



# ML-LBM: Predicting and Accelerating Steady State Flow Simulation in Porous Media with Convolutional Neural Networks

Ying Da Wang<sup>1</sup> · Traiwit Chung<sup>1</sup> · Ryan T. Armstrong<sup>1</sup> · Peyman Mostaghimi<sup>1</sup> 

Received: 8 October 2020 / Accepted: 29 March 2021 / Published online: 23 April 2021  
© The Author(s), under exclusive licence to Springer Nature B.V. 2021

## Abstract

Fluid mechanics simulation of steady state flow in complex geometries has many applications, from the micro-scale (cell membranes, filters, rocks) to macro-scale (groundwater, hydrocarbon reservoirs, and geothermal) and beyond. Direct simulation of steady state flow in such porous media requires significant computational resources to solve within reasonable timeframes. This study outlines an integrated method combining predictions of fluid flow (fast, limited accuracy) with direct flow simulation (slow, high accuracy) is outlined that reduces computation time by an order of magnitude without loss of accuracy. A convolutional neural network (CNNs) is trained with various configurations on simulations in 2D and 3D porous media to estimate steady state velocity fields. Permeability estimation (as an average of the field) is accurate, but the velocity fields themselves are error prone, unsuitable for further transport studies. This estimate can either be used as an indicative prediction, or as initial conditions in direct simulation to reach a fully accurate result in a fraction of the compute time. Using Deep Learning predictions (or potentially any other approximation method) to accelerate flow simulation to steady state in complex structures shows promise as a technique to push the boundaries fluid flow modelling.

## Article Highlights

- **Steady State velocity fields predicted in 2D and 3D using CNNs**
- **Permeability estimation with predicted fields over 95% accurate in most cases**
- **Fine scale velocity field prediction is error-prone, limited by CNN performance**

---

✉ Peyman Mostaghimi  
peyman@unsw.edu.au

Ying Da Wang  
yingda.wang@unsw.edu.au

Traiwit Chung  
traiwit.chung@unsw.edu.au

Ryan T. Armstrong  
ryan.armstrong@unsw.edu.au

<sup>1</sup> School of Minerals and Energy Resources Engineering, University of New South Wales, Sydney, NSW 2052, Australia

- **Fast, low accuracy CNN prediction is combined with slow, high accuracy simulation**
- **Accelerated technique produces fully accurate results in 10x less time**

**Keywords** Porous media · Convolutional neural networks · Flow simulation · Steady state · Fluid mechanics

## 1 Introduction

The flow of fluid within the void space of porous structures is a physical phenomenon that is pervasive in its occurrence, with applications in catalysis (Keil and Rieckmann 1994), groundwater hydrogeology, oil and gas extraction, environmental waste management (Fenwick and Blunt 1998; Hilpert and Miller 2001; Blunt et al. 2002; Culligan et al. 2006; Mostaghimi and Mahani 2010; Mostaghimi et al. 2016; Blunt 2017), carbon capture storage (Blunt et al. 2013), natural and biological membranes (Gruber et al. 2011), and medical applications (Khanafar et al. 2012). These examples highlight the need to accurately capture the physics of fluid flow within porous media.

Direct flow simulation by solving the Navier–Stokes equation (NVE) explicitly offers the highest level of accuracy with the finest degree of detail, projecting directly into the domain. This can be performed by finite method solutions of the NVE (White et al. 2006; Sun et al. 2010; Mahbub et al. 2020), or by Lattice Boltzmann Methods (LBM) that also solve the NVE (McClure et al. 2014; Wang et al. 2020). These techniques require significant compute time and memory. The computational issues with these methods arise due to them being (a) time-dependent, and/or (b) nonlinear/iterative. LBM is time-dependent, NVE is both time-dependent and nonlinear, and the simplified steady state stokes equation (Mostaghimi et al. 2013) is iterative. In this study the term “steady-state” refers to the point in which flow simulation converges such that time-stepping and/or iterations become static. Furthermore, in porous media where steady state solutions are sought after, the convergence rate of direct flow simulation to these steady state conditions is also dependent on the geometric complexity of the porous media.

From steady state flow simulation in porous media, velocity fields provide necessary information for the determination of critical parameters and phenomena. Of particular interest in this study are (a) the permeability, obtained from averaging the velocity fields, and b) the solute transport profile, a fine-scale value obtained by further simulation on top of the velocity fields. Permeability describes the average flow potential of a porous media, an intrinsic property of the geometry that dictates the relationship between flow and pressure, proportional to the average velocity within the system. It is used in the upscaling of the NVE in the pore-scale (typically in the mm to  $\mu\text{m}$  range), to core-scale (cm) and reservoir-scale (m) flow equations based on Darcy’s Law. As a bulk property, permeability is relatively insensitive to minor errors and approximations in the velocity field, evident in the reasonable accuracy achieved by Pore Network Models (PNM) (Blunt et al. 2002; Dong et al. 2008; Rabbani and Babaei 2019) and semi-analytical solvers (SAS) (Chung et al. 2019; Wang et al. 2019, 2020; Torskaya et al. 2014; Shabro et al. 2012) that estimate flow using geometric simplifications. However, the actual fine-scale velocity fields in the principle axes are also important in these applications, for example, to characterise the

transport of solute within the fluids (Mostaghimi et al. 2016; Liu et al. 2017, 2018; Liu and Mostaghimi 2017).

## 1.1 Related Work

In the specific task of predicting fluid flow and fluid flow properties in porous media, CNNs have been applied to reasonable success but all have suffered from a loss in accuracy, rendering them acceptable for rough estimation only (Wang et al. 2021). These can be categorised into a) the prediction of bulk properties by Regression, and b) the prediction of fine-scale fields by image-to-image translation.

The regression of physical properties and fluid flow properties in porous media has focused on using machine learning and has found success in the prediction of porous media properties (Sudakov et al. 2019; Erofeev et al. 2019; Alqahtani et al. 2020, 2021; Kamrava et al. 2019; Rabbani and Babaei 2019; Ahmadi et al. 2013; Tian et al. 2020), including flow, permeability, porosity, surface area, and other morphological parameters. These methods have shown acceptable accuracy competitive with traditional approximation techniques for a fraction of the computational cost. This is because the prediction of bulk properties lends itself reasonably well to approximation, with parameters such as the permeability naturally incurring uncertainty in its measurement (Chappell and Lancaster 2007).

Prediction of fine-scale fields have been applied to predict velocity fields in a number of different applications, including Magnetohydrodynamics (Van Oort et al. 2019), Darcy-scale Reservoir Simulation (Wang and Lin 2020), steady state flow in simple geometries (Guo et al. 2016) and PNMs (Rabbani and Babaei 2019), and steady state flow in 3D porous media (Santos et al. 2020). In some cases, the network is trained to simply advance a simulation forward (Hennigh 2017; Wang and Lin 2020), while in others, the steady state scalar fields are predicted in a single pass of the network (Santos et al. 2020; Guo et al. 2016; Jin et al. 2018). The accuracy achieved in these examples has been shown to be quite high in cases of simple geometries such as vehicle profiles and single objects with simple geometries, though visual discrepancies in the form of noise and velocity profile errors are common. In the few cases studying porous media, the influence that the geometric complexity of the domain (porous or otherwise) has on the accuracy of the network is underexplored. Furthermore, while accuracies are reported as high in terms of the standard pixel/voxelwise measures, the permeability accuracy, and visual comparison, the true usability of these velocity fields for anything more than a visual approximation remains an open question. In cases where the velocity field is predicted as a scalar field, this is more-so the case (Santos et al. 2020). If the network predicts timesteps, errors in the predictions are likely to build up if not corrected for in some manner (Wang and Lin 2020), while networks trained to directly predict the steady state configuration from only geometric data are prone to higher errors simply due to the extra complexity of the mapping (Santos et al. 2020). These errors may be acceptable when evaluating bulk properties such as the permeability, but may significantly impact usefulness of these predictions if the fine-scale detail is required. In this sense, predictive methods that fall back on the original direct algorithm in a self correcting manner are most effective in preserving accuracy while improving efficiency (Hennigh 2017; Wang et al. 2020). In the realm of fluid flow in any media of any geometry, solute transport phenomena is one of the most critical simulations that is influenced by the accuracy of the underlying velocity field.

An important aspect to emphasise in this study is the use of feature maps as a part of the input to the network, which is known as Feature Engineering (using encoded

features maps as input). In the case of velocity field estimation, feature engineering has shown good results when the trained network is utilised in a patch-based method of prediction (Santos et al. 2020) despite each patch having no explicit data regarding the geometry of other regions of the domain. Also, the simple existence of multiple possible flow paths is difficult for a CNN to handle. In the context of velocity field estimation, this local-to-global problem when applying patch-based methods by its nature requires some form of information, in this case, the time-of-flight (TOF) (Hassouna and Farag 2007), which is analogous to the local tortuosity of the geometry (Øren and Bakke 2002). Without this information, networks trained to predict velocity fields must always be utilised in their global format, which while facilitated by designing the CNN to be fully convolutional (able to be used on any size), can incur higher than acceptable computational cost during prediction. On the other hand, it should be mentioned that calculation of the TOF using the Efficient Fast Marching method scales less favourably to the calculation of the SAS flow approximation or the calculation of Laplace tortuosity (which utilise the exact same discrete mathematics). Both of these Laplace methods scale  $O[N]$  linearly (Chung et al. 2019; Øren and Bakke 2002) compared to TOF which scales  $O[N \log N]$  (Hassouna and Farag 2007). In this case, the SAS Laplace Flow approximation would be the best option as it provides a close, fine-scale approximation to the velocity fields (Shabro et al. 2012; Chung et al. 2019). The problem therein lies, if SAS approximations are already quite good, the CNN plays an increasingly lesser role in translating input to output. In the most effective layout, the SAS approximation being transformed to the true LBM approximation is likely the best approach from a Feature Engineering perspective. Since this study also introduces an acceleration procedure for correcting errors in approximated velocity fields, it can also be argued that an SAS approximation can be directly corrected by the acceleration procedure used in this study.

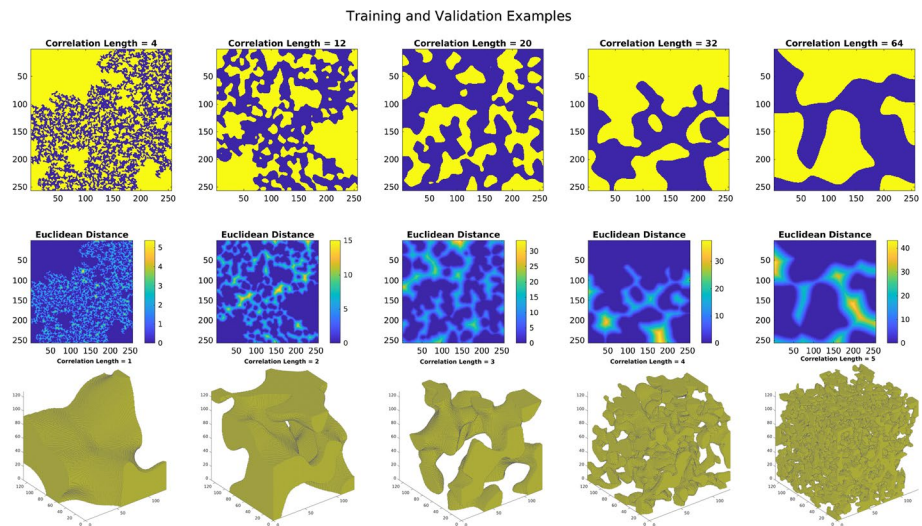
This study focuses on determining and surpassing limitations in accuracy achievable by end-to-end CNN based image-to-image translation networks on the problem of predicting velocity fields in all principal axes. Specifically, the permeability accuracy and the solute transport accuracy are examined, as one is insensitive to fine-scale fluctuations while the other is highly susceptible. These errors that occur due to the minimisation problem are then corrected for by an accelerated direct simulation using these approximations as input. Porous media is generated in 2D and 3D using correlated fields, and the steady state velocity fields are obtained by Multiple Relaxation Time Lattice Boltzmann Method (MRT-LBM) simulations (McClure et al. 2014). The networks are trained first in an end-to-end format, with only the binary geometry transforming to the predicted velocity field. This is followed by modifications to the input as a euclidean distance map, the addition of a custom mass conservation function, the removal of biases, and alterations to the aspect ratio of the network (by inversely varying the convolutional filters and kernels). The network achieves an accurate estimate of the steady state velocity fields, measured by permeability and L2 error. Network accuracy is shown to be dependent primarily on the tortuosity (geometric complexity) of the domain. Permeability estimation from these predicted fields reaches over 90% accuracy for 80% of cases, but fine-scale velocity fields are error prone when used for solute transport simulation. Using the predicted velocity fields as initial conditions is shown to accelerate direct flow simulation to steady state conditions with an order of magnitude less compute time. Using Deep Learning to accelerate flow simulation to steady state in complex pore structures shows promise as a technique push the boundaries fluid flow modelling.

