta D^i{}^2} \right], \quad (10)$$

where  $\varphi_{pred/true}^i$  and  $D_{pred/true}^i$  denote the predicted/true values for porosity and diffusion coefficient for the  $i$ -th sample, respectively. The  $\bar{\chi}^2$  includes information about the uncertainties,  $\Delta\varphi$ , and  $\Delta D$ , obtained from the MC dropout.

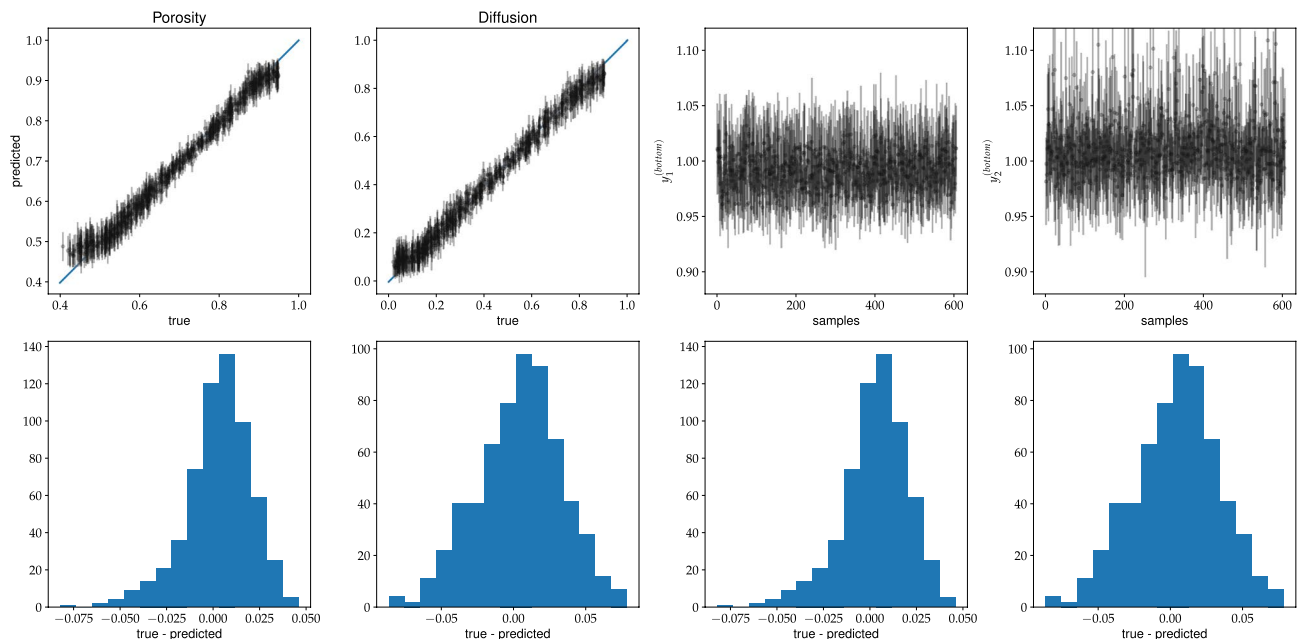
For the C-Net and the U-Net-Half (both with SN module) models trained on type-A dataset, we obtained  $\bar{\chi}^2 = 1058.4$  and  $\bar{\chi}^2 = 1465.4$  (computed on the validation dataset type-A). In this case, the U-Net-half tends to overfit the data more than the C-Net. Unsurprisingly, the C-Net network is defined by a much lower number of parameters than U-Net-Half. Indeed, the size of the first network is about 300 kB, while the other is about 3.3 MB. Accordingly, the C-Net with SN-module is a better model for predicting porosity and diffusion coefficient for type-A dataset. However, when the models are trained on the type-B dataset, the  $\bar{\chi}^2$ , computed on the validation dataset, takes the values: 342.7 and 237.0 for the C-Net and the U-Net-Half (both with SN module), respectively. Hence, for type-B data, the U-Net-Half is more certain. Moreover, the models trained on the type-B data sets are more accurate in the predictions than models trained on type-A. This property is summarized in Table 1, where we give the MSE errors for both analyzed scenarios.

Let us discuss the quality of our fits in a more detailed way. In Fig. 8, we present the predictions for the validation dataset of type A of the C-Net trained on the same type of data. A good match between the predictions and true values of  $(\varphi, D)$  is observed. It is not the case, when one makes predictions for type-B dataset, see Fig. 9. The model overestimates the porosity and underestimates the diffusion coefficient in this case. However, the network's predictions are consistent within the two-sigma level for the samples with  $\varphi > 0.4$ . Note that type-A data does not contain samples with  $\varphi < 0.4$ . Indeed, it is challenging to generate type-A data with porosity smaller than 0.4 as it is close to the percolation threshold of discrete overlapping quads, as shown in Koza et al.<sup>35</sup> Moreover, it is so unlikely that the algorithm generating type-A samples will produce samples of type-B data. Hence, it is no surprise that the network, trained on type-A data, does not work for type-B data, especially in the porosity range not accessible by samples of type-A.

In the second scenario, the models are trained on type-B data. As stated above, U-Net-Half works better than the C-Net architectures in this case. Fig. 10 presents the predictions of the U-Net-Half (with SN module) for the type-B validation dataset. The network reproduces the data well, even in the low porosity range. However, the model does not work for type-A data samples (not shown in the paper).

