

Learning a general model of single phase flow in complex 3D porous media

Original

Learning a general model of single phase flow in complex 3D porous media / Santos, J.E., Marcato, A., Kang, Q., Mehana, M., O'Malley, D., Viswanathan, H., Lubbers, N.. - In: MACHINE LEARNING: SCIENCE AND TECHNOLOGY. - ISSN 2632-2153. - 5:2(2024). [10.1088/2632-2153/ad45af]

Availability:

This version is available at: 11583/3010669 since: 2026-05-08T10:12:08Z

Publisher:

Institute of Physics - IOP

Published

DOI:10.1088/2632-2153/ad45af

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)



PAPER • OPEN ACCESS

Learning a general model of single phase flow in complex 3D porous media

To cite this article: Javier E Santos *et al* 2024 *Mach. Learn.: Sci. Technol.* **5** 025039

View the [article online](#) for updates and enhancements.

You may also like

- [Prediction of both diffusive and hydraulic conductance in the pore network model extracted from 3D images using deep learning](#)
Xiang Huang, Bin Lu, Zigang Yuan et al.
- [A ML framework to predict permeability of highly porous media based on PSD](#)
Haoyu Yang, Yan Ke and Duo Zhang
- [Coupling of Pore-Phase and Solid-Phase Networks for Simulating Heat and Mass Transport in Porous Transport Layers](#)
Vinaykumar Konduru and Jeffrey S Allen



PAPER

OPEN ACCESS

RECEIVED
25 October 2023REVISED
28 March 2024ACCEPTED FOR PUBLICATION
30 April 2024PUBLISHED
10 May 2024

Original Content from
this work may be used
under the terms of the
[Creative Commons
Attribution 3.0 licence](#).

Any further distribution
of this work must
maintain attribution to
the author(s) and the title
of the work, journal
citation and DOI.



Learning a general model of single phase flow in complex 3D porous media

Javier E Santos* , Agnese Marcato , Qinjun Kang, Mohamed Mehana, Daniel O'Malley, Hari Viswanathan and Nicholas Lubbers

Los Alamos National Laboratory, Los Alamos, NM, United States of America

* Author to whom any correspondence should be addressed.

E-mail: jesantos@lanl.gov**Keywords:** lattice-Boltzmann, porous media, Klinkenberg, nanoconfinement, complex geometries

Abstract

Modeling effective transport properties of 3D porous media, such as permeability, at multiple scales is challenging as a result of the combined complexity of the pore structures and fluid physics—in particular, confinement effects which vary across the nanoscale to the microscale. While numerical simulation is possible, the computational cost is prohibitive for realistic domains, which are large and complex. Although machine learning (ML) models have been proposed to circumvent simulation, none so far has simultaneously accounted for heterogeneous 3D structures, fluid confinement effects, and multiple simulation resolutions. By utilizing numerous computer science techniques to improve the scalability of training, we have for the first time developed a general flow model that accounts for the pore-structure and corresponding physical phenomena at scales from Angstrom to the micrometer. Using synthetic computational domains for training, our ML model exhibits strong performance ($R^2 = 0.9$) when tested on extremely diverse real domains at multiple scales.

1. Introduction

Estimating the transport properties of materials is challenging. On one hand, laboratory experiments are costly and require specialized facilities and in-hand processed samples that can not be re-used. On the other hand, numerical simulations often require complex meshing, assumptions and approximations, and are computationally intensive, especially for large complex domains. Alternatively, machine learning (ML) models are demonstrated to learn complex relations from large datasets with ease [1]. Most importantly, thanks to advances in software and hardware, ML models are able to provide predictions in new domains in milliseconds. This makes them attractive for forecasting physical processes, especially in the geosciences, where real-time predictions can be critical in field operations. However, training and testing ML models with subsurface data has many challenges. First, ML models are commonly optimized to 2D image benchmarks [2, 3]. On the other hand, imaging technologies (CT, FIB-SEM, MRI) have had breakthroughs [4] that allow us to obtain high definition samples that can be of size 256^3 or more; 3D domains are much more computationally challenging than most 2D images. Secondly, subsurface formations exhibit a wide range of complex structures due to the diverse rock genesis and diagenesis processes [5]. Finally, it is not straightforward to embed the complexity of varying degrees of confinement effects in the ML model. While much progress has been made in emulating transport in porous media using ML [6–15], models in the literature are not general enough to be used as a generic tool.

Fluid transport driven by an external force, for example, a pressure gradient or the gravitational force, is affected by viscous, slip, and diffusive effects depending on the characteristic length-scale of the available flow paths. The primary way to quantify the impact of these effects on the ability of a fluid to travel through a sample is the *apparent permeability*. This quantity provides a volume-average scalar that summarizes all the micro- and nano-scale effects into a single number, which can then be used in continuum simulators or analytical approximations for forecasting field-scale processes. A strict lower bound of the apparent

permeability is the *absolute permeability*, defined as the limit of the apparent permeability when nanoconfinement effects vanish. Slip and diffusive effects can enhance apparent permeability up to three orders of magnitude compared to absolute permeability, depending on the geometry and pressure (as described in [16] and shown in figure 2).

The structure and connectivity of the pore structures have the leading impact on the apparent permeability of a 3D porous medium. In micron-sized pores, fluid transport is approximated by *Darcy's Law* where viscous forces are dominant with negligible contributions from inertial forces. During viscous flow, the velocity profile of the fluid is non-uniform and affected by the pore morphology and pressure gradient. This non-uniformity stems from the viscous drag forces which are uneven across the flow paths. Fluids prefer the least-drag force paths. For a tube with a circular cross-section, the viscous forces are minimal at the center of the pore given rise to a parabolic-shaped velocity profiles. In contrast, in smaller pores, transport is affected by diffusion and slip at the solid-fluid boundary since the mean free path length of the molecule is comparable to or larger than the pore size, increasing molecular collisions [17] and enhancing permeability [18–20]. In nature, permeable media hosts conduits and heterogeneities in a wide range of length-scales. Subsurface formations have pores that can be a few Angstroms wide [21, 22] to fractures with apertures on the centimeter scale [23]. These wide range of scales encompass flow regimes going from free-molecular to viscous dominated [17]. These multiscale heterogeneities usually co-exist. For example, in fractured shale a fluid can flow much faster through the fracture, but the nanoporous matrix has a non-negligible contribution as shown by [24].