## 2 Materials and Methods

### 2.1 Datasets

To generalise the findings in this study, porous media is generated in a manner resulting in a uniform distribution of geometric complexities. 10,000 2D correlated fields are generated stochastically, which emulates the internal structure of porous media (Liu and Mostaghimi 2017). These correlated fields are generated in a similar fashion to other such synthetically generated procedures. The algorithm follows a process that involves (1) generating a random field of uniformly distributed numbers, (2) applying a Gaussian blurring kernel to the field with a kernel size (or correlation length) chosen in relation to the desired pore channel size, (3) transforming the field into a uniform distribution  $\in [0, 1]$ , and (4) selecting a threshold value to segment the image. In the case of this dataset,  $256 \times 256$  domains are generated with a correlation length between 8 and 64, whereby the segmentation threshold  $T$  is chosen as the first value  $T_i$  that allows the domain to be connected from the prescribed inlet to outlet plus a small value  $\epsilon$  to allow reasonably sized minimum throat diameters in the generated domains. In this dataset, this value  $\epsilon$  is equal to 0.03. Some examples are shown in Fig. 1. Overall, the dataset is designed to represent a diverse array of geometries, ranging from simple wide channels to tight, complex flow paths. The permeability distribution of the entire dataset spans 5 orders of magnitude.

In 3D applications, the simulation domain can exceed  $1000^3$  voxels, which is outside the feasible realm of Deep Learning, which requires relatively smaller domain sizes to maintain fast iterations during the training process. Typically, volumes smaller than  $100^3$  are used for training 3D networks (Wang et al. 2020; Santos et al. 2020; Guo et al. 2016; Wang et al. 2020), since GPUs are limited in available memory, and computational cost scales poorly in 3D. For reference, training the Resnet-like network on a batch size of  $4-8 \times 64 \times 64 \times 64$  will fit inside 8 GB of GPU memory (Alqahtani et al. 2021), which scales poorly to other GPUs. An Nvidia RTX Titan contains 24 GB of memory, while a Quadro RTX 8000 contains 48 GB



**Fig. 1** Example 2D and 3D images of porous media generated from correlated fields with varying correlation length, and the Euclidean Distance Transform

of memory. Doubling the domain size to  $128 \times 128 \times 128$  for the same network would incur 64 GB of memory on a single GPU. Even a single  $256^3$  batch image would not fit on a single GPU. To circumvent this, patch based learning can be applied (Santos et al. 2020), or a fully convolutional network can be trained, allowing the network to be used on domains of arbitrary size, or distributed training on GPU clusters can be performed (though the size would then be limited based on a batch size of 1). In this case, 1000 3D images of segmented correlated fields of size  $128^3$  are also generated for 3D neural network training, using the same methodology as the generation of 2D porous media.

All datasets are generated with both a binary image as input as well as the Euclidean Distance Transform (EDT) of the pore space. Major factors that influence flow fields such as (a) number of connected flow paths, (b) overall tortuosity, and (c) minimum throat radius have significant effects on simulation time and stability. Of these parameters, the minimum throat radius is the predominant factor that causes numerical instability and small time-stepping, resulting in slow computation towards steady state conditions. There are numerous examples within this dataset of tortuous and tight domains where the principal flow path contains a throat with 3–5 voxels in diameter, causing a significant increase in simulation time. One of the objectives of CNN based velocity field estimation is the bypassing of the need for long computation times in domains with restrictive minimum pore throats. Training, Validation, and testing ratios are split in the 8:1:1 convention.

## 2.2 Flow Simulation by the Lattice Boltzmann Method

Flow within the pore space is calculated by the Lattice Boltzmann Method (LBM) using a Multi-Relaxation Time (MRT) scheme in D3Q19 quadrature space (Wang et al. 2020). LBM reformulates the Navier-Stokes Equations (NVE) by numerically estimating the resulting continuum mechanics from underlying kinetic theory. The kinetics of a bulk collection of particles within a control volume is estimated with a vector velocity space  $\xi_q$  and velocity distributions  $f_q$ . For each velocity space vector  $\xi_q$ , the velocity component in the specified direction is given by  $f_q$ . In 2D,  $q = 9$  and in 3D,  $q = 19$ . Using these concepts, an equation can be constructed that details the development of fluid transport. In particular, the momentum transport equation at location  $x_i$  over a timestep  $\delta t$  takes the form in Eq. 1 that relies on a collision operation  $J$  which recovers the Navier Stokes Equation, and outlined in detail in (McClure et al. 2014):

$$f_q(x_i + \xi_q \delta t, t + \delta t) = f_q(x_i, t) + J(x_i, t) \quad (1)$$

Single phase flow is simulated within the pore space of the segmented test samples until steady state conditions are reached. This is measured by tracking changes to the sample permeability  $K$  by Eq. 2:

$$K = \frac{\mu \bar{v} L}{\Delta P_x} \quad (2)$$

where  $\mu$  is the kinematic viscosity,  $\bar{v}$  is the mean velocity within the bulk domain,  $L$  is the length of the sample in the direction of flow, and  $\Delta P$  is the pressure difference between the inlet and outlet. The permeability of any given porous media is a constant value at steady state conditions, when the velocity fields and the pressure fields become time-static. In this case, simulations are run until the change in permeability over 1000 LBM timesteps  $|\frac{K_{t+\Delta t} - K_t}{K_t}|_{\Delta t=1000}$  is less than  $10^{-5}$ , which is consistent with the conditions used in Wang et al. (2021). All samples are simulated with a constant pressure drop between the inlet and

outlet in the  $z$  direction, set such that the mean pressure gradient is  $1e-5$  ( $P_{inlet} - P_{outlet} = 10^{-5} \times N_z$ ), and no-slip wall boundary conditions  $v_{wall} = 0$  are imposed along the other sides to avoid geometric inconsistencies associated with periodic boundary conditions. This no-slip condition is imposed using the LBM bounce-back rule. As LBM is used for generating ground truth flow fields, the units used throughout this study refer to lattice units. All LBM simulations were run using an MRT scheme (McClure et al. 2014) with a relaxation time of 0.7.

While this study is performed using lattice units, it is singularly important to ensure that the flow regime is representative of porous media. This ensures that the non-dimensionalisation of LBM results in solutions that are physically accurate for flow in porous media. Flow in porous media is characterised by the Reynolds Number  $Re = \frac{vL}{\nu}$  as  $Re \ll 1$ , and the kinematic viscosity  $\nu$  is related to the relaxation time  $\tau$  by  $\nu = \frac{\tau - 0.5}{3}$ . The characteristic length  $L$  is related to the correlation length (local/mean pore diameter), and the velocity  $v$  in these simulations ranges from  $1e-3$  to  $1e-5$ . As such, it is reasonable to assume that these LBM simulations are representative of Stokes-regime flow in porous media.

The velocity fields are used as ground truth output during training. In 2D, this takes the form of a tensor measuring  $(N_x, N_y, 2)$ , and in 3D, a tensor measuring  $(N_x, N_y, N_z, 3)$ . Since this is a study on porous media permeability and steady state velocity fields, the pressure gradient must be set low enough for the flow regime to be laminar, Stokes flow. This is the base assumption for permeability calculations in porous media. Using a lower pressure gradient would not cause any material changes in prediction, while using a higher pressure gradient can cause non-Darcy turbulent effects, which would alter the results (Henigh 2017).

## 2.3 Neural Network Architecture

Deep Learning is essentially an optimisation (minimisation) problem, in many ways analogous to the familiar and simple root finding problem solved with Newton–Raphson iteration. The major difference is the number of parameters, which typically range in the millions. Naturally, more complex techniques are required to “train” (read: optimise) a neural network, primarily back propagation (Goodfellow et al. 2016) to determine derivatives, and stochastic adaptive momentum gradient descent algorithms (Goodfellow et al. 2016; Kingma and Ba 2014) to identify local minima.

The network formulated for the task of predicting steady state velocity fields takes the form of a U-Net structure (Ronneberger et al. 2015) that contains gated convolutional layers and concatenated activations within residual blocks situated at each node along the U-Net structure. The design is adapted as a combination of features from PixelCNN++ (Salimans et al. 2017) and from the relatively simple implementation for the purpose of flow field prediction in simple geometries (Guo et al. 2016). Instead of the usual ResNet-style residual block in each section of the network (which consists of a pair of conv+batchNorm+ReLU modules), after the initial convolution at the start of each block, a gated convolution is performed, followed by a skip connection that adds the average-pool into itself. The gated convolution simply consists of (1) convolution with twice the filter number, (2) splitting the output into 2 parts along the filter dimension, (3) applying logistical activation to one of the parts, and (4) multiplying the parts together.

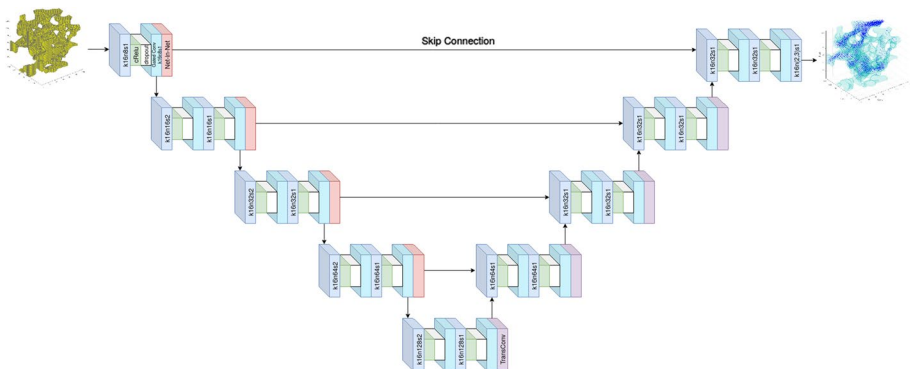
In each level of the descending portion of the U-Net structure, this set of operations is repeated twice, once with a stride of 1, and again with a stride of 2. The outputs from each level are skip connected with the ascending portions of the structure, which

consists of transposed convolutions with a stride of 2 followed by one instance of the residual block. The connection is preceded by a fully connected layer that operates only on the filter dimension (a network-in-network), preserving the fully convolutional structure of the network.