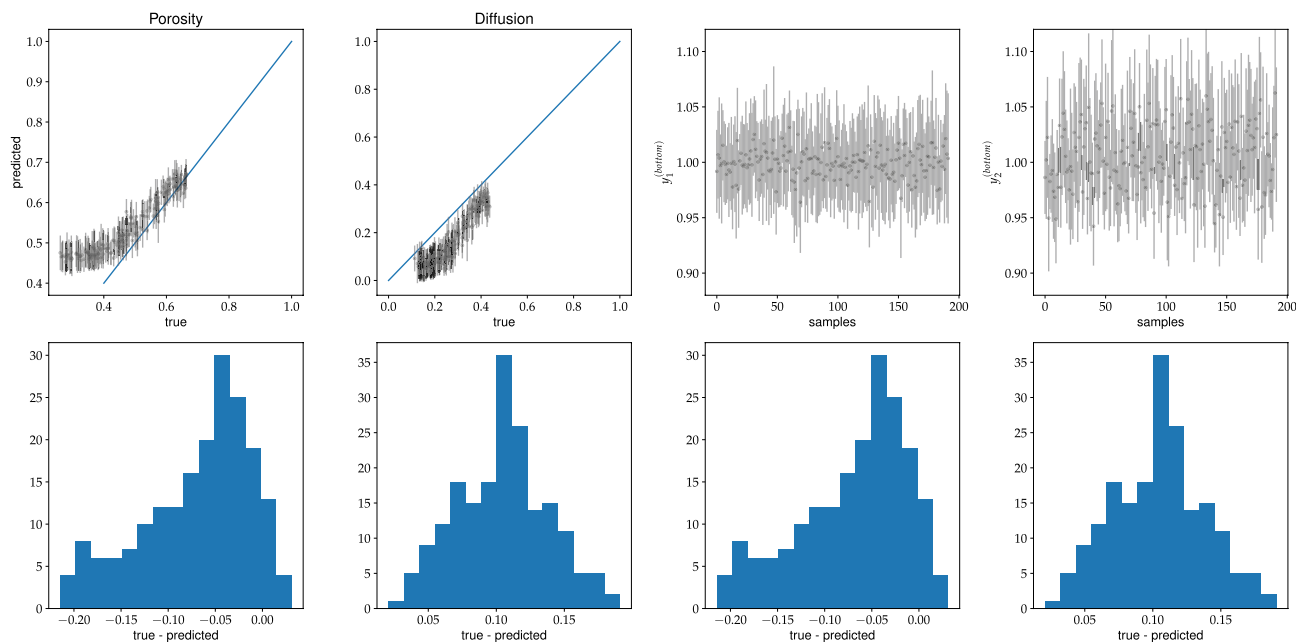
As the final cross-check, we present a plot of a diffusion coefficient as a function of true porosity, see Fig. 11. The data obtained with the LBM and predictions of the two U-Net-Half (with SN module) models trained on type-A and type-B data agree reasonably.

The data show a correlation between the diffusion coefficient and porosity. In Fig. 11, we plot the fits of  $D(\phi)$  as well. To fit the data, we assumed that  $D$  is the reciprocal of tortuosity squared as defined in<sup>36</sup>. Then, we used the Archie's law to express tortuosity,  $\lambda$ :

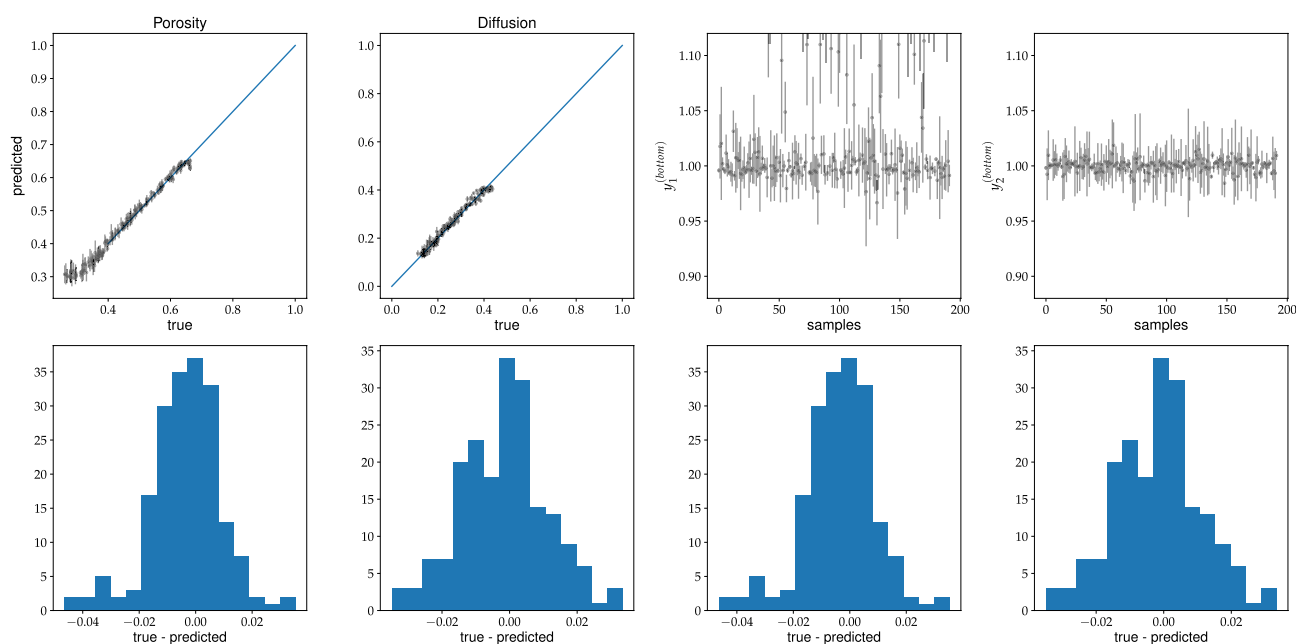


**Figure 8.** The C-Net with SN-module trained on type-A data. In the top row (the first and the second figure from the left) the predicted values of porosity and diffusion coefficient vs. true values for validation type-A data are shown. The figures third and fourth of the top row show the normalization outputs of the SN-module, namely,  $y_1^{bottom}$  and  $y_2^{bottom}$ , see Eq. 7. In the bottom row the corresponding histograms of true-predicted are given. The uncertainties are computed from the MC dropout with  $p = 0.1$ .





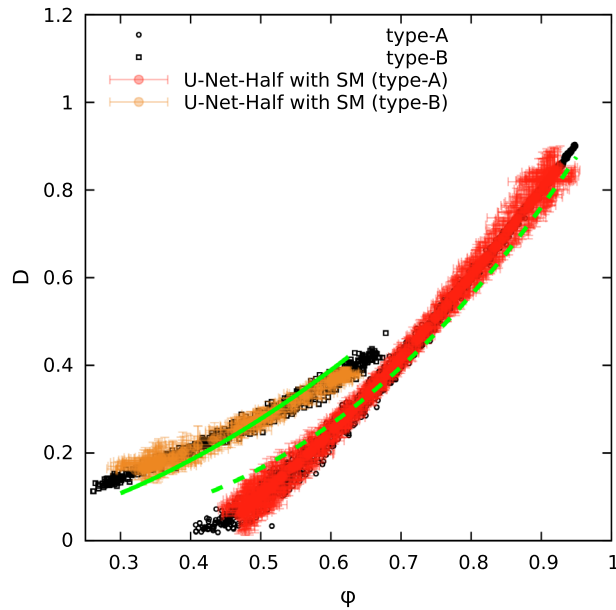
**Figure 9.** Caption the same as for Fig. 8 but the predictions made for validation type-B data.