Laboratory experiments can be set-up to study deviations from Darcy's law. However, since the resolution of the instruments is on the same scale as the phenomenon under investigation, running an experiment at subsurface conditions is very challenging. In practice, samples are subject to extremely high pressure gradients [25–29] but this does not reflect the conditions that subsurface flow modellers are interested in. On the computational side, analytical solutions considering fluid flow inside a tube have been derived either from empirical observations or classical fluid dynamics and can be used with pore-network models [30]. Although these pore-network models try to capture the 3D connectivity of real samples, they aggressively oversimplify the heterogeneity of the pore space by fitting it with spheres and tubes. The high aspect ratio of the tubes will significantly influence the solution of non-Darcy flows and as seen in SEM images of mudrocks, pore structures vary from 'spongy' kerogen patches, to long and skinny microfractures, to stacks of platelets [31]. Finally, molecular dynamics simulations can represent nanoconfinement effects accurately. Nonetheless, the computational resources needed are vast, which limits the calculations to very small volumes (boxes with side lengths of around 10 nm).

The lattice-Boltzmann method (LBM) can operate in larger domains, which yield in representative sample sizes since it operates at a mesoscale. LBM integrates physical insights from both the molecular- and micron-scales and can predict larger-scale properties. In short, fluids are simulated as swarms of particles, represented by particle distribution functions, which flow on a discrete lattice. Landry [21] proposed the local effective viscosity model (LEV-LBM) to simulate multiple scales encompassing multiple flow regimes by utilizing a spatially-varying mean free path (sMFP). The sMFP accounts for the reduction of the local kinetic viscosity caused by the confinement effect at the nanoscale. One thing that sets apart the LBM simulation of that at the nanoscale, is that flow fields do not rely exclusively on the pore-structure but also on length scale, confinement pressure, temperature, and gas properties. The LEV-LBM also accounts for slip and diffusive effects depending on the scale. The main downside is that simulating a large complex sample with confinement effects could take weeks running on 1000 processors.

While the LBM offers detailed insights into fluid dynamics, there is a growing interest in leveraging this data and the computational efficiency of neural networks for permeability modeling. Neural networks are an attractive avenue for modeling permeability since they can provide predictions in a split-second enabling real-time field applications. Therefore, there have been many attempts to train neural networks to predict permeability from images. In a pioneering study, [32] employed a convolutional network (ConvNet) to learn a mapping between a 2D image and its permeability, since then ConvNets have remained the model of choice for this task. [33] provides a comprehensive review of notable works to this date. Nevertheless, computational scaling with domain size is still one of the main hurdles in training models with 3D images of permeable media. Recently, we provided a proof of concept of a network that allowed training and testing with large 3D images on a single graphics processing unit, capturing representative elementary volumes within the weights of the network [34].

In this letter, we present a *general model for fluid transport through permeable media* building upon the work of MS-Net [34] by improving the model design, scaling-up the parameter count (model size), and increasing the size and diversity of the training set. For the first time, we used hundreds of computationally large flow simulations in synthetic samples to train our model, including a wide range of pore sizes, rock types, and boundary conditions. The comprehensive dataset used for all the training and testing data are

publicly available [35]. To accommodate the larger model and dataset, we likewise applied a number of computer science tools to ameliorate the increased computational cost, which we detail. Our contributions are divided into two main categories, (1) model architecture, where we describe in detail the model that allow for multi-scale processes to be accounted for and (2) training, which is essential for distilling information from such a diverse set of samples properly and its vital for managing a model of this size and complexity. This workflow is general and transferable to other properties and domains. Finally, we show that the trained model is able to generalize the problem of fluid flow through permeable media for more than one thousand simulations, including many microCT images of real porous media.

2. Machine learning workflow

We describe the ML workflow in the following subsections. Dataset describes the data used for both training and testing. Model architecture describes the functional form of the ML model. Training algorithms explains the various techniques required to scale the training to large datasets and complex geometries, including the training schedule which describes the actual train procedure. An schematic showing all the stages of this process is shown in figure 1.