Notationally, this CNN ( $G_{\text{CNN}}$ ) takes in inputs  $\mathbf{X}$  which are either the binary grain-pore space  $\mathbf{X}_{\text{binary}}$  or the Euclidean Distance Transform of the binary space  $\mathbf{X}_{\text{EDT}}$ . The CNN performs a series of nonlinear operations on the data described by the layer-by-layer structure in the previous paragraphs, and in Fig. 2. The output  $\mathbf{Y}$  from this takes the form of velocity vector data. In 2D, this takes the form of a tensor measuring  $(N_x, N_y, 2)$ , and in 3D, a tensor measuring  $(N_x, N_y, N_z, 3)$ . Overall, the CNN thus operates under the relationship  $\mathbf{Y}_{\text{CNN}} = G_{\text{CNN}}(\mathbf{X}_{\text{binary,EDT}})$ . In order for the network to perform predictions with any degree of accuracy, the underlying nonlinear operations must be "trained" (read: optimised). This is done by an iterative process that minimises the difference between the CNN output  $\mathbf{Y}_{\text{CNN}}$  and the real LBM solution  $\mathbf{Y}_{\text{LBM}}$ . This difference between  $\mathbf{Y}_{\text{CNN}}$  and  $\mathbf{Y}_{\text{LBM}}$  is calculated by a loss function  $f_{\text{gloss}}$ . Thus, the CNN training process is notationally:  $G_{\text{CNN}} \rightarrow \min(f_{\text{gloss}}(\mathbf{Y}_{\text{LBM}}, G_{\text{CNN}}(\mathbf{X}_{\text{binary,EDT}})))$ . Once trained, any new, unseen input  $\mathbf{X}$  of the same type as the training input data (in this case, correlated porous media images) can be passed through the CNN to produce a prediction.

Due to the presence of significant regions of sparsity in solid voxels (up to 90%) where both inputs and outputs equal zero, network biases are disabled in the final configuration. This effect is tested in later sections, and shows marked improvement, as the mapping from the Euclidean Distance Transform (EDT) to the velocity vector space can be mapped multiplicatively.

In the configuration with a base kernel size of 4 and a base filter size of 32, the total number of trainable parameters is 75M. The learning rate decayed during training from  $1e-5$  to  $1e-8$ , and the Adam optimiser is used (Kingma and Ba 2014). The network architecture is tested under different configurations of kernel and filter sizes, outlined in subsequent sections. The original binary geometry, or the Euclidean Distance Map is also used as input, and custom physics-based loss functions are used. Other types of feature maps can be added as extra input channels, such as the tortuosity field,



**Fig. 2** Architecture of the Gated U-Net based on PixelCNN++. The U-Net structure is preserved, and gated convolutions are added to each block. A network-in-network layer is also applied before skip connections are applied



time-of-flight, Local Distance Maximum and though they are mentioned and discussed, are not directly explored in this study.

## 2.4 Loss Functions and Accuracy Measures

In terms of training the network, the primary loss function used is the Mean Squared Error, or the L2 loss between the real field  $F_r$  and the predicted field  $F_p$  over all pixels/voxels  $i$ , shown as Eq. 3. Since velocity fields follow a roughly log-normal distribution, sparsely occurring large magnitude velocity values that significantly contribute to mass flux are emphasised during training. Alternatively, one might choose to train such a problem with the Mean Absolute Error, or the L1 loss, as it better accounts for intermediate and small velocity vectors. In this study, the L2 error is chosen as it reduces the largest pixelwise errors faster.

$$L_{2\text{-Loss}} = \frac{\sum_i (F_{r_i} - F_{p_i})^2}{\sum_i i} \quad (3)$$

In the prediction of velocity fields during steady state flow, a pixelwise loss such as the L2 loss does not explicitly enforce any underlying physics. In order to improve the physical accuracy of the predicted fields, a mass conservation loss is applied. This mass conservation loss  $L_{\text{cons}}$  is calculated as the mean squared error of all flow rate profiles between the real and predicted field. The flow rate profile  $q_{n=x,y,z}$  is calculated for each direction  $x, y, z$  by the summation of the cross-sectional velocities multiplied by the voxel area (in this case  $1^2$  lattice unit). Thus, an expression for  $q$  in the directions  $x, y, z$  at the location  $x, y, z = i, j, k$  is:

$$q_x|_{x=i} = \sum_y \sum_z v_x|_{i,y,z} \quad (4)$$

$$q_y|_{y=j} = \sum_x \sum_z v_y|_{x,j,z} \quad (5)$$

$$q_z|_{z=k} = \sum_x \sum_y v_z|_{x,y,k} \quad (6)$$

For example, in 2D, the flow rate  $q$  along the  $X$  axis consists of the perpendicular component  $q_x$ , calculated by the sum of all velocity vectors in each  $X$  slice, summed along the  $Y$  axis, and similarly for  $q_y$ . Thus, for a tensorfield of  $n$  dimensions, the mass conservation loss is given by Eq. 7

$$L_{\text{cons}} = \sum_n \frac{\sum_i (q_{n_r} - q_{n_p})^2}{N} \quad (7)$$

where  $N$  is the length of the domain in the principal direction of flow expressed in voxels. It should be noted that a mass conservation loss function can alternatively be defined using a pixelwise method, which is similar to a gradient loss function. While the use of a cross-sectional flow-based conservation loss function is observed to be more stable (training can converge with only the  $L_{\text{cons}}$  active), this is not further explored in this study, and the incremental accuracy improvements are used as-is.

During testing, as a measurement of mass conservation, the summation L1 version of the mass conservation loss is used, and can be interpreted as the Scaled Total Absolute Flow Error (STAFE), given by Eq. 8

$$L_{\text{STAFE}} = \sum_n \frac{\sum_i |q_{n_r} - q_{n_p}|}{\sum_i |q_{N_r}|} \quad (8)$$

where  $N$  is the principal direction of flow. Traditionally, porous media flow characteristics at steady state is expressed by the permeability, given by Eq. 2, and the permeability error between the ground truth LBM permeability  $K_{\text{LBM}}$  and the CNN predicted permeability  $K_{\text{CNN}}$  can be simply calculated as Eq. 9

$$L_{\text{Perm}} = \left| 1 - \frac{K_{\text{CNN}}}{K_{\text{LBM}}} \right| \quad (9)$$

While not a measure of loss or accuracy, the tortuosity of a pore space can provide valuable quantitative information regarding the geometric complexity of the domain. The tortuosity can be obtained by solving the Poisson Equation within the pore space of the domain (Øren and Bakke 2002) with Dirichlet boundary conditions along the inlet and outlet of 1 and 0, respectively, and calculating the root-mean-square of the divergence field. This process is a simplification of a similar procedure to estimate fluid flow in porous media using the Laplace Equation (Chung et al. 2019), and incurs a similar computational cost when solved using Finite Methods with the Algebraic Multi-Grid approach.

The tortuosity of a domain is calculated as:

$$a = \bar{v} \quad (10)$$

where  $a$  is the tortuosity, and  $v$  is calculated as:

$$v = -\nabla P \quad (11)$$

and  $P$  is the solution to the elliptical partial differential equation:

$$-\nabla \cdot (\nabla P) = 0 \quad (12)$$

Dirichlet boundary conditions of  $P = 1$  at the inlet and  $P = 0$  at the outlet are imposed.

## 2.5 Accelerating Simulation to Steady State

The minimisation problem of Deep Learning is a natural restriction to the ultimate degree of accuracy that can be obtained from predictions using neural networks. As such, it is prudent to couple together such soft computing techniques with their rigid counterparts. As such, predicted velocity fields can be used as initial/restart conditions in LBM to accelerate the simulation to its steady state configuration. If the errors associated with pressure fields (which are also predicted during training or by training a separate pressure prediction network) and velocity fields are reasonably small and well distributed, then the propagation of error waves within the initialised simulation domain relaxes quickly to the steady state condition compared to an initialisation of a constant field.

### 3 Results

In the following sections, it is found that an optimal network configuration is one with as wide a kernel as possible, and that geometric complexity plays a significant role in estimation accuracy. Bulk properties as expected, were well predicted, but fine-scale velocity fields are inaccurate and unsuitable for use. By coupling these network based estimates to a direct simulator, the fully accurate steady state result is obtained in a fraction of the compute time.

Training and testing was performed on Nvidia Titan RTX GPUs, and training time for 2D and 3D networks ranged in the 2 day mark.

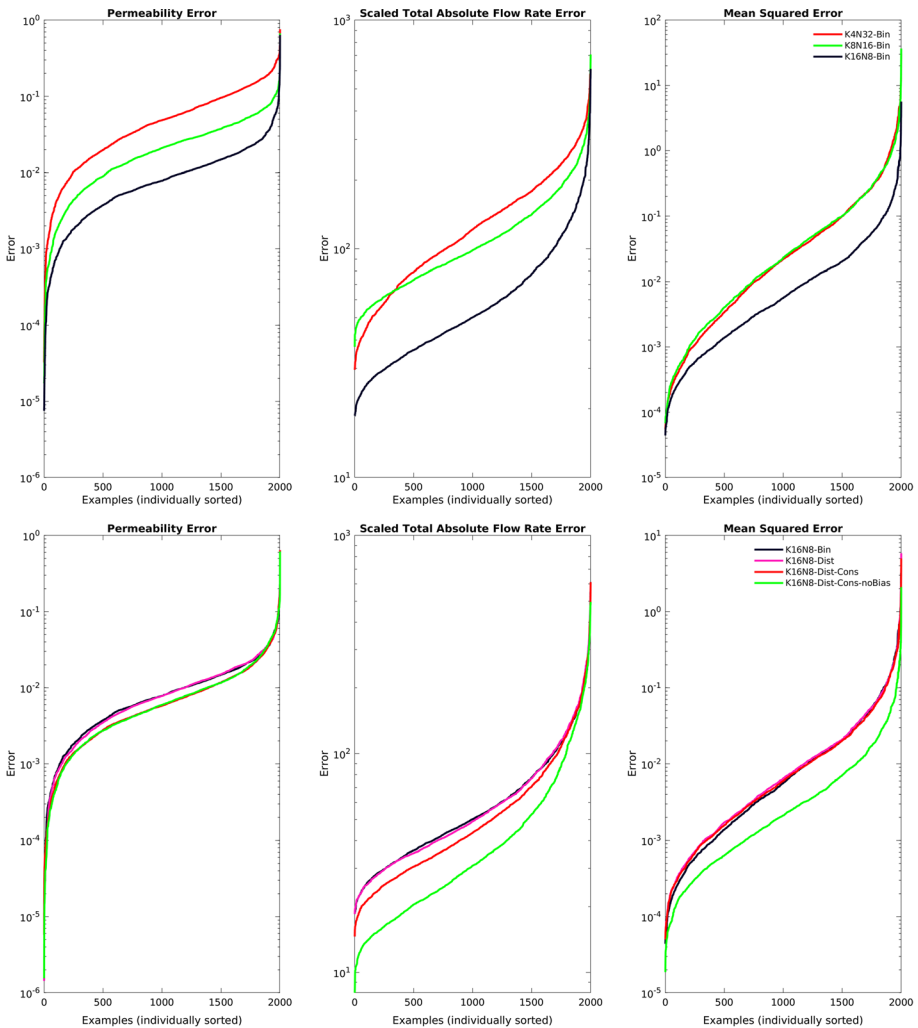
#### 3.1 Optimal Network Configuration

In 2D, by varying the kernel size and filter numbers inversely to each other, the network architecture retains the same number of trainable parameters, in this case 76M. Increasing the total number of trainable parameters would likely also result in increased performance by brute force, but this study aims to investigate the performance while keeping efficiency in mind. Configurations of K4N32 (Kernel Size = 4, Base Number of Filters = 32), K8N16, and K16N8 are tested on 2000 correlated fields with varying correlation lengths. It should be noted also, that a K32N4 configuration suffered from stability issues during training, likely due to poor numerical scaling from the larger set of kernel weights, setting K16N8 as the upper limit on kernel size. The resulting error is calculated using the pixel-wise MSE, the permeability error, and the scaled total absolute flow error (STAFE). The overall accuracies achieved shown in Fig. 3, presented as individually sorted graphs for the different configurations show that the K16N8 configuration performs best based on all accuracy metrics.

