Advances in
Geo-Energy Research

Vol. 10, No. 1, p. 39-55, 2023

Original article

Pore-GNN: A graph neural network-based framework for predicting flow properties of porous media from micro-CT images

Mohammed K. Alzahrani, Artur Shapoval, Zhixi Chen, Sheikh S. Rahman*

School of Mineral and Energy Resources Engineering, University of New South Wales, Kensington NSW 2033, Australia

Keywords:
Porous media
fluid flow modelling
artificial intelligence
deep learning
graph neural networks

Cited as:
Alzahrani, M. K., Shapoval, A., Chen, Z., Rahman, S. S. Pore-GNN: A graph neural network-based framework for predicting flow properties of porous media from micro-CT images. Advances in Geo-Energy Research, 2023, 10(1): 39-55. https://doi.org/10.46690/ager.2023.10.05

Abstract:
This paper presents a hybrid deep learning framework that combines graph neural networks with convolutional neural networks to predict porous media properties. This approach capitalizes on the capabilities of pre-trained convolutional neural networks to extract n-dimensional feature vectors from processed three dimensional micro computed tomography porous media images obtained from seven different sandstone rock samples. Subsequently, two strategies for embedding the computed feature vectors into graphs were explored: extracting a single feature vector per sample (image) and treating each sample as a node in the training graph, and representing each sample as a graph by extracting a fixed number of feature vectors, which form the nodes of each training graph. Various types of graph convolutional layers were examined to evaluate the capabilities and limitations of spectral and spatial approaches. The dataset was divided into 70/20/10 for training, validation, and testing. The models were trained to predict the absolute permeability of porous media. Notably, the proposed architectures further reduce the selected objective loss function to values below 35 mD, with improvements in the coefficient of determination reaching 9%. Moreover, the generalizability of the networks was evaluated by testing their performance on unseen sandstone and carbonate rock samples that were not encountered during training. Finally, a sensitivity analysis is conducted to investigate the influence of various hyperparameters on the performance of the models. The findings highlight the potential of graph neural networks as promising deep learning-based alternatives for characterizing porous media properties. The proposed architectures efficiently predict the permeability, which is more than 500 times faster than that of numerical solvers.

1. Introduction

Understanding fluid flow processes in porous media plays a crucial role in various fields, as they directly affect various industrial applications, such as contaminant treatment in ground water aquifers (Essaid et al., 2015), hydrocarbon recovery (Guo et al., 2019; Shapoval et al., 2022, 2023), and geologic carbon storage and sequestration (Iglauer et al., 2015; Ershadnia et al., 2020). These applications rely on the accurate characterization of the properties of porous media, which encompass a range of rock and fluid properties. In recent years, substantial efforts have been dedicated to studying both the macroscopic and microscopic properties of the subsurface (Ding et al., 2019, 2023; Wang et al., 2019) to optimize resources and design efficient subsurface engineering systems. Accurate characterization of porous media properties is of paramount importance, as it provides valuable insights that can lead to the development of innovative techniques for geothermal energy extraction (Ijeje et al., 2019), groundwater management, and remediation (Al-Hashimi et al., 2021), as well as benchmarking enhanced oil recovery techniques (Brantson et al., 2020; Shapoval et al., 2022).

Porous media properties can be broadly classified into statistical and dynamic properties. Statistical properties such as porosity, tortuosity, average pore size, and specific surface

Yandy
Scientific
Press

*Corresponding author.
E-mail address: mohammed_khaled_a.alzahrani@unsw.edu.au (M. K. Alzahrani); a.shapoval@unsw.edu.au (A. Shapoval); zhixic@unsw.edu.au (Z. Chen); sheik.rahman@unsw.edu.au (S. S. Rahman).

2207-9963 © The Author(s) 2023.
Received July 29, 2023; revised August 22, 2023; accepted September 13, 2023; available online September 20, 2023.

area provide insights into the geometric characteristics of the porous media. These properties can be determined through direct laboratory measurements and micro-computed tomography (micro-CT) image analysis techniques at the micro- and micro-scale, respectively. On the other hand, dynamic properties refer to properties of porous media that relate to how fluids interact, move through, and are influenced by the porous structure over time. One of the most important dynamic properties is permeability, which characterizes the ability of porous medium to transmit fluids within the porous structures. The formation and evolution of permeability are influenced by the various geological conditions under which subsurface rocks exist (Cheng et al., 2021).