**Figure 10.** Caption the same as for Fig. 8 but for the U-Net-Half (with SN module) trained on type-B dataset, and the predictions made for validation type-B dataset.

	C-Net with SN	U-Net-Half with SN
MSE (type-A data)	0.73	0.72
MSE (type-B data)	0.33	0.29

**Table 1.** The MSE error computed on validation datasets, for the models run in the Monte Carlo dropout framework.



**Figure 11.** Effective diffusion against true porosity. Open symbols denote the data from LBM solution, filled circles with errorbars denote predictions of the two U-Net-Half (with SN module) networks trained on type-A or type-B datastes. The predictions are made for the validation data sets of type-A or type-B, respectively. Lines (green in color) denote the fits of the effective diffusion as a function of porosity based on Archie law, see Eq. 11.

$$\lambda^2 = \varphi^{1-n}, \quad (11)$$

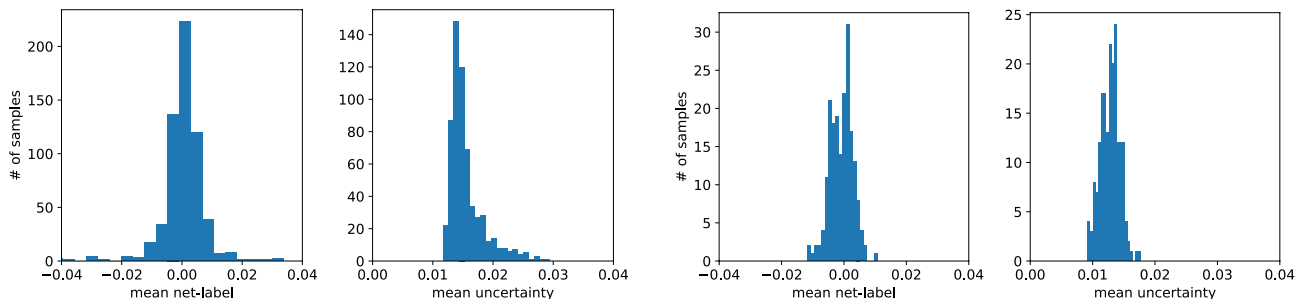
where  $n$  is the Archie's exponent<sup>36</sup>. We found  $n = 3.6$  for type-A data (dashed line) and  $n = 2.8$  for type-B data (solid line), which corresponds well to  $n = 2.14$  reported by Boudreau<sup>36</sup> for fine-grained sediment.

**Reconstruction of the concentrations by U-Net.** This subsection summarizes the results obtained for task II. Our goal was to bring the network, which takes as an input the figure with system geometry configuration, and as an output, it gives the picture with geometry and distribution of concentration. The training data consists of pairs of pictures: input geometry and concentration map.

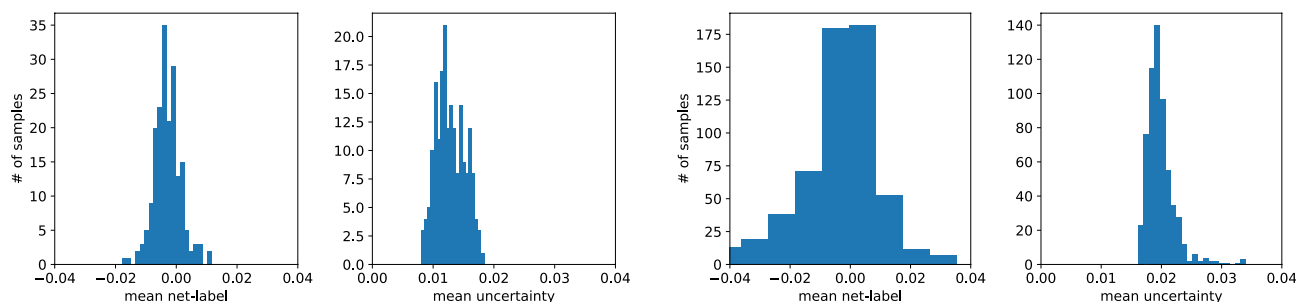
Similarly, as in task I, we consider two scenarios. First, we train and validate the network on type-A data. Then, we test the model on the type-B dataset. Second, we train and validate the network on type-B data and test the model on the type-A dataset. We adopt the MC dropout technique (with  $p = 0.1$ ) to estimate uncertainties as in the previous investigations.

To assess the obtained results for every picture sample, we compute the difference between the input and output pictures and the mean over all nodes. Note that for each site, we obtain  $1\sigma$  uncertainty, and to estimate the average uncertainty for the output picture, we compute the mean over all nodes, see Figs. 12 and 13.

As can be seen, Fig. 12, the U-Net, trained on type-A data, predicts the concentration field well. Unlike task I, the model works well for data type B too. Indeed, in task I, the model trained on type-A data could not predict the porosity and diffusion coefficient for samples with  $\varphi < 0.4$ , which is reached only by type-B samples. Moreover, the U-Net trained on type-B data samples accurately predicts the concentration field for the type-B and type-A data samples, see Fig. 13.