Next, using the K16N8 configuration, input domains are converted from their binary solid-pore representation to the EDT of the pore space. Similarly, the mass conservation loss function Eq. 7 is also applied to training, and finally, biases are removed from the network. It is important to explicitly mention at this point, that "Biases" refers to the offset term  $b$  within the CNN layers, which would normally perform the linear operation  $y = ax + b$ . By making these alterations, performance is further improved as shown in Fig. 3.

The kernel size plays a far more important role in velocity field prediction than the filter depth, which is a reflection of the manner in which velocity profiles develop in the pore space as a relation to the surrounding wall geometry. The wider the kernel, the more of this influence-at-a-distance is captured. The further use of EDT, mass conservation loss, and removal of biases all contribute towards improving the accuracy of the network when measured by metrics that are sensitive to the fine-scale accuracy such as the STAFE. These modifications to the network do not appear to significantly affect the CNN predicted permeability, as it tends to average out these errors.

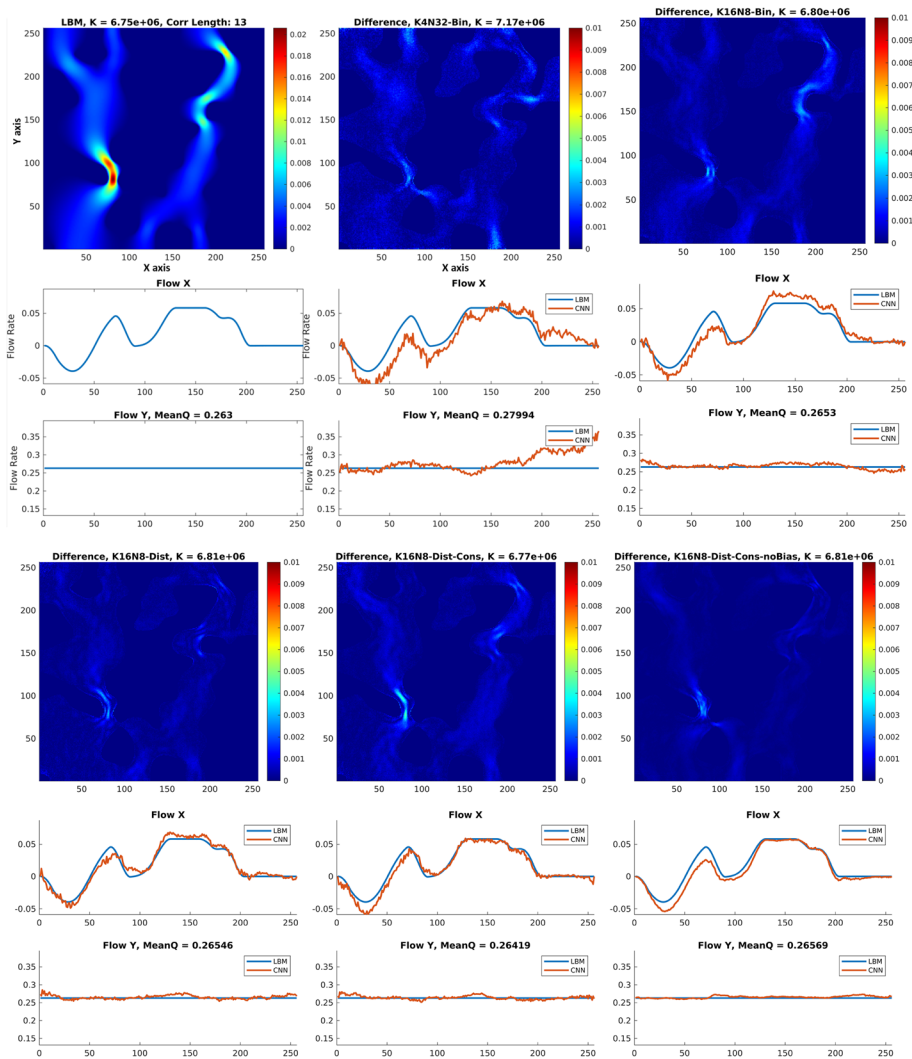
To illustrate the improvements achieved by these modifications to the network, visual comparisons of the difference map between the predicted velocity fields and the real LBM velocity fields are shown in Fig. 4. Plots of the perpendicular mass flux are also shown, and it can be seen that the improvement in accuracy is consistent with trends shown in previous Fig. 3. In these plots, shown in later Figs. 4 and 7, the velocity fields magnitudes are plotted as  $v_{\text{mag}} = \sqrt{v_x^2 + v_y^2 + v_z^2}$ , and flow rates are calculated in each direction as per Eqs. 4,



**Fig. 3** Top: Sorted plots of the accuracy achieved by Gated U-ResNet under different configurations of kernel and filter sizes. The best accuracies achieved over the testing dataset are generated by the K16N8 configuration, reflecting the relatively far influence that wall geometry has on velocity field prediction. Bottom: same, but further showing the improvement obtained from altering the input, loss function, and network weights

5, and 6.  $K$  refers to the permeability as calculated by Eq. 2, and MeanQ refers to the average value of Flow  $Y$ .

Visually, difference maps show that most errors occur in regions of high velocity, while plots of the cross-sectional mass flux show that, on a slice-by-slice basis, errors manifest as undifferentiable jumps and cusps in the flow profile. This difference between smooth ground truth flux profiles and the jagged CNN predictions in some cases (sub-optimal network/geometric complexity) could possibly be reduced using a total variational loss function (You et al. 2018). While the overall permeability error is low and the velocity fields



**Fig. 4** From top left to bottom right, the LBM ground truth velocity field, followed by difference maps and cross-sectional comparisons of the predicted velocity fields generated by the various configurations on the median testing image. Difference maps suggest good visual match, also supported by flow rate profiles in the  $X$  and  $Y$  axis. The small fluctuations in the flow profiles can be a source of error when using these predicted velocity fields in fine-scale analysis

difference maps suggest a low visual error, the flow rate errors are an indication that the individual voxelwise velocities are misaligned relative to the LBM simulation, which is analysed in further detail in a later section.

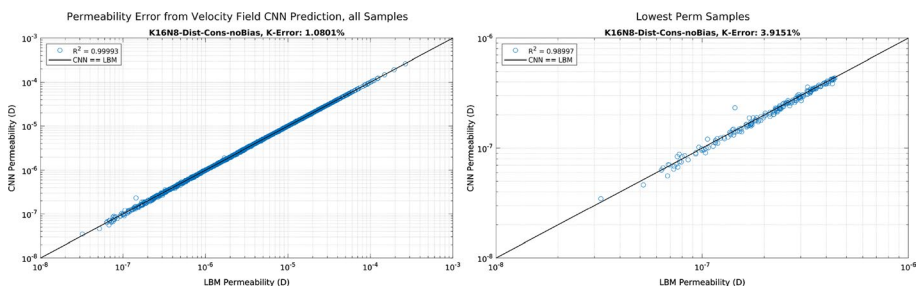
A permeability error of less than 10% is achieved by the K16N8 network over 99% of the 2000 correlated fields in the testing dataset. This degree of permeability error is reasonable, and consistent with other types of predictions using regression (Kamrava et al. 2019). This is depicted in a more traditional plot of LBM vs CNN predicted permeability, shown

in Fig. 5 of all 2000 unique 2D testing images spanning 4 orders of magnitude, and of the permeability error of the 100 samples with the lowest LBM permeability (and highest error). An important point to note is the relative insensitivity of CNN permeability prediction compared to the fine-scale flow and velocity predictions. In plots of the 100 lowest LBM permeability testing samples, the average permeability error for the best performing network configuration remains low, under 4% for the most geometrically complex testing samples. This does not translate well to the errors associated with velocity fields and flow profiles, as shown in Sect. 3.2. Both regression and velocity prediction based methods tend to give good estimates of the permeability of porous media. These 2D results are expected to be more accurate than those achievable in 3D, as the topology of the media is important for the prediction and this is more complex in 3D. This is explored in Sect. 3.5. Furthermore, the purpose of predicting velocity fields is to use these fields in analysis that depend on the finer voxel-by-voxel detail afforded by a direct translation of the domain geometry. This is investigated in detail in later sections.

Another important point to note is that, since the velocity sensitivity is much higher than permeability sensitivity, for the purposes of measuring the accuracy achieved by velocity field prediction, the permeability is a poor candidate. It is common for permeability errors in the order of 10% to be considered a good result using semi-analytical models, pore network models, or otherwise (Chung et al. 2019), but in this case, we see that even a 1% error (or significantly less) in the permeability as shown in Fig. 5 results in a significant deviation in the locally predicted velocity fields as shown in Fig. 7. To better understand the fine-scale errors, plots of the cross-sectional flow profiles, and the STAFE accuracy measure are better representations of velocity prediction.

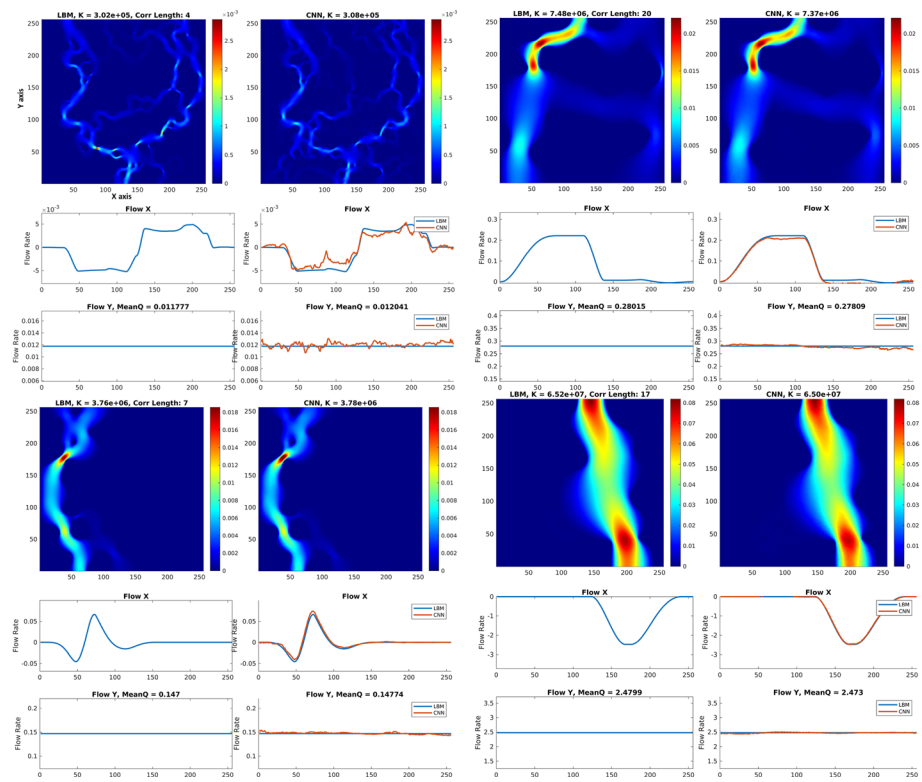
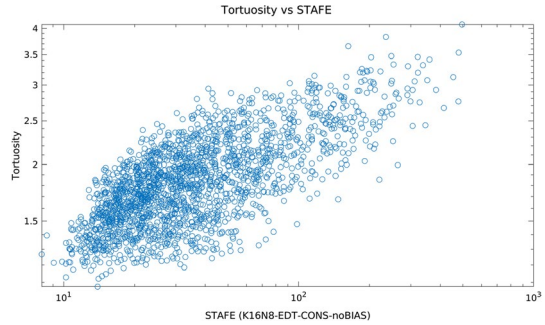
### 3.2 Effect of Geometric Complexity on Network Accuracy

From plots of the accuracy achieved by the trained networks in previous Fig. 3, a clear distribution of accuracy can be seen. It can be expected that this distribution is due to the variation in the porous structures trained and tested on, so this is quantitatively investigated. Plotting the tortuosity of each testing sample against the CNN accuracy achieved by the best performing network as shown in Fig. 6 shows a trend where accuracy falls with increasing geometric complexity. In more complex geometries, the prediction accuracy achieved is notably lower, and suggests the need for further refinement in accuracy in order to be quantitatively useful.