The accurate quantification of permeability is challenging because of the complex and heterogeneous nature of porous media. Experimental approaches such as core-flooding techniques (Perrin et al., 2009; Hussain et al., 2021) offer direct measurements of core-scale permeability; however, they may not capture the full complexity of the studied porous medium. At the pore scale, permeability can be computed using simplified techniques, such as pore network modelling, which aims to represent porous media in terms of an idealized network of balls (pores) and sticks (throats) (Ryazanov et al., 2010; Gostick et al., 2019; Gerke et al., 2020). This allows for an efficient numerical simulation of fluid flow and rapid permeability computation. Despite their efficiency, permeability computed using PNMs usually deviates from more complex approaches because of the inherent simplification of these methods. More complex and computationally demanding numerical simulation techniques, such as finite element/volume methods (Wang et al., 2021; Hussain et al., 2023), smoothed particle hydrodynamics (Mohammadi and Riazi, 2022), and lattice Boltzmann method (LBM) (Akai et al., 2020; Santos et al., 2022), have been successfully employed to estimate the pore-scale permeability of subsurface rocks by solving the Navier-Stokes equation of flow. These simulation methods provide insights into fluid flow behavior, namely fluid velocity fields, within porous structures, which allows for the estimation of permeability with high accuracy under different conditions.

Recent advancements in artificial intelligence, specifically deep learning (DL), have enabled researchers to build robust models for various challenging computational tasks, including but not limited to image classification (Deng et al., 2009), object detection (Redmon et al., 2016), semantic segmentation (Ronneberger et al., 2015), and video analysis (Tran et al., 2015). More specifically, DL models have shown great potential as surrogate models for characterizing porous media properties. This is because of the capabilities of deep learning models, mostly convolutional neural networks (CNNs), to extract meaningful features from micro-CT images. These models can capture intricate patterns and relationships within the data, enabling the training of CNN models for various classification and regression tasks related to porous media. For example, multi-mineral classification can be accomplished by deploying semantic segmentation using Deep CNNs (Wang et al., 2020; Li et al., 2022). Moreover, researchers have been able to rapidly classify the wettability of porous media surfaces from two dimensional (2D) images of micromodels (Yun et al., 2020). In terms of regression tasks, most implementations of CNN for porous media characterizations relate to image-based prediction of statistical and dynamical properties (Alqahtani, 2018, 2020; Graczyk and Matyka, 2020). Additionally, researchers have shown that incorporating physics-based approaches, in which the model is fed with additional physical features, can enhance the accuracy of CNN models (Alqahtani et al., 2021; Gärttner et al., 2023). Other approaches, such as PoreFlow-Nets (Santos et al., 2020), aim to regress the velocity field values instead of directly predicting macroscopic properties. Recently, a comprehensive literature review was published highlighting the latest advances in the application of CNN-based approaches to pore-scale porous media modelling (Li et al., 2023).

Despite the evident popularity of CNNs for the image-based regression of porous media properties, they are restricted to operating on uniform grids and require training inputs of fixed dimensions. Techniques such as resizing and cropping are not valid, as they cause loss of image features, which are important when evaluating properties, especially dynamical properties. DL methods that operate on unstructured data and hence accept variable input sizes can provide an alternative that can mitigate some of the pitfalls of CNNs. For example, PointNet is a pioneering deep learning architecture designed to directly process point cloud data without the need for voxelization or other preprocessing steps. It captures the spatial distributions and properties of point clouds by applying a series of symmetric functions, and has been widely used in tasks such as object classification (Qi et al., 2017) and recognition (Wu et al., 2020) in three dimensional (3D) datasets. With respect to pore-scale modelling, a PointNet-based approach has been proposed to predict porous media properties from point cloud representations of porous media (Kashefi and Mukerji, 2021). In this approach, the authors extracted the point cloud coordinates of the rock surface from binary images and used them to train the PointNet architecture to regress the permeability of the input point clouds. Another DL approach that has gained popularity in recent years is graph neural networks (GNNs), which can learn from graph-structured data. These networks have been utilized in various computer vision tasks such as segmentation (Liu et al., 2021), scene analysis, scene generation, and motion detection and tracking (Ashual and Wolf, 2019; Liang et al., 2020). The applicability of GNNs can be extended to almost any problem that can be modelled (represented) by graphs or a graph of interconnected networks of nodes and edges, such as problems related to quantum and molecular chemistry (Gilmer et al., 2017), biological network analysis (Ding et al., 2023), and web-scale recommendation systems (Ying et al., 2018). Mesh-based GNNs have also been implemented to accurately model various complex physical processes such as fluid flow dynamics, airfoil aerodynamics, and mechanical deformation (Pfaff et al., 2021). At the reservoir scale, GNNs have recently been employed for reservoir-scale simulations of multiphase fluid flow (Jiang and Guo, 2023). The first study to investigate the applicability of GNNs in pore-scale modelling of porous media was conducted by Cai et al. (2023). The researchers