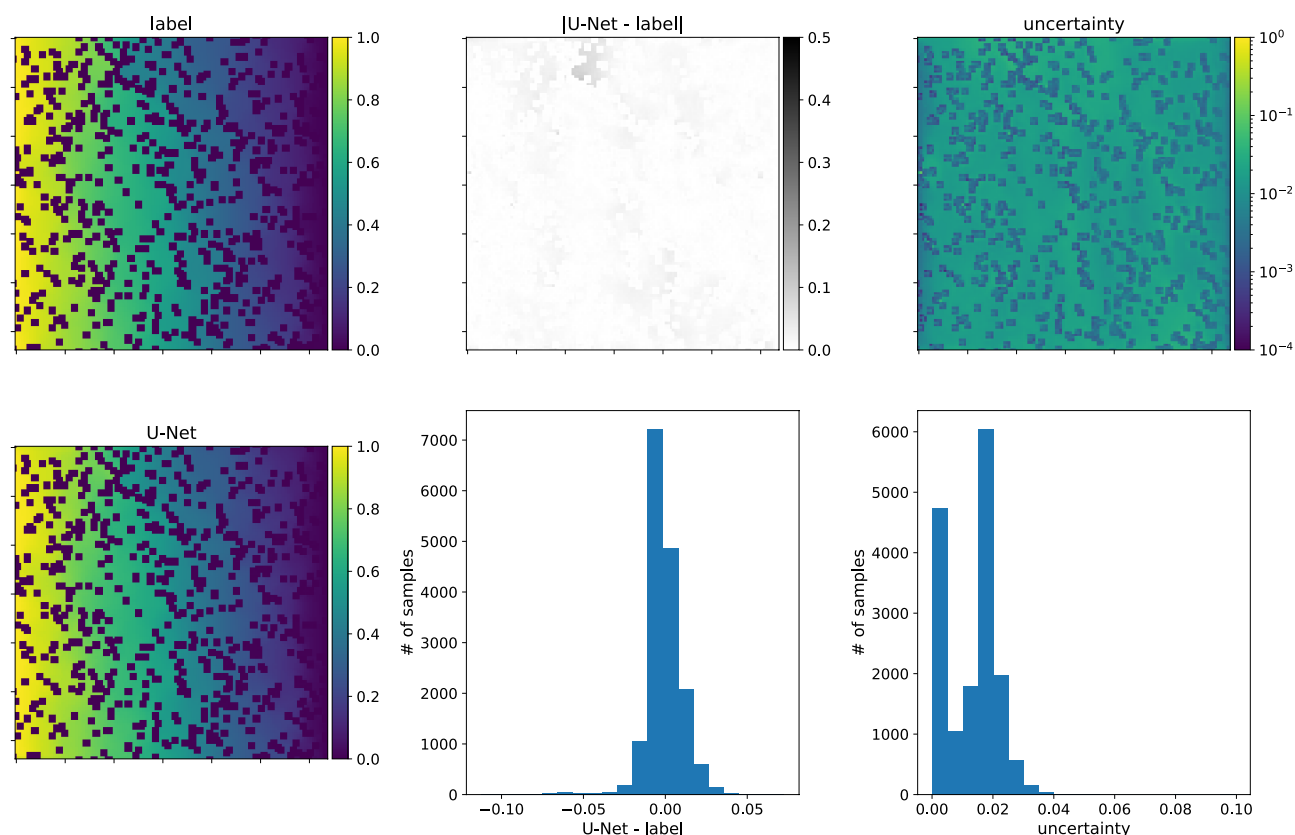


**Figure 12.** The summary of the predictions of concentrations by the U-Net model, trained on type-A data, for the validation datasets of type-A (two figures on the right), and type-B (two figures on the left).

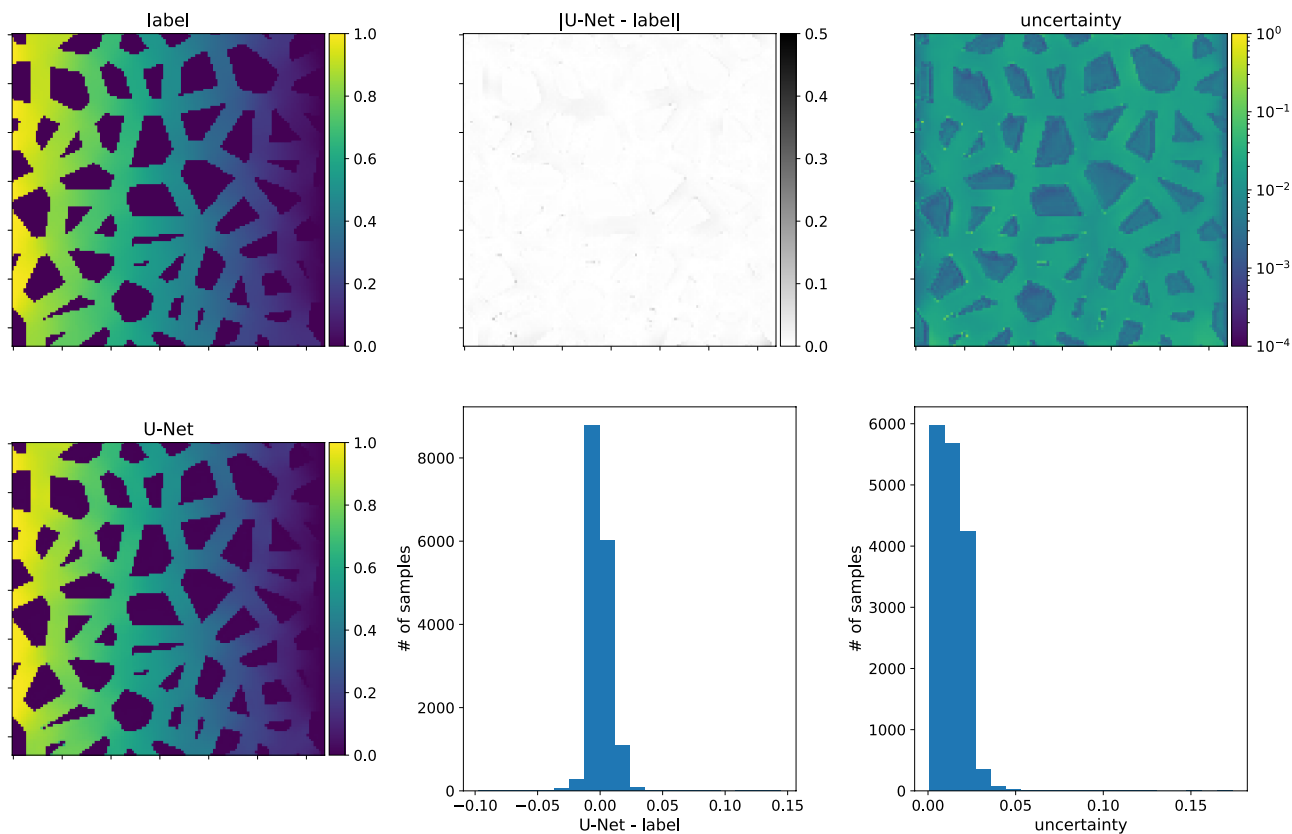


**Figure 13.** The summary of the predictions of concentrations by the U-Net model, trained on type-B data, for the validation datasets of type-B (two figures on the right), and type-A (two figures on the left).