**Fig. 5** CNN permeability prediction accuracy of velocity field predictions. Overall errors are low, even though local velocity field error is high, seen in previous sections

**Fig. 6** Plot of tortuosity versus error, showing a trend where accuracy falls with increasing tortuosity, a measure of the geometric complexity of the porous media



**Fig. 7** Comparison of the predicted velocity fields for 4 domains of decreasing complexity. Geometric complexity is observed to result in higher errors. Lower error samples have larger, smoother, and wider flow channels. It can be expected that, CNN performance is acceptable up to a certain degree of complexity, and should be applied in a local-to-global manner

To now further visualise this effect, a finer analysis of the velocity fields is required. A plot of some geometrically complex samples as generated by the best performing K16N8 network (trained on EDT inputs, with conservation loss, and no biases) is shown in Fig. 7. These plots reveal that the overall errors drop dramatically once a certain threshold in the domain complexity is reached, which is seen on previous studies with flow on simpler geometries (Guo et al. 2016). The number of possible flow paths and the width of the pore

channels are clearly evident factors that influence the performance of the network. These plots show not only that the velocity field prediction is highly dependent on the geometry of the domain, they also show that mass conservation is not enforced adequately, with small to large fluctuations in the slice-by-slice mass flux in the  $X$  and  $X$  axes. While the overall error is visually and quantitatively small (see the  $X$  and  $X$  difference maps), mass flux profiles are as a whole reasonably predicted, and the velocity distributions are closely aligned, this fluctuation is likely to cause problems in directly using these velocity fields for other applications.

As can be seen, there is a clear relationship between the accuracy achieved in predicting the velocity field and the geometric complexity of the domain. The best matches occur in relatively simple tubular structures without any branching pathways, while the worst matches occurs in domains with multiple thin, low resolution pathways. When predicting the velocity field of a domain, these limitations should be considered. While a patch based approach may be applied whereby subsections of an overall domain are fed into the network, the local prediction of velocity fields is inconsistent with the highly non-local true solution. A wide flow path outside the bounds of a local patch will significantly affect local velocity fields. This issue can possibly be addressed by encoding some information regarding the magnitude of the local velocity field relative to the unseen global domain. In the interests of preserving accuracy however, a fully global reconstruction methodology is recommended. U-Net and its variants are fully convolutional (meaning that the trained network can be applied to domains of any size), so the variable domain size is useful if the problem is non-local such is the case with flow as it is dependent on boundary conditions and far-away geometries.

### 3.3 Fine Scale Sensitivity of Solute Transport to Predicted Fields

An application of flow simulation that can highlight the problematic errors that occur in velocity field prediction is the use of such velocity fields for analysis of solute transport in porous media. In this case, 2D convection-diffusion within the pore space is modelled using a Finite Volume solution of the convection diffusion equation. An inlet concentration of 1 is set, and an outlet concentration of 0 set (perfectly absorbing). To highlight the influence of velocity fields, the Peclet Number is set to a reasonably high value, near the limits of computational diffusion. An explicit time-stepping scheme is used, again to minimise numerical diffusion. In order to also ensure stability, upstreaming is applied instead of TVD methods.

For reference, the convection-diffusion equation in this case takes the form:

$$\frac{\delta c}{\delta t} = \nabla \cdot (D\nabla c) - \nabla \cdot (vc) \quad (13)$$

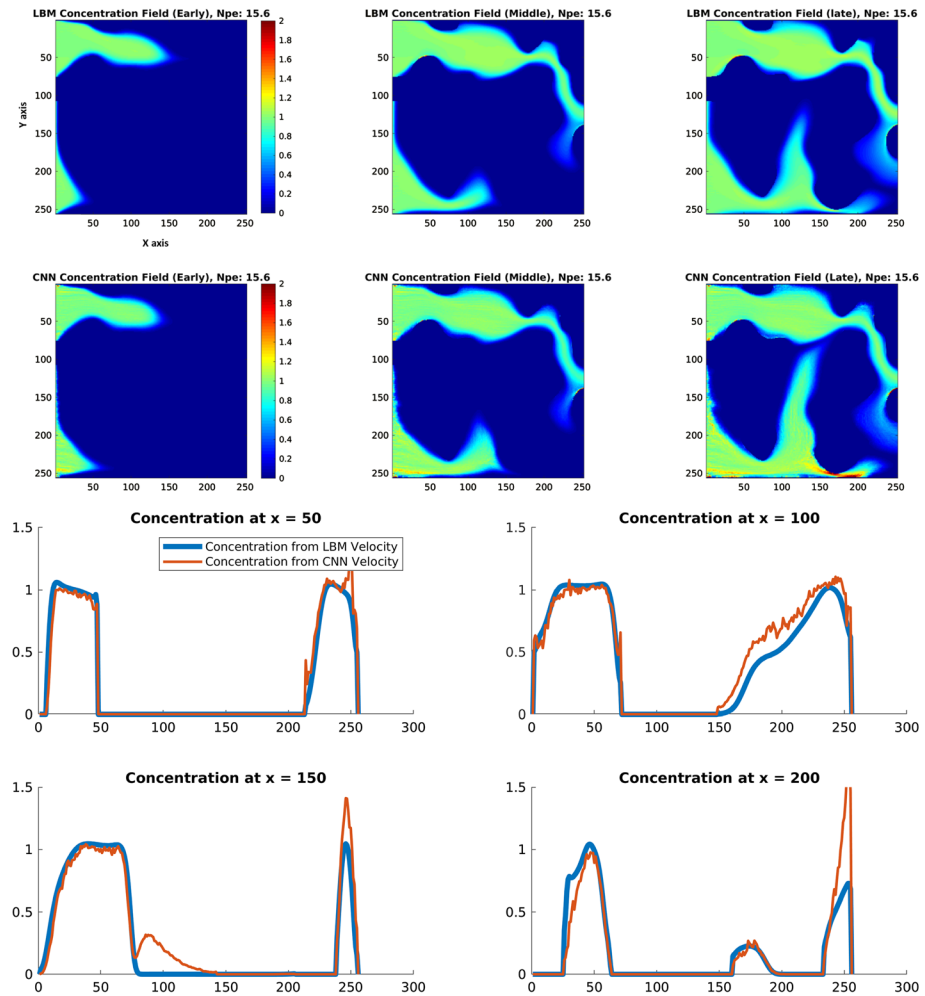
and solves the evolution of the concentration field  $c$ . The Peclet number in this study is defined as:

$$N_{pe} = \frac{L\bar{v}}{D} \quad (14)$$

and is set to 15.6, which represents a velocity-dominant system to gauge the accuracy of the predicted velocity fields. The value is chosen as it is close to the numerical accuracy limits of the flux upstreaming scheme (which tend to cause excessive numerical diffusion).



The issue with errors at the fine-scale is shown easily with a single, simple example, as shown in Fig. 8, of a relatively well predicted field (the testing sample with 33-percentile accuracy, shown in Fig. 7). While the real velocity fields result in expected transport of solute, without major zones of erroneous accumulation, the misaligned and off-magnitude velocity fields (relative to the LBM simulation) predicted near the wall, and in tight throats clearly shows a high error when applying these fields to sensitive tasks, such as solute transport. This error is shown in further detail in the cross-sectional concentration profiles at  $x = 50, 100, 150,$  and  $200$  for the late time concentration fields. The concentration profiles are reasonably well matching in their shape and profile, which is especially the case



**Fig. 8** Top: Comparison of concentration fields generated by convection diffusion with underlying velocity fields sourced from both LBM and CNN predictions. The fine-scale misalignment and errors in the CNN velocity fields at the walls and at throats can be seen to cause errors in the form of mass accumulation. Bottom: Cross-sectional comparison of concentration profiles at Late time. The CNN results show erroneous accumulation and irregular peaks due to misalignment of velocity field vectors from CNN prediction

when visually observing the colormap renderings. Much like the velocity field predictions in the previous section, there is a clear visual match that is useful for approximate evaluation of mass transport. However, the increased noise and the mismatching peaks are indicative of erroneous solute accumulation due to misaligned velocity vectors relative to the LBM simulation. This type of error can be problematic in cases where the accumulation of solute is a focal point in analysis, such as contaminant transport modelling and where reaction rates are highly sensitive to local concentration.

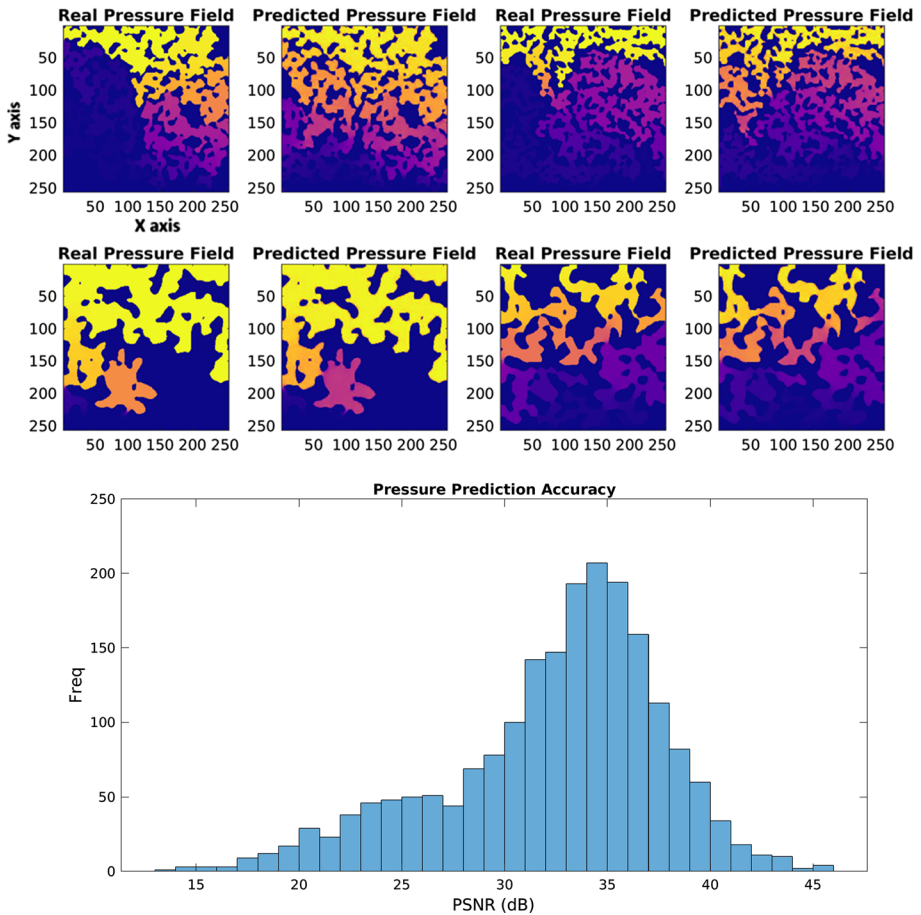
### 3.4 Accelerating Flow Simulation to Steady State

A potential solution to fix these prediction errors shown in previous sections is to use these velocity field predictions as initial estimates in direct flow simulation to realign the velocity fields and accelerate the computation to steady state. To accomplish this, both the velocity field and pressure fields are required for a given domain. Thus the network is trained to predict both the velocity field as well as the pressure field. Pressure fields obtained from LBM are scaled to  $[0, 1]$ , and samples are shown in Fig. 9 and the resulting accuracy is measured by the peak signal-to-noise ratio (PSNR) in decibels obtained on predicting the pressure fields of the 2000 testing images.