2.1. Dataset

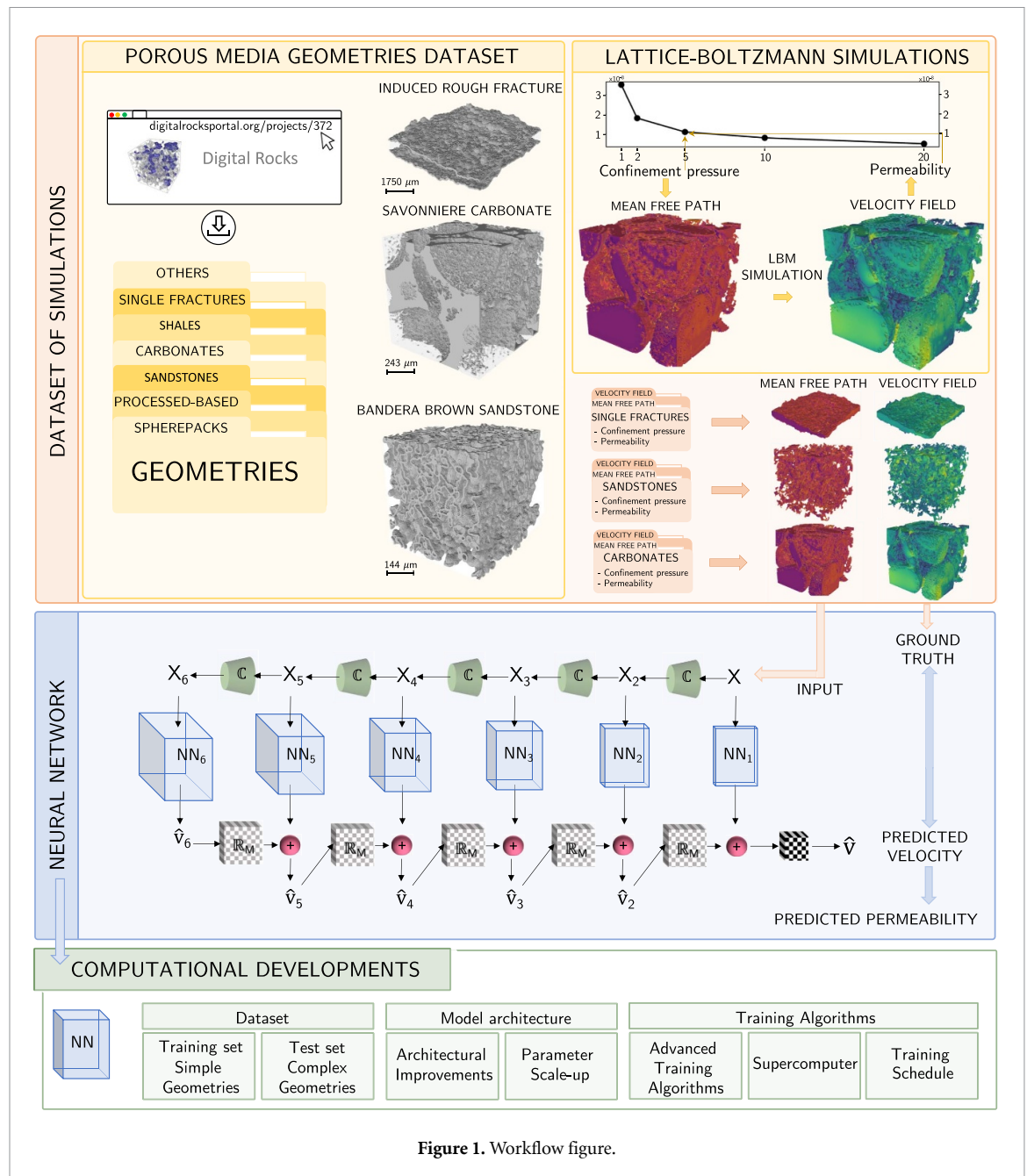
For a thorough review of the dataset, we refer the reader to [35]. The DRP-372 dataset includes more than 1200 flow simulations from a wide variety of samples hosted in the Digital Rocks Portal [36]. These simulations were carried out using the MP-LBM simulator [37]. We categorized the dataset based on 6 lithology groups: spherepacks, process-based geometries, shales, carbonates, sandstones, and single fractures. The flow simulations were carried-out in computational sizes of 256^3 and 480^3 where some of the latter took up to 4 d running on 1000 CPU cores to complete. Even with GPU acceleration, these models take approximately 10 million iterations to converge, so real-time execution is unfeasible. The resolution of these samples spanned from 0.5 nm to $5\ \mu\text{m}$. These simulations were carried-out using a local-effective viscosity lattice-Boltzmann model [21] which captures the impact of nanoconfinement effects, slip boundary conditions, and diffuse boundary layers when applicable. The confinement pressure was varied among individual simulations yielding a dataset that captures a wide range of scales and flow regimes (figure 2). The lower pressure bound is set at 1 MegaPascal (MPa) where significant nanoconfinement effects are present. The upper pressure was 20 MPa, where negligible confinement effects were shown and the solution is the Navier–Stokes equation solution. These simulations encompass flow regimes from the free molecular to viscous flow. The data spans 9 orders of magnitude of apparent permeability. The computational cost of these simulations required more than 5000 000 core-hours.

To train our model, we utilized 67 simple synthetic geometries described in [38], including various types of sphere packs, both homogeneous and heterogeneous, fractures, among others. These simple samples are designed to emulate various lithologies. These synthetic samples feature well-resolved pores and interconnected structures in the flow direction. The main motivation to use synthetic samples is to show that our model is able to abstract how fluid flow behaves under different physical conditions and is able to be used in a wide range of unseen real geometries. The training dataset includes 1 synthetic shale sample, 39 spherepacks, 20 synthetic fractures, 6 process-based vuggy spherepacks, and 1 carbonate.

Consequently, during testing, we evaluate the model's performance predominantly on real porous media or complex process-based reconstructions, highlighting its capacity to transfer knowledge from synthetic to real-world scenarios. This dataset includes 138 diverse samples from research groups worldwide [39–44], captured using various imaging devices or simulated using different process-based codes. The test set includes 11 shales, 7 fractures, 35 complex process-based geometries, 24 carbonates, and 61 sandstones. These were contributed by institutions including: Imperial College London, Australian National University, Penn State University, Statoil, Ghent University, UT-Austin, University of Calgary, Equinor, Carl Zeiss, Tescan XRE, IBM, Heriot-Watt University, Universite de Pau et des Pays de l'Adour, Xi'an Jiaotong University, and the Bureau Economic Geology. By leveraging this extensive and diverse dataset of almost 700 large, high performance simulation runs, our study rigorously evaluates the generalization capabilities of our model across a wide range of real-world samples.

2.2. Model architecture

Traditionally, analytical solutions have been proposed for permeability estimation using average (effective) structural properties, the most prominent example is the use of the porosity. Nevertheless, the average porosity of a sample is a non-unique proxy (many samples have the same average porosity value) and fails to capture the sample's pore-space complexity and spatial distribution. Another notable example is the fracture width used in the cubic law to estimate single-fracture permeability. To get a more complete picture, our



proposed ML model uses the whole complex 3D structure, with its thermodynamic conditions and scale. These are all included to aid the network to compute the resultant apparent permeability under these conditions. We explain in detail our network architecture in the following subsection.

We use the PyTorch library [48] as our ML backend. The main benefits that PyTorch provides are a dynamic computational graph and an intuitive interface that allows fast development. Our workflow is framework agnostic and could be ported to other modern libraries like JAX or TinyGrad with ease. We build our model using a larger version of the MS-Net [34] architecture, which demonstrated a remarkable ability to learn from large porous media samples. In short, the MS-Net is a system of ConvNets that process data at different resolutions. The system is trained end-to-end by passing the outputs of the coarser networks sequentially as inputs to the finer ones¹. There are two main motivating factors for using this architecture: (1) this is an elegant way to process large (>256³) geometries without subsampling the domain, and (2) the coarsest model can see the entire image at once and predict the bulk of the flow while the subsequent ones refine it. In this work, we have implemented the following enhancements to the previous model:

¹ The details about the coarsening and refinement procedure can be found in section 2.4 of [34].