We finish the discussion of the numerical results with the presentation of the U-Nets' performance on chosen samples. Figs. 14 and 15 present results for the network trained on type-A data, its predictions of the concentration field for type-A and type-B data samples, respectively. Figs. 16 and 17 show the analogical results as before but for the network trained on type-B data, and its prediction for type-B and type-A samples, respectively. Each figure contains two rows of plots. The first row consists of a map of the true concentration field, a map of absolute error between true and predicted, and a map of uncertainties. The second row contains a map of the predicted concentration field and histograms of absolute error and uncertainties. One sees that the network prediction error is small for all four cases compared to the maximum concentration value in the LBM solution. The qualitative comparison of mapping obstacles' shapes and positions shows that the networks handle this task well. The error (true-predicted) distributions in each figure are concentrated around the value of zero. For most of the samples, the uncertainty for the network predictions in the void space is more prominent than in the obstacle positions; see Figs. 14, 15, 16, 17.



**Figure 14.** Predictions of the concentrations of the U-Net, trained on type-A data, for type-A input sample. Top row, from left to right: the input figure, the differences between the input and the network's response and a map of uncertainties for each node. Bottom row, from left to right: the network prediction, histogram of differences between input and network's outcome histogram of uncertainties at each node. The direction of transport is from the left to the right. Note that the smallest errors are obtained for the nodes where obstacles are placed, while in the pore space, the uncertainties are more considerable but uniformly distributed.



**Figure 15.** The same as Fig. 14 but for the model trained on type-A data, and the predictions made for type-B input sample.

## Summary

We have developed the DL models to predict the basic properties of diffusion in the porous media. We considered two different types of CNNs. Both take as the input the picture of the system's geometry, but one predicts the porosity and diffusion coefficient, while the other reconstructs the systems' geometry and concentration map.

For the first task, we discussed two variants of the CNN: the C-Net and the U-Net-Half. Both are accompanied by the self-normalized module proposed by Graczyk et al.<sup>29</sup>. We show that the models with the SN module have better performance than those without the module. For the second task, we proposed to consider the U-Net architecture. The model perfectly reconstructs the concentration field for all types of data.

We have considered two types of data: one mimics the sand packings, and the other mimics the systems derived from the extracellular space of biological tissues. One of our goals was to verify whether the network, trained on one type of data, can be used for analysis in the case of the other and conversely. The models that predict the porosity and diffusion coefficient work well only on the data type they were trained on. The model's porosity and diffusion coefficient predictions are within  $1\sigma$  confidence level bounds.

In contrast to the task I, the models developed for task II (reconstruction of the concentration map) work well for the data type that was not included in the training process. The best accuracy in the reconstruction of the concentration maps is achieved for type-B data, namely, systems derived from the extracellular space of biological tissues. In this case, the mean error is smaller than 1% while the mean map reconstruction uncertainty is slightly larger.

## Data availability

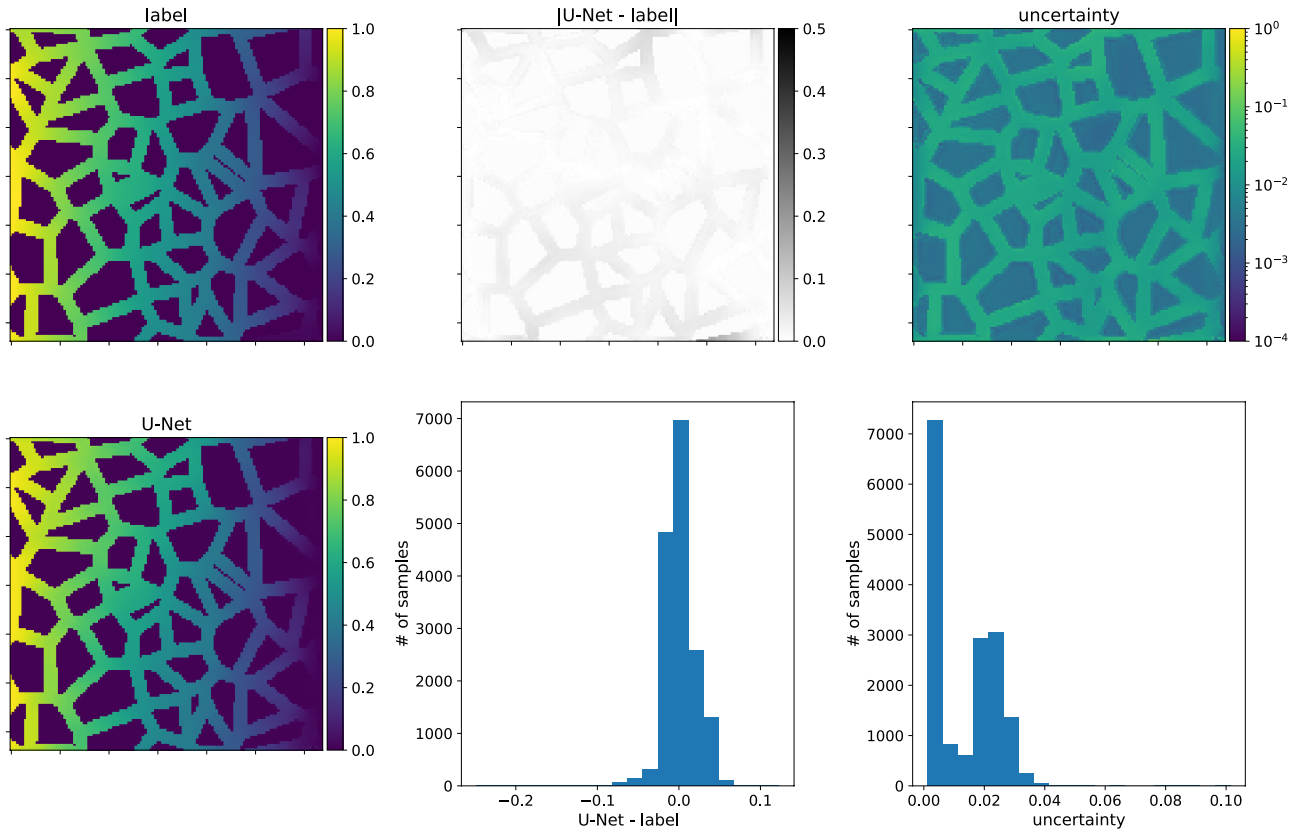
The dataset used during the current study is available on request, please contact Krzysztof Graczyk.