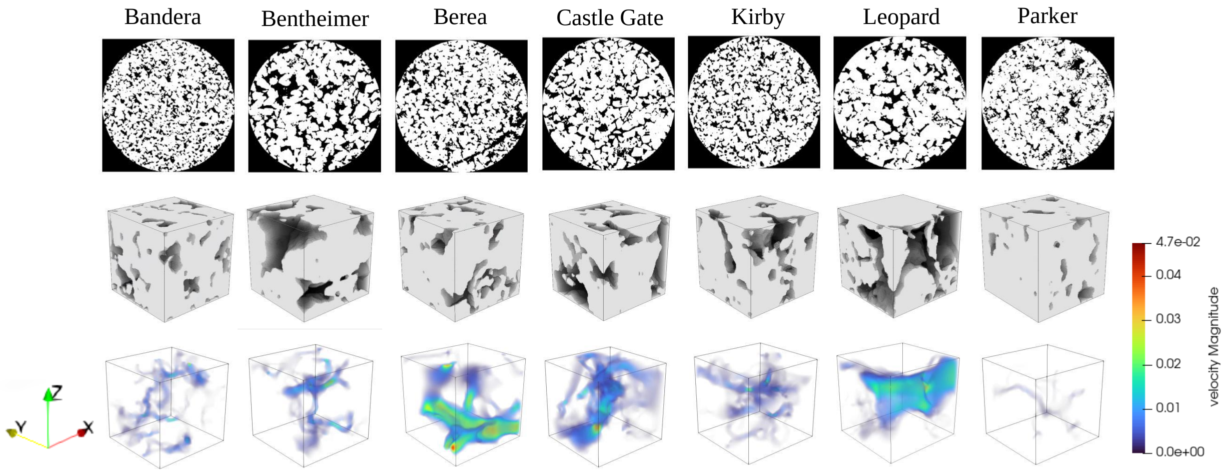

Fig. 1. Visualization of the selected sandstones showing (top) 2D slices of the binary rock image of size $1,000^2$ pixels, where 1 = rock matrix (white) and 0 = pore (black), (middle) 3D sub-volumes of size 100^3 cubic voxels, where the brighter regions represent the rock and the darker, but slightly transparent, regions represent the pore space, and (bottom) velocity fields throughout the corresponding pore space obtained from the LBM flow simulations.

Table 1. Summary of the properties of the seven sandstone rocks selected for model training.

Rock	Label	Porosity (%)	Tortuosity	SSA (1/μm)	APS (μm^3)	K_{exp}(mD)
Bandera	BANDERA	18.56	1.72	0.0637	15,088	386
Bentheimer	BH	22.64	1.64	0.0469	58,499	86
Berear	BEREA	18.56	1.96	0.0533	23,543	121
Castle gate	CG	26.54	1.33	0.0530	37,067	269
Kirby	KIRBY	19.95	1.59	0.0293	21,830	62
Leopard	LEO	20.22	2.00	0.0381	38,058	327
Parker	PAR	14.77	2.12	0.0466	13,445	10

represented pore space in terms of persistence-based Morse graph representations and trained equivariant graph neural networks to predict the formation factor and effective permeability tensors.