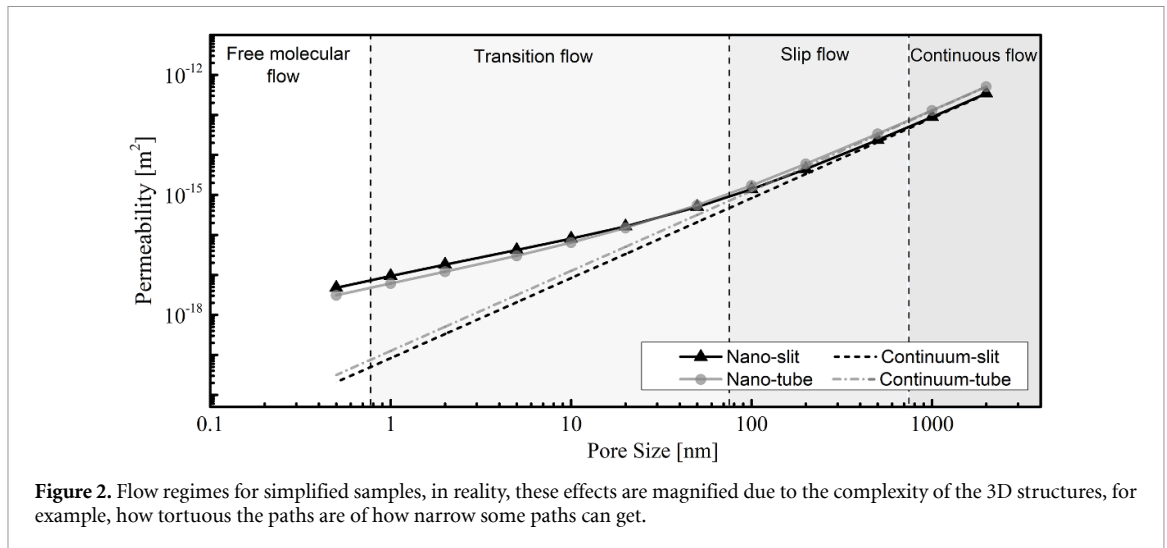


Figure 2. Flow regimes for simplified samples, in reality, these effects are magnified due to the complexity of the 3D structures, for example, how tortuous the paths are or how narrow some paths can get.

- Input features:** to capture simulations encompassing diverse thermodynamic conditions, we introduce a spatially-varying 3D normalized MFP in our model. This feature, which arises from the Peng–Robinson equation, accounts for the reduction of the local kinetic viscosity caused by the confinement effects at the nanoscale (the detailed explanation can be found in [35])². This input feature captures the resolution of the various samples, which is crucial when working with a heterogeneous dataset of geological materials. We also include the 3D Euclidean distance of the pore-space. The Euclidean distance labels each voxel with the distance to the closest solid boundary³. This feature aids to get a spatially-smooth prediction. The computational cost of these two features is negligible, even in very large domains. We opted not to add more input features as the model training was stable, and we sought to streamline the computational efficiency of the model.
- Large number of scales:** to significantly expand the model’s field of vision, we augmented our system by incorporating six convolutional sub-networks. This design decision enables the coarsest network (NN₆ in figure 1) to receive an input that is $\frac{1}{32}$ th the size of the original domain. Despite the downsizing, the input retains crucial global structural features that affect the fluid flow, which allows the kernels in the sixth network to see most of the image at once.
- Padding:** to facilitate the model’s comprehension of flow continuity along the flow direction, we incorporated periodic padding for the 3D convolution operations. Meanwhile, since the faces perpendicular to this axis feature no-flow boundaries in the simulation, we assigned zero-padding to these regions. These design choices are particularly valuable in the networks operating at coarser scales, as their field of vision is comparable to the input size (At the coarsest scale, a 256^3 domain corresponds has an input size of 8^3).
- No flow in the solid-space:** to explicitly ensure no flow within the solid space, we introduced an additional constraint at the end of our model. The final output of the model is multiplied by the input binary image, which contains ones in the pore-space and zeros in the solid-space. This multiplication effectively disregards the gradients of voxels outside the pores. By significantly reducing the search-space, this approach leads to notably faster training. Additionally, we conducted experiments where we allowed gradients of the first solid voxel surrounding the grains, but observed no discernible difference in the training process.
- Fewer activation functions and normalization layers:** to enhance accuracy, we implemented a strategy recommended in [45] by reducing the number of activation functions and normalization layers in our model. Specifically, we removed the normalization layer preceding the last convolutional layer, as well as the activation layer following it. This adjustment resulted in an improvement in accuracy.
- GeLU activation:** to align with recent advancements in models, we transitioned our remaining activations to Gaussian Error Linear Units (GeLU) [46]. GeLU serves as a smoother alternative to ReLU and has gained popularity in the field. By adopting GeLU, we achieved a slight increase in accuracy while maintaining the same computational overhead.

² Initially, we tried training our network with only this input. The trained model provided accurate predictions of permeability but the flow was not spatially-smooth, since this map does not have enough variance, as shown in figure 3.

³ The Euclidean distance outputs large values in the middle of the pores and smaller values near the solid boundaries, is amenable to the cylindrical profile of a fluid flowing on laminar flow.

- **Number of filters at each layer:** initially, we experimented with increasing the depth of each network to enhance model capacity. However, this approach proved challenging to train, likely due to increasingly difficult gradient paths. Consequently, we opted for a substantial increase in the number of parameters, significantly raising the count to 400 M compared to the 10 k of the MS-Net. This substantial capacity boost resulted in a significant improvement in accuracy. Notably, the augmented model's training process required adjustments to the training algorithm, as elaborated in section 2.3, to effectively handle the increased memory requirements.

In addition, we explored the inclusion of attention layers in the coarser networks. While this design choice led to slight improvements in accuracy, it also incurred significant computational costs. Furthermore, incorporating attention mechanisms posed challenges in accommodating different image sizes, making it less practical for our purposes. As a result, we opted for a fully convolutional architecture, which better addressed the need for flexibility in working with diverse image sizes.