## Lattice Boltzmann method

Lattice Boltzmann Method<sup>37,38</sup> is a numerical tool for solving discrete Boltzmann transport equation:

$$\frac{\partial f_k}{\partial t} + \vec{\xi}_k \cdot \nabla f_k = \Omega(f_k^{\text{eq}}, f_k); \quad f_k = f_k(\vec{r}, t) \quad (12)$$

where  $f_k$  is the distribution function discretized along  $k$ -th population velocity and  $\Omega$  is collision operator. Depending on the set of discretized velocities  $\xi_k$  and the used equilibrium functions  $f_k^{\text{eq}}$ , certain moments of the equation's solution are also the solution of macroscopic transport equations. For example, in two dimensional space ( $\vec{r} \in \mathbb{R}^2$ ) with the set of nine discrete velocities:



**Figure 16.** The same as for Fig. 14 but for the model trained on type-B data, and the predictions made for type-B input sample.

$$\begin{aligned}
 \xi_0 &= [0, 0]^T & \xi_1 &= [1, 0]^T & \xi_5 &= [1, 1]^T \\
 \xi_2 &= [0, 1]^T & \xi_6 &= [-1, 1]^T & & \\
 \xi_3 &= [-1, 0]^T & \xi_7 &= [-1, -1]^T & & \\
 \xi_4 &= [0, -1]^T & \xi_8 &= [1, -1]^T & & 
 \end{aligned}
 \tag{13}$$

and equilibrium distribution function:

$$f_k^{\text{eq}} = \omega_k c, \quad \text{where } \omega_0 = 4/9, \quad \omega_{1,2,3,4} = 1/9, \quad \omega_{5,6,7,8} = 1/36,
 \tag{14}$$

one obtains the concentration field of diffusing molecules (solution of Eq. (2)) as the zeroth moment of the distribution function:

$$c(\vec{r}, t) = \sum_{k=0}^8 f_k(\vec{r}, t)
 \tag{15}$$

In our analysis we use the set of discrete velocities from Eq. (13), equilibrium distributions given in Eq. (14) and the collision operator introduced by Bhatnagar, Gross and Krook<sup>39</sup> (BGK collision):

$$\Omega_{\text{BGK}}(f_k^{\text{eq}}, f_k) = \frac{1}{\tau} (f_k^{\text{eq}} - f_k),
 \tag{16}$$

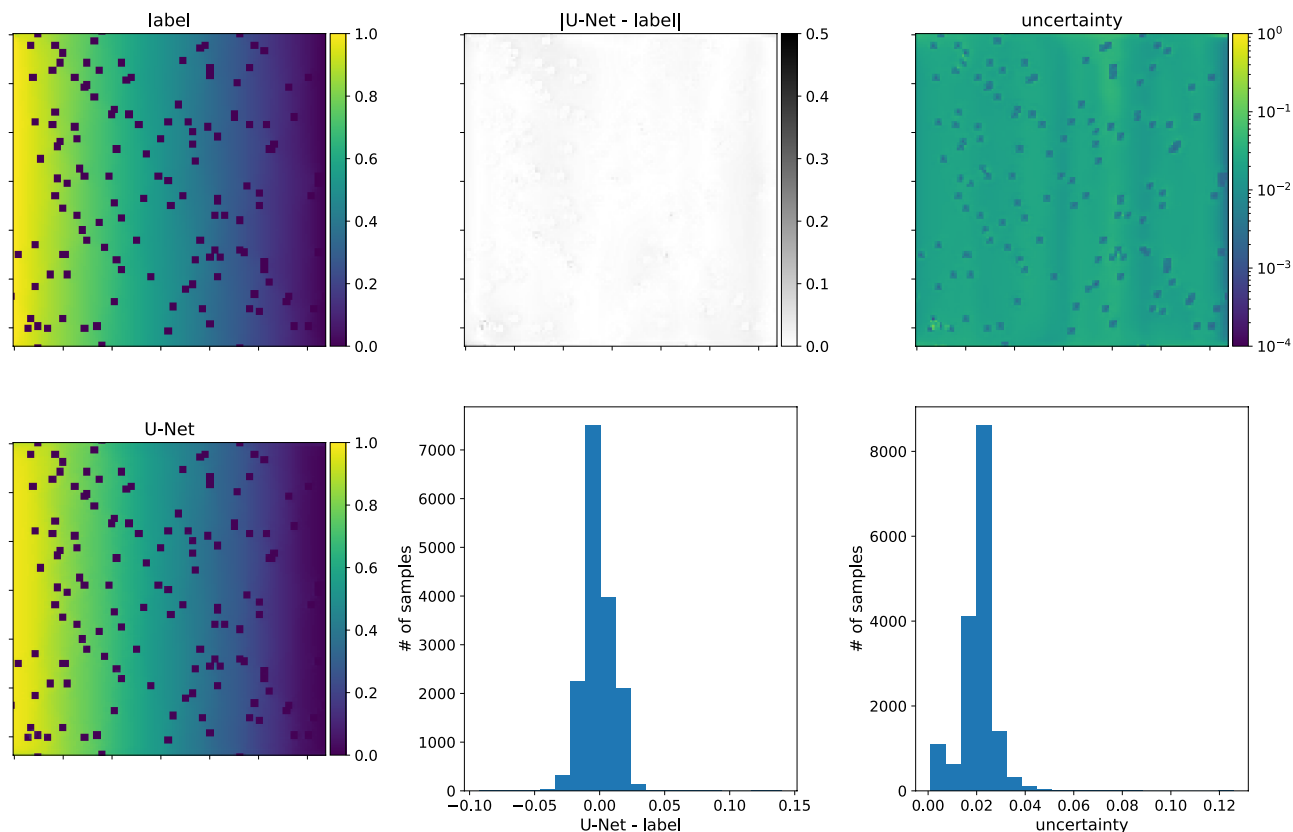
where  $\tau$  is relaxation time of a distribution function to its equilibrium. The bulk diffusion coefficient of LBM model equals then

$$D_{0,\text{LBM}} = \frac{1}{3} \left( \tau - \frac{1}{2} \right).
 \tag{17}$$

We use above quantity to calculate macroscopic diffusion coefficient of the solved diffusion equation:

$$D_0 = D_{0,\text{LBM}} \frac{\delta x^2}{\delta t},
 \tag{18}$$

where  $\delta x$  [m] and  $\delta t$  [s] are LBM physical space and time discretization parameters, respectively.