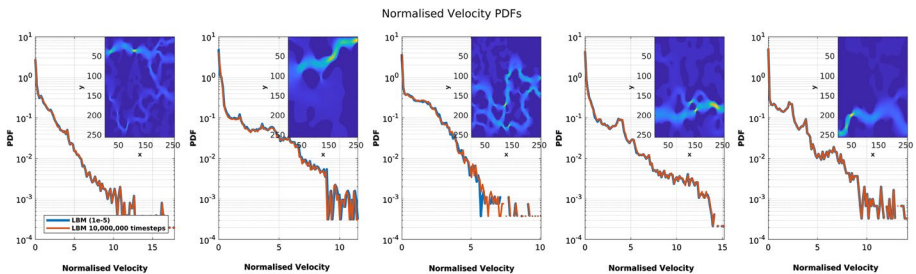
The visualisation of the predicted pressure fields shows that pressure prediction is reasonably accurate, though much like the case in velocity field prediction, chambers that are not prone to flow are highlighted by the network, as identification of the principle flow channels remains difficult with this implementation. PSNR plots confirm that the accuracy is relatively high. For ease of reference, a PSNR of 10 corresponds to a MSE of 10%, a PSNR of 20 is 1% and so on.

Using both the velocity predictions and the pressure field predictions as input into LBM (or any direct Navier-Stokes solver), the speed up in convergence to steady state conditions is tracked and compared to the case with simulation initialised with zero velocity and uniform pressure. The convergence criteria is defined in this case by the relative change in calculated system permeability every 1000 LBM timesteps. In these simulations, this criteria is set to  $1e-5$ . This criterion was also cross-referenced to the mean local relative velocity magnitude difference as calculated by  $|\frac{v_{i,t+\Delta t} - v_{i,t}}{v_{i,t}}|_{\Delta t=1}$ , and over the 2000 testing samples, resulted in a convergence factor of  $1e-6$  to  $1e-5$ , which is a commonly used metric and result for velocity field convergence. To validate this, 5 example samples are simulated to 10,000,000 LBM timesteps, and their velocity fields are compared to the velocity fields obtained from the criteria of  $1e-5$ , which took less than 50,000 LBM timesteps to reach for all samples. The velocity PDF ( $\frac{|v|}{\int_v \bar{v} dv}$ ) for the 5 samples is calculated and compared between the 10,000,000 timestep case, and the convergence criteria of  $1e-5$ , and shown in Fig. 10. The results show a very close match in the PDFs between simulation cases, over a range of porous domains of varying complexity. This supports that the choice of  $1e-5$  as convergence criteria for Eq. 9 is adequate in these cases to represent steady state flow fields.

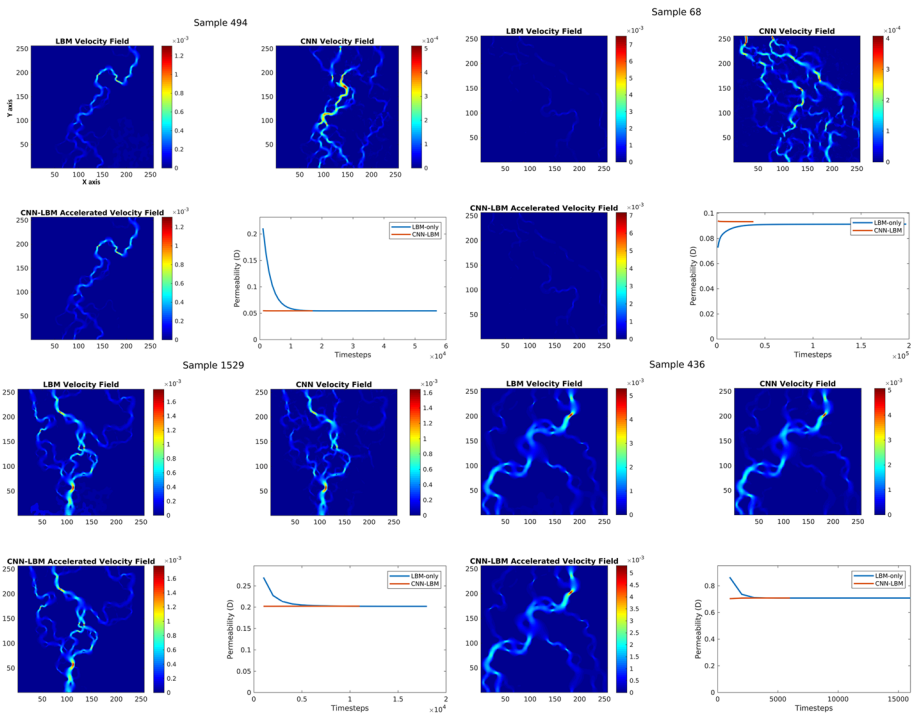
Plots of the worst, 0.5%, 5% and 10% ranked samples are shown in Fig. 11, and indicate that the initialisation with even poorly predicted velocity fields results in rapid convergence to steady state conditions. A speed up of 2–5 is observed when terminating simulations at the aforementioned  $1e-5$  criteria, and resulting velocity fields are identical to those obtained from LBM-only simulations that take longer to stabilise. It is observed that there is a long plateau-like tail to the permeability calculated by LBM-only compared to CNN



**Fig. 9** Top: Examples comparing the pressure field obtained from LBM, and the predicted pressure field generated by the network. The major pressure gradients are captured accurately, though there is pressure mismatch in some porous chambers. Bottom: Histogram of the accuracy achieved by pressure field prediction over the testing dataset. With the exception of the most complex and tortuous geometries, accuracy is high, with the majority of cases achieving a PSNR of over 20 dB, corresponding to an L2 error of 1%



**Fig. 10** The velocity PDF ( $\frac{|\vec{v}|}{\frac{1}{V} \int_V \vec{v} dV}$ ) for the 5 samples shown in the figure (for visual aid), for simulation cases of 10,000,000 timesteps, and 1e−5 convergence criterion (< 50,000 timesteps for these 5 cases)



**Fig. 11** From top left to bottom right, comparison of the LBM-CNN Accelerated velocity fields generated for the worst, 0.5%, 5% and 10% ranked samples within the validation dataset. Even for the worst case, where predicted velocity fields are poorly predicted, the LBM simulation converges rapidly to the true steady state solution. In all cases shown here, representing some of the most error-prone geometries, the benefits of acceleration to steady state is apparent

accelerated LBM in this region, significant change to the convergence do not occur, it is that the criteria of  $1e-5$  is relatively slower to reach for LBM-only, with much of the plateau spent around the  $1e-4$  mark in terms of relative permeability error. It is important to note that this speedup is dependent on the heterogeneity and geometric complexity of the porous media, and the effect is most apparent in the 4 example shown in Fig. 11, with some complex geometries. In the case of a mostly cylindrical, pipe-like domain, the speedup is measured in this study to be negligible, but at the same time, the CNN predictions are also more accurate. Thus accelerating flow simulation with such predictions works best in complex geometries.

It is important to consider the resources and time required to train this network as well, as these CNN networks require many hours of GPU-time to train up. In this case, the roughly 2 day training time required on a single RTX Titan GPU for these networks represents a significant initial resource requirement. The benefit is that on unseen porous media of similar morphology (i.e. correlated fields or otherwise), this network does not require any further training to produce an initial guess as accurate as what was achieved on the testing data in previous sections. As such, it is common to simply report the testing CPU time (Santos et al. 2020), which is typically less than 1 s-true in this study as well. Despite this, it is important to be mindful of the initial training requirements. On a porous media sample that is outside the scope of the training dataset that the CNN is trained on, such as heterogeneous carbonate,

accuracy is much lower and velocity fields tend to be unreliable (Santos et al. 2020), and thus retraining the network becomes a non-trivial and highly intensive task that involves sourcing a new dataset, training, testing, and implementing on the sample of interest. In such cases, the usefulness of an initial guess via CNN predictions, and the speedup during subsequent simulation is less clear.

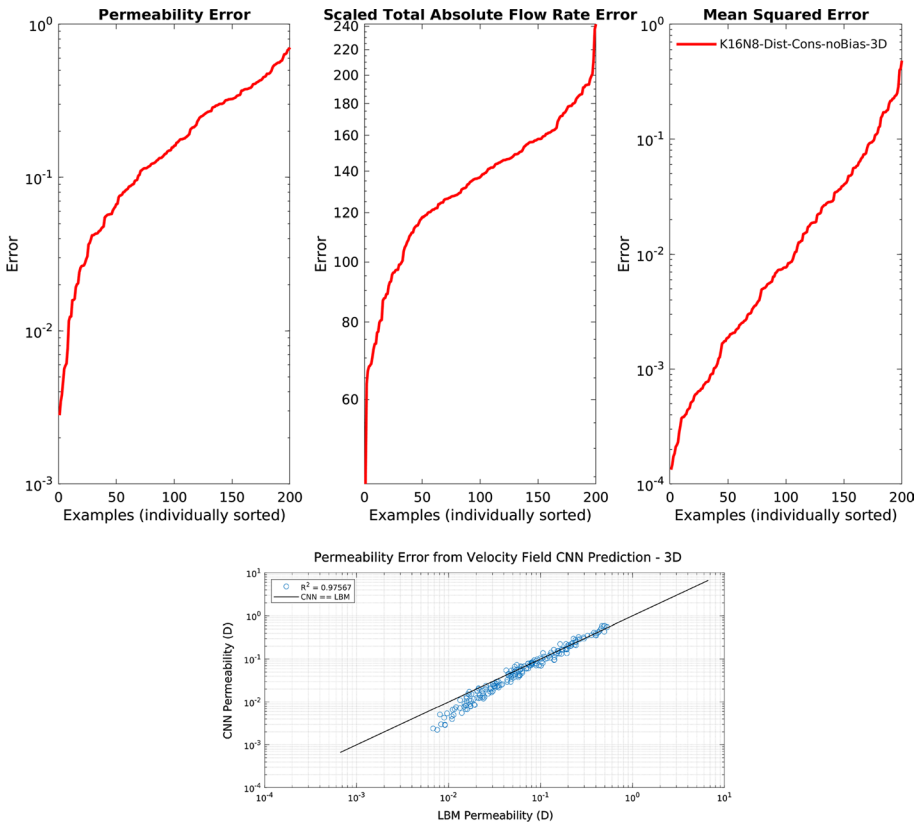
### 3.5 3D Network Performance

After having established the best configurations, the limitations in terms of accuracy, and a method of eliminating said limitations, the network is now trained and tested in 3D. Here, 1000 correlated fields of  $128^3$  voxels are used, 800 for training and 200 for testing. In 3D, the generated correlated fields range from simple to highly heterogeneous, and as such, much like the case in 2D where geometric complexity plays a large role in determining the accuracy of the prediction, this is more-so the case in 3D with a wider solution space. Plots shown in Fig. 12 of the permeability error, STAFE, and L2 Error show a marked decrease in achieved accuracy compared to 2D results. This is likely due to the added complexity due to the dimensionality increase, the reduction in training data, and a dataset comprised of more complex examples than in 2D. In terms of the permeability error, a majority portion of the testing samples result in predictions with an error higher than 10%.