2.3. Training algorithms

The development of an enhanced model, encompassing larger networks with increased parameter counts, and training it on computationally larger domains with more features, has yielded a notable improvement in accuracy for both training and testing datasets. However, scaling up the model presented significant challenges, particularly in addressing the memory constraints imposed by the GPUs. In this section, we emphasize the algorithms that were crucial in overcoming these obstacles and enabling the successful implementation of the model.

- **Mixed-precision training:** in order to accommodate the larger model and facilitate training with bigger domains, we carried out the training process using 16-bit precision. This approach offers several advantages, including reduced memory consumption as the network parameters occupy half the space, and accelerated data transfer due to operations requiring less bandwidth and executing at a faster pace. However, it is important to note that training stability can be compromised due to precision loss, which poses a significant challenge in our specific case.
- **Activation checkpointing:** to reduce GPU memory usage, we implemented activation checkpointing in our model. This technique avoids the storage of layer activations, recomputing them during the backward pass instead. By sacrificing some computation time, we achieve a reduction in memory requirements, allowing us to train larger models with the same memory footprint.
- **Gradient accumulation:** even with the utilization of 16-bit precision, the size of our model and samples restricts us from performing a backward pass with more than one domain at a time. Since training with a single sample per step (*online training*) would be extremely noisy, we use gradient accumulation to 'simulate' bigger batches. The idea behind gradient accumulation is simple: after calculating the loss and gradients for each sample, instead of immediately updating the model parameters, the gradients are accumulated across consecutive samples. The model parameters are then updated based on the cumulative gradient after a specified number of samples. This approach serves the same purpose as having a multiple sample mini-batch, all while maintaining the memory footprint of a single sample.
- **Loss function:** our dataset encompasses a wide range of scales, with permeabilities spanning 8 orders of magnitude and individual pixel-wise velocities spanning 24. To effectively convey the relative importance of each sample to the model, we employed a loss function that calculates the pixel-wise error relative to the variance of each velocity field. This approach ensures that every sample carries the same level of "importance" during training

$$\mathcal{L} = \sum_{i=1}^{g_{\text{steps}}} \sum_{s=1}^6 \frac{\text{sum} \left((u_{i,s} - \hat{u}_{i,s})^2 \right)}{N_{\text{vox},s} \cdot \sigma(u_i)^2} \quad (1)$$

g_{steps} are the number of gradient accumulation steps, u_i represent each of the 6 coarsened velocity fields, $N_{\text{vox},s}$ the number of voxels per scale and $\sigma(u_i)^2$ the standard deviation of the velocity field. Our loss function tries to reconstruct this. Solely computing the mean squared error in the last layer would assign greater importance to samples with higher velocities.

- **Input and output normalization:** To enhance training speed and stability, we normalized the Euclidean distance with a constant value of 50 and the velocity with a value of 1×10^{-9} . These value choices accelerated the training process and enhanced its stability. Additionally, we kept the normalized MFP in its original range of zero to one without further normalization.
- **Gradient clipping:** to mitigate potential instabilities during training, we employed gradient clipping, a technique that limits the magnitude of gradients to a specified value. By rescaling the error derivative, the

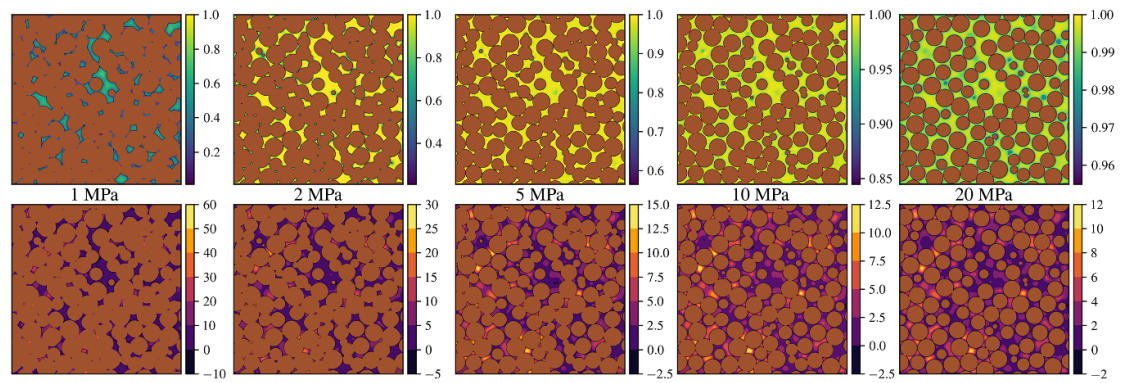


Figure 3. The bottom panel shows a cross-section of the nanoconfined simulation at five different pressures. The normalized velocity (with respect to the maximum value) contours are shown. It is visible that the sample at the lowest pressure (1 MPa) presents the higher degree of slip, shown as ‘fat’ velocity profile (fewer contour lines), while the one at 20 MPa exhibits no slip. This can also be seen in the top panels, where normalized mean free path (that could be interpreted as the deviation from the Navier–Stokes solution) is shown.