**Figure 17.** The same as for Fig. 14 but for the model trained on type-B data, and the predictions made for type-A input sample.

We solve the discrete Boltzmann equation with lagrangian transport, in which post-collision populations  $f_k^*$  are streamed from departure nodes  $\vec{r} - \vec{\xi}_k \delta x$  to arrival nodes  $\vec{r}$ :

$$f_k(\vec{r}, t + 1) = \underbrace{f_k^*(\vec{r} - \vec{\xi}_k, t)}_{\text{streaming}} = \underbrace{f_k(\vec{r} - \vec{\xi}_k, t) + \frac{1}{\tau} (f_k^{\text{eq}}(\vec{r} - \vec{\xi}_k, t) - f_k(\vec{r} - \vec{\xi}_k, t))}_{\text{collision}}. \quad (19)$$

Eventually, we assume reflecting boundary conditions on obstacles surfaces and on top and bottom boundary. It was realised with bounceback scheme:

$$f_k(\vec{r}, t + 1) = f_{k'}^*(\vec{r}, t) \quad \text{when} \quad \vec{r} + \vec{\xi}_{k'} \in \mathcal{D}_s \quad (20)$$

where  $k'$  denotes lattice vector opposite to  $k$ , i.e.  $\vec{\xi}_{k'} = -\vec{\xi}_k$ . On left and right boundaries (inlet and outlet, respectively) constant concentrations were assumed (Dirichlet boundary conditions) equal to  $c([0, y], t) = c_{\text{in}} = 1$  and  $c([L, y], t) = c_{\text{out}} = 0$  which are realized by setting the unknown populations to:

$$f_k = \left( c_B - \sum_{i \in k_{\mathcal{D}}} f_i \right) \frac{\omega_k}{\sum_{i \in k_{\mathcal{D}'}} \omega_i} \quad \text{for} \quad \vec{r} + \vec{\xi}_{k'} \in \mathcal{D}', \quad (21)$$

where  $c_B$  denotes the prescribed concentration at the boundary (0 or 1) and summation over  $k_{\mathcal{D}}$  and  $k_{\mathcal{D}'}$  denotes summing over lattice neighbors laying inside and outside of domain  $\mathcal{D}$ , respectively. Thus, Eq. (21) amounts to distributing the concentration from outside of  $\mathcal{D}$  according to lattice weights  $\omega_k$  of the nodes laying outside of  $\mathcal{D}$ .

Received: 11 April 2023; Accepted: 4 June 2023

Published online: 16 June 2023

## References

- Bell, J. & Grosberg, P. Diffusion through porous materials. *Nature* **189**, 980–981 (1961).
- Shen, L. & Chen, Z. Critical review of the impact of tortuosity on diffusion. *Chem. Eng. Sci.* **62**, 3748–3755 (2007).
- Kuhn, T. *et al.* Single-molecule tracking of nodal and lefty in live zebrafish embryos supports hindered diffusion model. *Nat. Commun.* **13**, 1–15 (2022).
- Muñoz-Gil, G. *et al.* Objective comparison of methods to decode anomalous diffusion. *Nat. Commun.* **12**, 6253. <https://doi.org/10.1038/s41467-021-26320-w> (2021).