A plot of the CNN permeability predicted vs the real LBM permeability in Fig. 12 shows that accuracy is lower for samples with lower permeabilities. This is likely due to the lower overall velocity values within the domains contributing less to the training of the network. While the mass conservation loss function reduces this effect, the local minima problem inevitably creates upper limits to the accuracy achieved. These samples are  $128^3$  correlated fields with a correlation length of around 10% the domain length, and are thus highly tortuous domains, representing domains with a mean pore body resolved to only several voxels. This can physically occur for example in sandstone  $\mu$ CT images with a resolution that does not adequately resolve the pore bodies between grains. When predicting the velocity field within 3D porous media, much like the case in 2D, accuracy is limited by the geometric complexity of the domain.

While accuracy limitations of the network limit the use of these velocity fields as-is, accelerating direct simulation to steady state conditions using these predicted fields remains beneficial. Visualisation of the porous media with embedded velocity fields in Fig. 13 shows that errors manifest similarly to the 2D cases in previous sections. The principal flow path remains elusive when the geometry is too tortuous, and velocity values near the walls and in tight throats are inaccurate. These errors can be overcome by accelerated direct simulation, and show that the convergence rate to steady state conditions is an order of magnitude faster.

This technique of preconditioning a direct simulation with a steady state estimate to accelerate the simulation to its true steady state conditions is a simple and generic technique. Aside from using it for correcting CNN predictions, it can potentially also be used with the outputs of Pore Network Models, Laplace Solvers, and other Navier–Stokes Estimators.

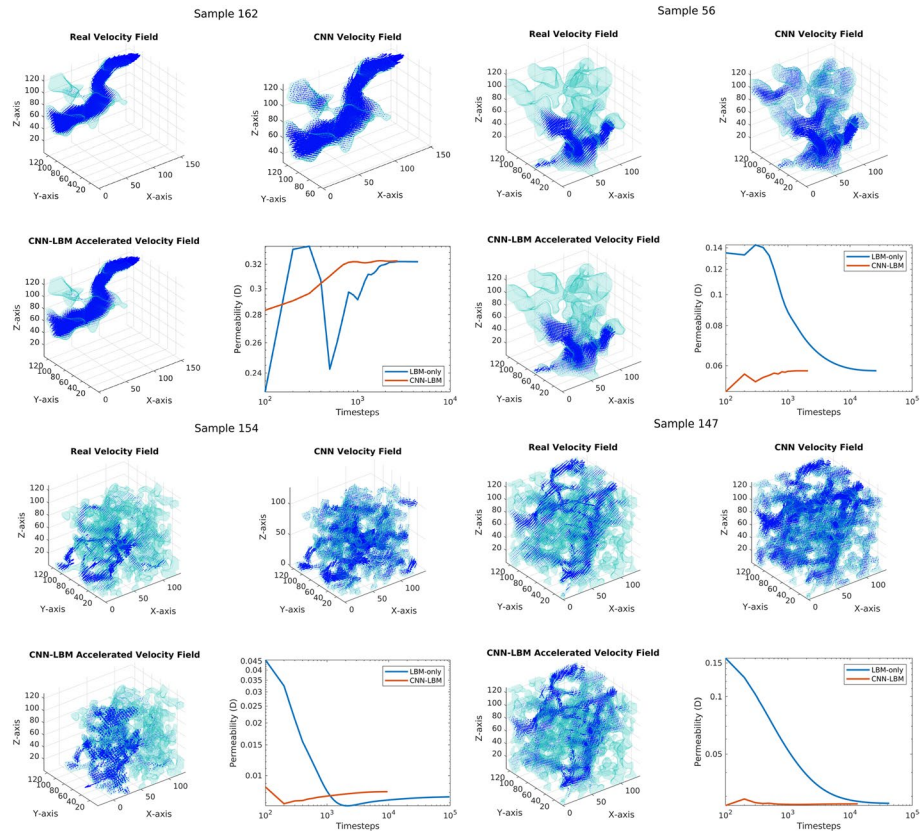


**Fig. 12** Top: Plots showing the accuracy achieved by the network when predicting velocity fields, over 200 examples of  $128^3$  correlated fields. Due to the increase in network complexity and geometric complexity, a larger proportion of testing samples result in a permeability error higher than 10%. Bottom: Plot of 3D testing sample permeability accuracy. Accuracy remains high for intermediate to high permeability samples, while the network underestimates values for lower permeability samples, due to the smaller values of velocity present within them

## 4 Conclusions and Recommendations

This study shows that, in the tortuous flow paths of porous media, an accurate prediction of the steady state velocity field and permeability can be obtained, with accuracy dependent on the geometric complexity. Accuracy achieved in 2D and 3D testing is consistent with previous studies in simple and porous media (Guo et al. 2016; Santos et al. 2020).

It is observed that the network architecture achieves a good result in permeability estimation (down to less than 1% error in 2D and less than 10% in 3D) by the prediction of velocity fields, which is comparable to results achieved by other velocity prediction networks and regression networks. However, the underlying velocity field is not guaranteed to possess the necessary voxel-by-voxel accuracy required for actually using these velocity fields for further analysis. Advection dominant solute transport in relatively high Peclet Numbers (near the numerical diffusion limit of explicit finite volume methods) using even the better performing predicted velocity fields in relatively simple 2D geometries shows that fine-scale errors are too large near walls and tight throats to be directly useful.



**Fig. 13** Visualisation of some selected velocity fields in 3D porous media. Consistent with the results obtained in 2D testing, simple geometries result in a better match in velocity fields, while complex porous media results in predictions that struggle to ascertain the principal flow paths. Also shown are loglog plots of the acceleration achieved by preconditioning LBM with the velocity field predictions. A speedup factor of an order of magnitude is observed

In simpler geometries, this estimate can be used as-is for permeability estimation and garnering an approximate understanding of the flow paths. In complex geometries that the tested network design in this study struggles with, the predictions can be used as initial conditions (alongside a pressure field prediction) in direct simulation to reach a fully accurate result in a fraction of the compute time. This concept is supported by tests using LBM as the direct simulation approach, and the idea of using a "good initial guess" is also present in solving the NVE in the form of the SIMPLE (semi-implicit method for pressure-linked equations) algorithm for steady state flows.

Limitations resulting from the architecture of the predictor in this study are expected to be surmountable by improvements in soft computing methods of approximating velocity fields. Self correcting approximation methods are evidently the superior choice in these cases (Wang et al. 2020; Hennigh 2017). Regardless, the efficacy of using these predictions (even the worst performing predictions), to accelerate direct simulation to steady state conditions is shown to be effective, particularly in more geometrically complex porous media.

Furthermore, the acceleration technique used in this study is general, and can be applied to any output from a flow approximator, such as pore network models, Laplace Solvers.

The use of correlated fields for the analysis of porous media is commonly used as a representation of generic geometries (Liu and Mostaghimi 2017), and acts as a proxy for the types of irregular porous media that benefit from direct simulation, such as rocks, and natural filters and membranes. The dataset is designed to be a stress test of the capabilities of CNN based predictors, and the capacity for tolerance that the acceleration method can handle while still posting reductions in compute time to steady state.

While CNNs are focused on end-to-end mappings (raw input to final output), the use of only the distance map as an encoded input can be improved to also include other metrics as a form of feature engineering, such as the maximally inscribed radius (Santos et al. 2020), the local distance maximum chamber (Wang et al. 2020), the local diffusivity [which is an output of the tortuosity calculation (Øren and Bakke 2002)], the time-of-flight (Hassouna and Farag 2007) (which is similar to the tortuosity). These encoded inputs should be approached with caution, as some of these are surprisingly intensive to solve and scale poorly in 3D. For example, the tortuosity solved by the Algebraic Multigrid (AMG) method scales by  $O[N]$  (Chung et al. 2019) and the time-of-flight solved by efficient fast marching scales by  $O[N \log N]$  (Hassouna and Farag 2007), which reduces the effectiveness of the method if a large 3D porous domain requires significant preprocessing. A middle ground between speed and input encoding should be met, and in this study, the  $O[N]$  scaling euclidean distance transform is used. It should be mentioned also that the Laplace method of approximating flow in porous media is also  $O[N]$  scaling, again emphasising that the cost of computing input parameters should be considered.

The choice of loss functions and the use of custom loss functions can be further improved, much like the encoding of the input data. Scaling the local L2 loss or using the L1 is an option that is under-explored in this study, but has shown promise in other similar work (Santos et al. 2020). The loss in achievable accuracy when transitioning from 2D to 3D networks is further indication that the prediction of velocity fields can be improved further. While this loss in accuracy is patched over with an acceleration routine, it would be beneficial to see further improvements in 3D results. Achieving a less than 10% error in permeability estimation in 3D velocity field prediction provides no guarantees whatsoever that the predicted velocity fields are physically accurate or useful in further analysis that required such fine-scale accuracy. Despite these limitations, the accuracy is sufficient to be an excellent preconditioner for accelerating simulations to steady state conditions in a fraction of the otherwise compute time.

**Data Availability** The source code used in this study is available at <https://github.com/yingDaWang-UNSW/VelCNNs>.

## References

- Ahmadi, M.A., Ebadi, M., Shokrollahi, A., Majidi, S.M.J.: Evolving artificial neural network and imperialist competitive algorithm for prediction oil flow rate of the reservoir. *Appl. Soft Comput.* **13**(2), 1085–1098 (2013). <https://doi.org/10.1016/j.asoc.2012.10.009>
- Alqahtani, N., Alzubaidi, F., Armstrong, R.T., Swietojanski, P., Mostaghimi, P.: Machine learning for predicting properties of porous media from 2d x-ray images. *J. Pet. Sci. Eng.* **184**, 106514 (2020). <https://doi.org/10.1016/j.petrol.2019.106514>