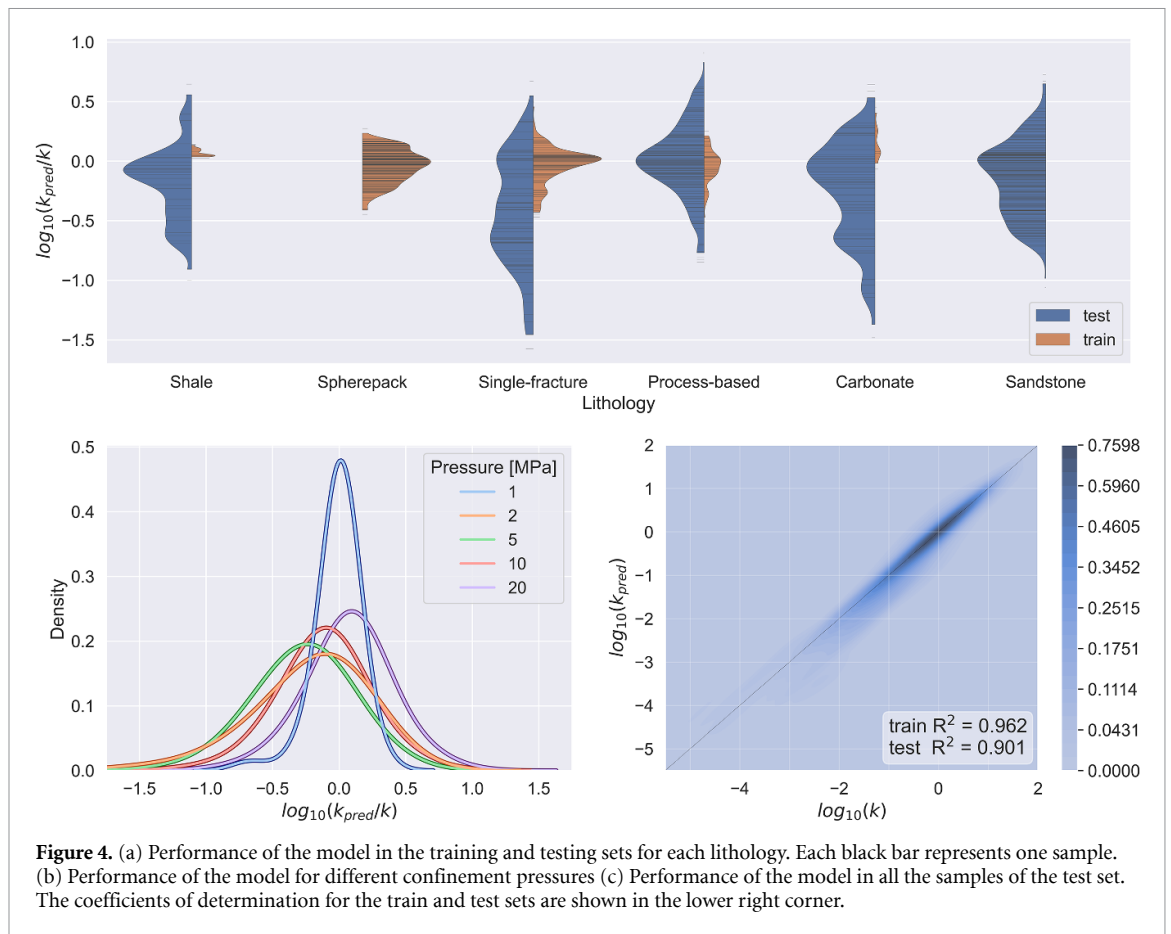
weight updates are also appropriately adjusted, minimizing the risk of overflow or underflow. This approach ensures that gradient descent exhibits stable behavior, even when faced with a complex and irregular loss landscape. By implementing gradient clipping, we successfully prevent training instabilities from adversely affecting the model’s training path.

- **Optimizer:** to enhance training stability and improve generalization to unseen data, we utilized the AdamW optimizer during model training. One key benefit of AdamW is the decoupling of weight decay from gradient updates, which encourages the model to maintain smaller weights, promoting stability throughout the training process and generalizing to unseen data better.

2.3.1. Training schedule

Given the significantly increased parameter count resulting from the strategies detailed in the previous section, combined with the diversity of the dataset, and the inherent instability of 16-bit training, network initialization plays a critical role. However, when using the weight initialization strategy proposed in [34], we faced challenges due to the wide range of orders of magnitude in the individual pixel-wise velocity samples, which lead to overflow or underflow issues shortly after a few training iterations. To address this issue, we devised a solution that involved ‘warming up’ the model weights using the simplest geometry, spherepacks. To ensure the model’s weight expressiveness in predicting a wide range of velocity outputs, we trained the model using five 256^3 spherepacks. These geometries covered porosity ranges of 8%–25% and pressures ranging from 1 to 20 Mpa (refer to figure 3). Although these spherepacks had similar permeability values, the velocities spanned a significantly broader range. This diverse training data enabled the model’s weights to capture the necessary complexity and variability required for training with more diverse samples. This training process, conducted over 72 h with a learning rate of 1×10^{-6} and a batch size of one, necessitated the incorporation of a constraint. If the loss exceeded 1×10^{-4} , the corresponding sample was ignored, and the network parameters were not updated, since very high losses triggered network updates results in the training diverging. This constraint, enforced through the PyTorch distributed data parallel engine, effectively identified and froze the unused parameters. While this constraint was frequently triggered during the early stages of training, it gradually diminished over time, and disappearing completely after a few h, allowing the training process to effectively adjust the weights.

After achieving stabilization of the loss function, we implemented a multi-stage training approach. Initially, we added one extra sample from each spherepack, selected randomly at a specific pressure. During this stage, we utilized an increased learning rate of 1×10^{-5} and a batch size of two, which continued for 24 h. Subsequently, we included all the samples sized 256^3 in the training process, further increasing the learning rate to 1×10^{-4} and expanded the batch size to 8 for an additional 24 h. Finally, in the last training stage, we included the remaining samples (the 480^3 volumes) and performed eight random augmentations (including a 180-degree flip in the flow axis and 90, 180, 270-degree flips on the XY-axes). This resulted in approximately 1000 samples of varying sizes to train on. We carried out this process for approximately 168 h. In the final stage, we fine-tune our model by incorporating stochastic weight averaging (SWA) to further enhance its performance. SWA offers notable advantages as it is easy to implement, improves generalization, and imposes minimal computational overhead. This aligns with the findings of [47], who also reported enhanced generalization capabilities through the use of SWA. Throughout this last stage, we utilized a 10%



validation split and saved the model with the best loss value observed in this set. To prevent overfitting, early stopping was employed after 300 epochs without improvement. The sample size used for training does not limit the computational sizes for inference. In fact, through padding techniques, our model can be utilized as a predictive model for any samples with sizes that fit in memory.