5. Syková, E. & Nicholson, C. Diffusion in brain extracellular space. *Physiol. Rev.* **88**, 1277–1340 (2008).
6. Nicholson, C. Diffusion and related transport mechanisms in brain tissue. *Rep. Prog. Phys.* **64**, 815 (2001).
7. Postnikov, E. B., Lavrova, A. I. & Postnov, D. E. Transport in the brain extracellular space: Diffusion, but which kind?. *Int. J. Mol. Sci.* **23**, 12401 (2022).
8. Tartakovsky, D. M. & Dentz, M. Diffusion in porous media: Phenomena and mechanisms. *Transp. Porous Media* **130**, 105–127 (2019).
9. Chen, K. C. & Nicholson, C. Changes in brain cell shape create residual extracellular space volume and explain tortuosity behavior during osmotic challenge. *Proc. Natl. Acad. Sci.* **97**, 8306–8311 (2000).
10. Weber, R. M., Korneev, S. & Battiato, I. Homogenization-informed convolutional neural networks for estimation of li-ion battery effective properties. *Transport in Porous Media* 1–22 (2022).
11. Wernert, V. et al. Tortuosity of hierarchical porous materials: Diffusion experiments and random walk simulations. *Chem. Eng. Sci.* **264**, 118136 (2022).
12. Li, H., Li, H., Gao, B., Wang, W. & Liu, C. Study on pore characteristics and microstructure of sandstones with different grain sizes. *J. Appl. Geophys.* **136**, 364–371. <https://doi.org/10.1016/j.jappgeo.2016.11.015> (2017).
13. Kinney, J. P. et al. Extracellular sheets and tunnels modulate glutamate diffusion in hippocampal neuropil. *J. Compar. Neurol.* **521**, 448–464 (2013).
14. Godin, A. G. et al. Single-nanotube tracking reveals the nanoscale organization of the extracellular space in the live brain. *Nat. Nanotechnol.* **12**, 238–243. <https://doi.org/10.1038/nnano.2016.248> (2017).
15. Tartakovsky, A. M., Tartakovsky, D. M. & Meakin, P. Stochastic langevin model for flow and transport in porous media. *Phys. Rev. Lett.* **101**, 044502. <https://doi.org/10.1103/PhysRevLett.101.044502> (2008).
16. Kalz, E. et al. Collisions enhance self-diffusion in odd-diffusive systems. *Phys. Rev. Lett.* **129**, 090601. <https://doi.org/10.1103/PhysRevLett.129.090601> (2022).
17. Alexandre, A., Mangeat, M., Guérin, T. & Dean, D. S. How stickiness can speed up diffusion in confined systems. *Phys. Rev. Lett.* **128**, 210601. <https://doi.org/10.1103/PhysRevLett.128.210601> (2022).
18. Kasthuri, N. et al. Saturated reconstruction of a volume of neocortex. *Cell* **162**, 648–661. <https://doi.org/10.1016/j.cell.2015.06.054> (2015).
19. Kamrava, S., Tahmasebi, P. & Sahimi, M. Linking morphology of porous media to their macroscopic permeability by deep learning. *Trans. Porous Media* <https://doi.org/10.1007/s11242-019-01352-5> (2020).
20. Santos, J. E. et al. Computationally efficient multiscale neural networks applied to fluid flow in complex 3d porous media. *Transp. Porous Media* **140**, 241–272 (2021).
21. Graczyk, K. M. & Matyka, M. Predicting porosity, permeability, and tortuosity of porous media from images by deep learning. *Sci. Rep.* **10**, 1–11 (2020).
22. Cawte, T. & Bazylak, A. Accurately predicting transport properties of porous fibrous materials by machine learning methods. *Electrochem. Sci. Adv.* e2100185 (2022).
23. Ranjan Sethi, S., Kumawat, V. & Ganguly, S. Convolutional neural network based prediction of effective diffusivity from microscope images. *J. Appl. Phys.* **131**, 214901 (2022).
24. Rödning, M., Wählstrand Skärström, V. & Lorén, N. Inverse design of anisotropic spinodoid materials with prescribed diffusivity. *Sci. Rep.* **12**, 17413. <https://doi.org/10.1038/s41598-022-21451-6> (2022).
25. Kamrava, S., Tahmasebi, P. & Sahimi, M. Physics- and image-based prediction of fluid flow and transport in complex porous membranes and materials by deep learning. *J. Membr. Sci.* **622**, 119050. <https://doi.org/10.1016/j.memsci.2021.119050> (2021).
26. Wu, J., Yin, X. & Xiao, H. Seeing permeability from images: Fast prediction with convolutional neural networks. *Sci. Bull.* **63**, 1215–1222. <https://doi.org/10.1016/j.scib.2018.08.006> (2018).
27. Kamrava, S., Sahimi, M. & Tahmasebi, P. Simulating fluid flow in complex porous materials by integrating the governing equations with deep-layered machines. *npj Comput. Mater.* **7**, 127. <https://doi.org/10.1038/s41524-021-00598-2> (2021).
28. Tahmasebi, P., Kamrava, S., Bai, T. & Sahimi, M. Machine learning in geo- and environmental sciences: From small to large scale. *Adv. Water Resour.* **142**, 103619. <https://doi.org/10.1016/j.advwatres.2020.103619> (2020).
29. Graczyk, K. M., Pawłowski, J., Majchrowska, S. & Golan, T. Self-normalized density map (sndm) for counting microbiological objects. *Sci. Rep.* **12**, 10583. <https://doi.org/10.1038/s41598-022-14879-3> (2022).
30. Ronneberger, O., Fischer, P. & Brox, T. U-net: convolutional networks for biomedical image segmentation (2015). [arXiv:1505.04597](https://arxiv.org/abs/1505.04597).
31. Gal, Y. & Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning (2015). [arXiv:1506.02142](https://arxiv.org/abs/1506.02142).
32. Wu, H., Fang, W.-Z., Kang, Q., Tao, W.-Q. & Qiao, R. Predicting effective diffusivity of porous media from images by deep learning. *Sci. Rep.* **9**, 20387. <https://doi.org/10.1038/s41598-019-56309-x> (2019).
33. Paszke, A. et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems* 32, 8024–8035 (Curran Associates, Inc., 2019).
34. Gal, Y. & Ghahramani, Z. Dropout as a bayesian approximation: Appendix (2015). [arXiv:1506.02157](https://arxiv.org/abs/1506.02157).
35. Koza, Z., Kondrat, G. & Suszczyński, K. Percolation of overlapping squares or cubes on a lattice. *J. Stat. Mech. Theory Exp.* **2014**, P11005 (2014).
36. Boudreau, B. P. The diffusive tortuosity of fine-grained un lithified sediments. *Geochim. Cosmochim. Acta* **60**, 3139–3142. [https://doi.org/10.1016/0016-7037\(96\)00158-5](https://doi.org/10.1016/0016-7037(96)00158-5) (1996).
37. Krüger, T. et al. *The Lattice Boltzmann Method - Principles and Practice* (Springer Cham, 2016).
38. Succi, S. *The lattice boltzmann equation: For complex states of flowing matter* (Oxford University Press, 2018).
39. Bhatnagar, P. L., Gross, E. P. & Krook, M. A Model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems. *Phys. Rev.* **94**, 511–525. <https://doi.org/10.1103/PhysRev.94.511> (1954).

## Acknowledgements

D.S. and M.M supported by National Science Centre. Funded by National Science Centre, Poland under the OPUS call in the Weave programme 2021/43/I/ST3/00228. K.M.G. has been partly supported by the program "Excellence initiative - research university" for the years 2020-2026 for the University of Wrocław. The publication was partially financed by the Initiative Excellence-Research University program for University of Wrocław. This research was funded in whole or in part by National Science Centre (2021/43/I/ST3/00228). For the purpose of Open Access, the author has applied a CC-BY public copyright licence to any Author Accepted Manuscript (AAM) version arising from this submission.

## Author contributions

K.M.G. D.S. and M.M. conceived the project. K.M.G. designed and performed the deep learning analysis and proposed usage of the CNN with the Self-Normalized module and U-Net architectures. D.S. designed and

performed diffusion simulation within LBM and prepared the data. All authors participated in writing the text and discussing the results. MM found the exponents in Archie's law.

### Competing Interests

The authors declare no competing interests.

### Additional information

**Correspondence** and requests for materials should be addressed to K.M.G.

**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 licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence 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 licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2023