- Alqahtani, N., Chung, T., Wang, Y.D., Armstrong, R.T., Swietojanski, P., Mostaghimi, P.: Flow-based characterization of digital rock images using deep learning. *Adv. Water Resour.* (2021). <https://doi.org/10.2118/205376-PA>
- Blunt, M.J.: *Multiphase Flow in Permeable Media: A Pore-Scale Perspective*. Cambridge University Press, Cambridge (2017)
- Blunt, M.J., Jackson, M.D., Piri, M., Valvatne, P.H.: Detailed physics, predictive capabilities and macroscopic consequences for pore-network models of multiphase flow. *Adv. Water Resour.* **25**(8), 1069–1089 (2002). [https://doi.org/10.1016/S0309-1708\(02\)00049-0](https://doi.org/10.1016/S0309-1708(02)00049-0)
- Blunt, M.J., Bijeljic, B., Dong, H., Gharbi, O., Iglauer, S., Mostaghimi, P., Paluszny, A., Pentland, C.: Pore-scale imaging and modelling. *Adv. Water Resour.* **51**, 197–216 (2013). <https://doi.org/10.1016/j.advwatres.2012.03.003>
- Chappell, N.A., Lancaster, J.W.: Comparison of methodological uncertainties within permeability measurements. *Hydrol. Process. Int. J.* **21**(18), 2504–2514 (2007)
- Chung, T., Wang, Y.D., Armstrong, R.T., Mostaghimi, P.: Approximating permeability of micro-ct images using elliptic flow equations. *SPE J.* **24**, 1–154 (2019)
- Culligan, K., Wildenschild, D., Christensen, B., Gray, W., Rivers, M.: Pore-scale characteristics of multiphase flow in porous media: a comparison of air-water and oil-water experiments. *Adv. Water Resour.* **29**(2), 227–238 (2006). <https://doi.org/10.1016/j.advwatres.2005.03.021>
- Dong, H., Fjeldstad, S., Alberts, L., Roth, S., Bakke, S., Øren, P.-E., et al.: Pore network modelling on carbonate: a comparative study of different micro-ct network extraction methods. In: *International Symposium of the Society of Core Analysts, Society of Core Analysts* (2008)
- Erofeev, A., Orlov, D., Ryzhov, A., Koroteev, D.: Prediction of porosity and permeability alteration based on machine learning algorithms. *Transp. Porous Media* **128**(2), 677–700 (2019)
- Fenwick, D.H., Blunt, M.J.: Three-dimensional modeling of three phase imbibition and drainage. *Adv. Water Resour.* **21**(2), 121–143 (1998). [https://doi.org/10.1016/S0309-1708\(96\)00037-1](https://doi.org/10.1016/S0309-1708(96)00037-1)
- Goodfellow, I., Bengio, Y., Courville, A.: *Deep Learning*. MIT Press, Cambridge (2016)
- Gruber, M., Johnson, C., Tang, C., Jensen, M., Yde, L., Hélix-Nielsen, C.: Computational fluid dynamics simulations of flow and concentration polarization in forward osmosis membrane systems. *J. Membr. Sci.* **379**(1), 488–495 (2011). <https://doi.org/10.1016/j.memsci.2011.06.022>
- Guo, X., Li, W., Iorio, F.: Convolutional neural networks for steady flow approximation. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16*. Association for Computing Machinery, New York, NY, USA, 2016, pp. 481–490. <https://doi.org/10.1145/2939672.2939738>
- Hassouna, M.S., Farag, A.A.: Multistencils fast marching methods: a highly accurate solution to the Eikonal equation on Cartesian domains. *IEEE Trans. Pattern Anal. Mach. Intell.* **29**(9), 1563–1574 (2007)
- Hennigh, O.: Lat-net: compressing lattice Boltzmann flow simulations using deep neural networks. arXiv preprint [arXiv:1705.09036](https://arxiv.org/abs/1705.09036) (2017)
- Hilpert, M., Miller, C.T.: Pore-morphology-based simulation of drainage in totally wetting porous media. *Adv. Water Resour.* **24**(3), 243–255 (2001). [https://doi.org/10.1016/S0309-1708\(00\)00056-7](https://doi.org/10.1016/S0309-1708(00)00056-7)
- Jin, X., Cheng, P., Chen, W.-L., Li, H.: Prediction model of velocity field around circular cylinder over various Reynolds numbers by fusion convolutional neural networks based on pressure on the cylinder. *Phys. Fluids* **30**(4), 047105 (2018)
- Kamrava, S., Tahmasebi, P., Sahimi, M.: Linking morphology of porous media to their macroscopic permeability by deep learning. *Transp. Porous Media* **131**, 427–448 (2019)
- Keil, F.J., Rieckmann, C.: Optimization of three-dimensional catalyst pore structures. *Chem. Eng. Sci.* **49**(24, Part A), 4811–4822 (1994). [https://doi.org/10.1016/S0009-2509\(05\)80061-2](https://doi.org/10.1016/S0009-2509(05)80061-2)
- Khanafar, K., Cook, K., Marafie, A.: The role of porous media in modeling fluid flow within hollow fiber membranes of the total artificial lung. *J. Porous Media* **15**(2) (2012)
- Kingma, D., Ba, J.: Adam: a method for stochastic optimization (2014)
- Liu, M., Mostaghimi, P.: Characterisation of reactive transport in pore-scale correlated porous media. *Chem. Eng. Sci.* **173**, 121–130 (2017). <https://doi.org/10.1016/j.ces.2017.06.044>
- Liu, M., Shabaninejad, M., Mostaghimi, P.: Impact of mineralogical heterogeneity on reactive transport modelling. *Comput. Geosci.* **104**, 12–19 (2017). <https://doi.org/10.1016/j.cageo.2017.03.020>
- Liu, M., Shabaninejad, M., Mostaghimi, P.: Predictions of permeability, surface area and average dissolution rate during reactive transport in multi-mineral rocks. *J. Pet. Sci. Eng.* **170**, 130–138 (2018). <https://doi.org/10.1016/j.petrol.2018.06.010>
- Mahbub, F., Shi, M.A.A., Nasu, N.J., Wang, Y., Zheng, H.: Mixed stabilized finite element method for the stationary stokes-dual-permeability fluid flow model. *Comput. Methods Appl. Mech. Eng.* **358**, 112616 (2020). <https://doi.org/10.1016/j.cma.2019.112616>

- McClure, J., Prins, J., Miller, C.: A novel heterogeneous algorithm to simulate multiphase flow in porous media on multicore CPU–GPU systems. *Comput. Phys. Commun.* **185**(7), 1865–1874 (2014). <https://doi.org/10.1016/j.cpc.2014.03.012>
- Mostaghimi, P., Mahani, H., et al.: A quantitative and qualitative comparison of coarse-grid-generation techniques for modeling fluid displacement in heterogeneous porous media. *SPE Reservoir Eval. Eng.* **13**(01), 24–36 (2010)
- Mostaghimi, P., Blunt, M.J., Bijeljic, B.: Computations of absolute permeability on micro-ct images. *Math. Geosci.* **45**(1), 103–125 (2013)
- Mostaghimi, P., Liu, M., Arns, C.H.: Numerical simulation of reactive transport on micro-ct images. *Math. Geosci.* **48**(8), 963–983 (2016). <https://doi.org/10.1007/s11004-016-9640-3>
- Øren, P.-E., Bakke, S.: Process based reconstruction of sandstones and prediction of transport properties. *Transp. Porous Media* **46**(2–3), 311–343 (2002)
- Rabbani, A., Babaei, M.: Hybrid pore-network and lattice-Boltzmann permeability modelling accelerated by machine learning. *Adv. Water Resour.* **126**, 116–128 (2019)
- Ronneberger, O., Fischer, P., Brox, T.: U-net: convolutional networks for biomedical image segmentation (2015). [arXiv:1505.04597](https://arxiv.org/abs/1505.04597)
- Salimans, T., Karpathy, A., Chen, X., Kingma, D.P.: Pixelcnn++: improving the pixelcnn with discretized logistic mixture likelihood and other modifications (2017). [arXiv:1701.05517](https://arxiv.org/abs/1701.05517)
- Santos, J.E., Xu, D., Jo, H., Landry, C.J., Prodanović, M., Pyrcz, M.J.: Poreflow-net: a 3d convolutional neural network to predict fluid flow through porous media. *Adv. Water Resour.* **138**, 103539 (2020)
- Shabro, V., Torres-Verdín, C., Javadpour, F., Sepehrnoori, K.: Finite-difference approximation for fluid-flow simulation and calculation of permeability in porous media. *Transp. Porous Media* **94**(3), 775–793 (2012)
- Sudakov, O., Burnaev, E., Koroteev, D.: Driving digital rock towards machine learning: predicting permeability with gradient boosting and deep neural networks. *Comput. Geosci.* **127**, 91–98 (2019)
- Sun, Z., Logé, R.E., Bernacki, M.: 3d finite element model of semi-solid permeability in an equiaxed granular structure. *Comput. Mater. Sci.* **49**(1), 158–170 (2010)
- Tian, J., Qi, C., Sun, Y., Yaseen, Z.M.: Surrogate permeability modelling of low-permeable rocks using convolutional neural networks. *Comput. Methods Appl. Mech. Eng.* **366**, 113103 (2020). <https://doi.org/10.1016/j.cma.2020.113103>
- Torskaya, T., Shabro, V., Torres-Verdín, C., Salazar-Tio, R., Revil, A.: Grain shape effects on permeability, formation factor, and capillary pressure from pore-scale modeling. *Transp. Porous Media* **102**(1), 71–90 (2014)
- Van Oort, C.M., Xu, D., Offner, S.S.R., Gutermuth, R.A.: A convolutional neural network approach for shell identification. *Astrophys. J.* **880**(2), 83 (2019). <https://doi.org/10.3847/1538-4357/ab275e>
- Wang, Y.D., Shabaninejad, M., Armstrong, R.T., Mostaghimi, P.: Physical accuracy of deep neural networks for 2d and 3d multi-mineral segmentation of rock micro-ct images (2020). [arXiv:2002.05322](https://arxiv.org/abs/2002.05322)
- Wang, Y., Lin, G.: Efficient deep learning techniques for multiphase flow simulation in heterogeneous porous media. *J. Comput. Phys.* **401**, 108968 (2020)
- Wang, Y.D., Chung, T., Armstrong, R.T., McClure, J.E., Mostaghimi, P.: Computations of permeability of large rock images by dual grid domain decomposition. *Adv. Water Resour.* **126**, 1–14 (2019). <https://doi.org/10.1016/j.advwatres.2019.02.002>
- Wang, C.-S., Shen, P.-Y., Liou, T.-M.: A consistent thermal lattice Boltzmann method for heat transfer in arbitrary combinations of solid, fluid, and porous media. *Comput. Methods Appl. Mech. Eng.* **368**, 113200 (2020). <https://doi.org/10.1016/j.cma.2020.113200>
- Wang, Y., Chung, T., Armstrong, R., McClure, J., Ramstad, T., Mostaghimi, P.: Accelerated computation of relative permeability by coupled morphological and direct multiphase flow simulation. *J. Comput. Phys.* **401**, 108966 (2020). <https://doi.org/10.1016/j.jcp.2019.108966>
- Wang, Y., Armstrong, R.T., Mostaghimi, P.: Boosting resolution and recovering texture of 2d and 3d micro-ct images with deep learning. *Water Resour. Res.* (2020). <https://doi.org/10.1029/2019WR026052>
- Wang, Y.D., Blunt, M.J., Armstrong, R.T., Mostaghimi, P.: Deep learning in pore scale imaging and modeling. *Earth Sci. Rev.* (2021). <https://doi.org/10.1016/j.earscirev.2021.103555>
- Wang, Y.D., Chung, T., Rabbani, A., Armstrong, R.T., Mostaghimi, P.: Fast direct flow simulation in porous media by coupling with pore network and Laplace models. *Adv. Water Resour.* **150**, 103883 (2021)
- White, J.A., Borja, R.I., Fredrich, J.T.: Calculating the effective permeability of sandstone with multi-scale lattice Boltzmann/finite element simulations. *Acta Geotech.* **1**(4), 195–209 (2006)

You, C., Li, G., Zhang, Y., Zhang, X., Shan, H., Ju, S., Zhao, Z., Zhang, Z., Cong, W., Vannier, M.W., Saha, P.K., Wang, G.: CT super-resolution GAN constrained by the identical, residual, and cycle learning ensemble (GAN-CIRCLE). arXiv e-prints (2018) [arXiv:1808.04256](https://arxiv.org/abs/1808.04256)

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