3. Results

Our primary focus is the apparent permeability of a 3D structure under varying confinement pressures. While it is possible to directly train a model to predict this quantity as a floating-point number, our preliminary experiments revealed a tendency for such models to overfit quickly and exhibit poor generalization to unseen data. Therefore, we trained our model using the complete flow field, even though our main interest lies only in the apparent permeability. In essence, the prediction of the flow field serves as an auxiliary task to support the training process. We would like to highlight the significant distinction in computational requirements between the creation of the training data and the testing phase. Generating the training data was accomplished with relative ease, requiring around 10 000 CPU hours due to the simplicity of the geometries. In contrast, the testing phase demanded a substantial investment of computational resources, totaling close to 5 million hours of CPU time.

We evaluated the performance of our model on unseen samples by quantifying the ratio between predicted and simulated apparent permeability (refer to figure 4). The model's performance was assessed across different lithologies and pressures. Remarkably, the model demonstrated excellent performance on unseen samples, despite being exclusively trained on synthetic data. We observed a well-defined normal distribution for the process-based samples in the test set, which closely resemble the training set. For the remaining lithologies, we observed strong agreement between the predicted and simulated permeability values. Notably, the single fracture lithology exhibited the lowest performance, likely due to its distinct characteristics such as roughness, tortuosity, and pinch points, which differ from the single-fracture counterparts in the training set. Confinement effects are magnified as pore size and pressure decrease, we noted that our model tends to exhibit stronger agreement at lower confinement pressures, suggesting that the input feature of MFP captures more salient features under such conditions. In terms of accuracy, most of the test samples demonstrate results within an order of magnitude. This achievement is particularly remarkable

considering that the permeability of the test data spans eight orders of magnitude and local phenomena encompasses regimes from free molecular to viscous dominated.

4. Conclusions

For decades, approximations such as the cubic law in fractures, variations of the Kozeny–Carman equation for granular media, and empirical equations based on laboratory measurements have been relied upon to estimate permeability. However, these approaches are known to have significant limitations and inherent flaws. In contrast, ML provides a promising avenue for permeability estimation, offering enhanced accuracy and improved performance. Our developed model represents a significant advancement in this field, setting a new state-of-the-art benchmark for permeability estimation.

We have developed a data-driven model that utilizes a large and diverse dataset to accurately predict the apparent permeability of complex pore networks. This model can be easily fine-tuned with new data, enabling improved accuracy for specific types of porous media, or can be directly utilized as is. To improve performance and avoid overfitting, we carefully optimized the model size and training methods. Our approach incorporates a comprehensive dataset, encompassing a wide range of sample sizes, features, and boundary conditions.

By employing a convolutional neural network, we trained our model on synthetic data and tested it on realistic heterogeneous samples, spanning various thermodynamic conditions, pore sizes, pore structures, and computational sample sizes. The versatility of this model lends itself to numerous practical and scientific applications. For instance, it can aid in establishing new relationships for fluid flow in porous media, and quantifying the effects of diagenetic processes over time. Since this model is extremely fast, The real-time prediction aspect could be exploited to optimize image segmentation parameters for binarization during x-ray scanning, estimate relative permeability curves while imagining an experiment, optimize porous media synthesis for specific properties, and permeability estimation from cuttings to complement logging while drilling, among others. In a broader view, our approach could be adopted for other phenomena and domains which will benefit from real-time predictions of commonly challenging-to-estimate properties.

Data availability statement

All the data used to train and test our model is described in [35] and it is openly-accessible at www.digitalrockportal.org/projects/372.

ORCID iDs

Javier E Santos  <https://orcid.org/0000-0002-2404-3975>

Agnese Marcato  <https://orcid.org/0000-0002-9934-9004>

References

- [1] LeCun Y, Bengio Y and Hinton G 2015 Deep learning *Nature* **521** 436–44
- [2] Deng J, Dong W, Socher R, Li L-J, Li K and Li F-F 2009 ImageNet: a large-scale hierarchical image database *2009 IEEE Conf. on Computer Vision and Pattern Recognition (IEEE)* pp 248–55
- [3] Krizhevsky A 2009 Learning multiple layers of features from tiny images
- [4] Wildenschild D and Adrian P S 2013 X-ray imaging and analysis techniques for quantifying pore-scale structure and processes in subsurface porous medium systems *Adv. Water Resour.* **51** 217–46
- [5] Mackenzie F T 2005 *Sediments, diagenesis, and sedimentary rocks* (available at: <https://shop.elsevier.com/books/sediments-and-diagenesis-and-sedimentary-rocks/mackenzie/978-0-08-044849-7>)
- [6] Marcato A, Boccardo G and Marchisio D 2022 From computational fluid dynamics to structure interpretation via neural networks: an application to flow and transport in porous media *Indust. Eng. Chem. Res.* **3** 8530–8541
- [7] Kashefi A and Mukerji T 2021 Point-cloud deep learning of porous media for permeability prediction *Phys. Fluids* **33** 9
- [8] Da Wang Y, Chung T, Armstrong R T and Mostaghimi P 2021 ML-LBM: predicting and accelerating steady state flow simulation in porous media with convolutional neural networks *Trans. Porous Med.* **0123456789** 4
- [9] Ting A K, Santos J E and Gultinan E 2022 Using machine learning to predict multiphase flow through complex fractures *Energies* **15** 8871
- [10] Cawte T and Bazylak A 2022 A 3D convolutional neural network accurately predicts the permeability of gas diffusion layer materials directly from image data *Curr. Opin. Electrochem.* **35** 101101
- [11] Caglar B, Broggi G, Ali M A, Orgéas L and Michaud V 2022 Deep learning accelerated prediction of the permeability of fibrous microstructures *Composites A* **158** 106973
- [12] Zhang H, Yu H, Yuan X, Xu H, Micheal M, Zhang J, Shu H, Wang G and Wu H 2022 Permeability prediction of low-resolution porous media images using autoencoder-based convolutional neural network *J. Pet. Sci. Eng.* **208** 109589
- [13] Tang P, Zhang D and Li H 2022 Predicting permeability from 3D rock images based on CNN with physical information *J. Hydrol.* **606** 127473

- [14] Marcato A, Santos J E, Liu C, Boccardo G, Marchisio D and Franco A A 2023 Modeling the 4D discharge of lithium-ion batteries with a multiscale time-dependent deep learning framework *Energy Storage Mater.* **63** 102927 (available at: www.sciencedirect.com/science/article/pii/S2405829723003057)
- [15] Chang B, Santos J and Prodanovic M 2022 IRock-Net: Assessing the Utility of Machine Learning to Initialize 3D Electric Potential Simulations *Society of Core Analysisists*
- [16] Yin Y, Zhiguo Q, Prodanović M and Landry C J 2022 Identifying the dominant transport mechanism in single nanoscale pores and 3D nanoporous media *Fundam. Res.* **3** 409–21
- [17] Javadpour F, Fisher D and Unsworth M 2007 Nanoscale gas flow in shale gas sediments *J. Can. Pet. Technol.* **46** 55–61
- [18] Klinkenberg L J 1941 The permeability of porous media to liquids and gases *Drilling and Production Practice* vol 1941 pp 200–13
- [19] Javadpour F 2009 Nanopores and apparent permeability of gas flow in mudrocks (Shales and Siltstone) *J. Can. Pet. Technol.* **48** 16–21
- [20] Song W, Yao J, Yang Li, Sun H, Zhang L, Yang Y, Zhao J and Sui H 2016 Apparent gas permeability in an organic-rich shale reservoir *Fuel* **181** 973–84
- [21] Landry C J, Prodanović M and Eichhubl P 2016 Direct simulation of supercritical gas flow in complex nanoporous media and prediction of apparent permeability *Int. J. Coal Geol.* **159** 120–34
- [22] Loucks R G, Reed R M, Ruppel S C and Hammes U 2012 Spectrum of pore types and networks in mudrocks and a descriptive classification for matrix-related mudrock pores *AAPG Bulletin* **96** 1071–98
- [23] Marrett R, Ortega O J and Kelsey C M 1999 Extent of power-law scaling for natural fractures in rock *Geology* **27** 799
- [24] Chen Li, Fang W, Kang Q, De'Haven Hyman J, Viswanathan H S and Quan Tao W 2015 Generalized lattice Boltzmann model for flow through tight porous media with Klinkenberg's effect *Phys. Rev. E* **91** 033004
- [25] Tison S A 1993 Experimental data and theoretical modeling of gas flows through metal capillary leaks *Vacuum* **44** 1171–5
- [26] Marino L 2009 Experiments on rarefied gas flows through tubes *Microfluid. Nanofluid.* **6** 109–19
- [27] Anovitz L M and Cole D R 2015 Characterization and analysis of porosity and pore structures *Rev. Mineral. Geochem.* **80** 61–164
- [28] Curtis M E, Sondergeld C H, Ambrose R J and Rai C S 2012 Microstructural investigation of gas shales in two and three dimensions using nanometer-scale resolution imaging *AAPG Bulletin* **96** 665–77
- [29] Sondergeld C H, Ambrose R J, Rai C S and Moncrieff J 2010 Micro-structural studies of gas shales *All Days SPE* (<https://doi.org/10.2118/131771-MS>)
- [30] Mehmani A, Prodanović M and Javadpour F 2013 Multiscale, multiphysics network modeling of shale matrix gas flows *Transp. Porous Med.* **99** 377–90
- [31] Rui X, Prodanovic M and Landry C 2020 Pore scale study of water adsorption and subsequent methane transport in clay in the presence of wettability heterogeneity
- [32] Jinlong W, Yin X and Xiao H 2018 Seeing permeability from images: fast prediction with convolutional neural networks *Sci. Bulletin* **63** 1215–22
- [33] Da Wang Y, Blunt M J, Armstrong R T and Mostaghimi P 2021 Deep learning in pore scale imaging and modeling *Earth-Sci. Rev.* **215** 103555
- [34] Santos J E, Yin Y, Honggeun J, Pan W, Kang Q, Viswanathan H S, Prodanović M, Pycrz M J and Lubbers N 2021 Computationally efficient multiscale neural networks applied to fluid flow in complex 3D porous media *Trans. Porous Med.* **140** 241–72
- [35] Santos J E, Chang B, Gigliotti A, Yin Y, Song W, Prodanović M, Kang Q, Lubbers N and Viswanathan H 2022 A dataset of 3D structural and simulated transport properties of complex porous media *Sci. Data* **9** 579
- [36] Prodanovic M, Esteve M, Hanlon M, Nanda G and Agarwal P 2015 Digital rocks portal: a repository for porous media images
- [37] Santos J E, Gigliotti A, Bihani A, Landry C, Hesse M A, Pycrz M J and Prodanović M 2022 MPLBM-UT: Multiphase LBM library for permeable media analysis *SoftwareX* **18** 101097
- [38] Santos J E, Pycrz M J and Prodanović M 2022 3D Dataset of binary images: a collection of synthetically created digital rock images of complex media *Data Brief.* **40** 107797
- [39] Bultreys T and De Boever W 2020 Belgian Fieldstone
- [40] Bultreys T 2018 Luc Van Hoorebeke, and Veerle Cnudde (Estailades Carbonate #2)
- [41] Van Offenwert S, Bultreys T and Cnudde V 2019 Saturated solute transport micro-CT dataset in sintered glass and Bentheimer sandstone
- [42] Spurin C, Krevor S, Blunt M and Bultreys T 2021 Decane and brine injected into Estailades carbonate-steady-state experiments
- [43] Armstrong R and Mostaghimi P 2017 Moura Coal
- [44] Singh K and Blunt M J 2018 High resolution x-ray micro-tomography datasets for in-situ effective contact angle analysis in carbonate rocks
- [45] Liu Z, Mao H, Chao-Yuan W, Feichtenhofer C, Darrell T and Xie S 2022 Facebook AI Research a convnet for the 2020s (arXiv:2201.03545)
- [46] Hendrycks D and Gimpel K 2016 Gaussian error linear units (GELUs) (arXiv:1606.08415)
- [47] Song H-J, Roh S, Lee J, Nam G, Yun E, Yoon J and Sa Kim P 2022 Benefits of stochastic weight averaging in developing neural network radiation scheme for numerical weather prediction *J. Adv. Mod. Earth Syst.* **14** e2021MS002921
- [48] Paszke A et al 2019 PyTorch: an imperative style, high-performance deep learning library *Advances in Neural Information Processing Systems*