The main contribution of this work is to present Pore-GNN, a novel GNN-based framework for predicting the properties of porous media. The proposed architectures demonstrate that by leveraging the powers of pre-trained CNNs as backbone feature extractors, image features can be embedded into graph-structured data, and GNN models composed of various graph convolutional layers (GConv) can be trained to predict sub-sample permeabilities. In addition, two approaches for graph construction were investigated: sample as a node and sample as a graph, and for GNN training, node-level, and graph-level regression tasks. The models were validated by comparing the model prediction to the ground truth values of permeability obtained by running single-phase LBM flow simulations on the entire dataset. In addition, the performance of the models was compared with that of a baseline CNN in terms of predicting the permeability of unseen rock sandstone and carbonate samples. This work demonstrates that building GNNs on top of CNN feature extractors can further improve model prediction accuracy. Finally, a sensitivity analysis was conducted to discuss the effect of hyper-parameters on the performance and accuracy of the proposed models.

2. Dataset

A dataset of different sandstone rock micro-CT images published on the Digital Rocks Portal (Prodanovic et al., 2015) was utilized to train the models, which originally contained 11 different sandstone rock samples (Lucas-Oliveira et al., 2020; Neumann et al., 2021). Seven of these rocks were chosen, which had varying ranges of porous media properties. The original micro-CT images had a size of 100^3 voxels and a resolution of 2.25 μm^3/voxel. Raw grayscale images were processed by the publisher and segmented into binary images with a rock matrix = 1 and pore = 0. Typically, to train deep neural networks for porous media property predictions from 3D input images, the original images are cut into subsamples (cubes) of smaller sizes, ranging from 80^3 to 130^3 voxels, depending on the available memory and computational resources (Santos et al., 2020; Alqahtani et al., 2021). Therefore, the selected input image size for the experiment was 100^3 voxels, resulting in 1,000 subsamples for each full-rock image (7,000

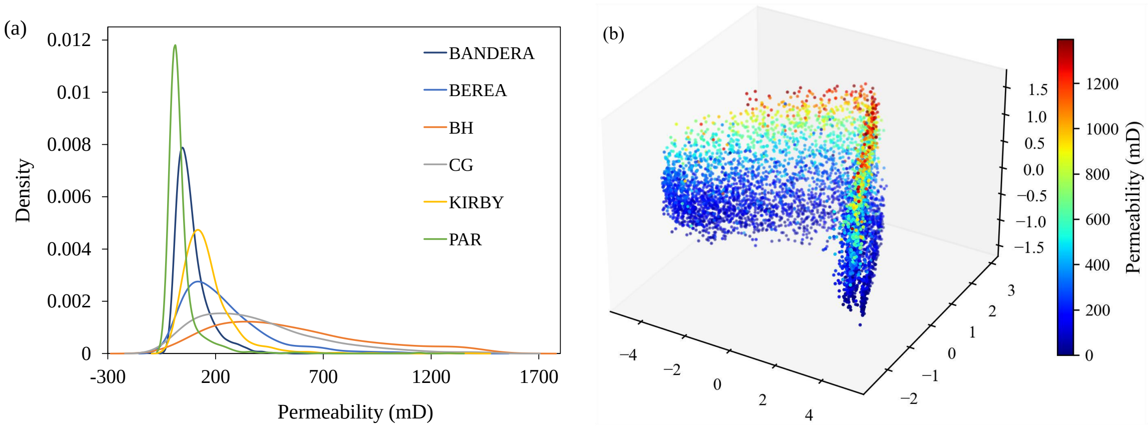

Fig. 2. Illustrations of (a) distribution of permeability of subsamples of size 100^3 cubic voxels of each rock type and (b) reduced features to three arbitrary dimensions using Principal Component Analysis.

images in total). A visualizations of the dataset is shown in Fig. 1. The properties of the seven sandstone rocks are shown in Table 1.

3. Methods

In this section, the numerical simulation method of fluid flow physics, CNN feature extractor and baseline CNN, training graph construction approaches, and proposed Pore-GNN architectures are highlighted.

3.1 Numerical fluid flow simulation

Direct numerical simulations were performed using LBM, which is one of the most popular methods for simulating fluid flow through irregular porous media domains (d'Humires, 2002; Akai et al., 2020; Santos et al., 2022). This method provides a solution for the Navier–Stokes flow equation by simulating the streaming and collision of fluid particles on a grid. The Boltzmann equation is expressed as:

$$\frac{\partial f}{\partial t} + \underline{v}\frac{\partial f}{\partial \underline{x}} + \underline{F}\frac{\partial f}{\partial \underline{v}} = \left(\frac{\partial f}{\partial t}\right)_{\text{coll}} \tag{1}$$

where f is the probability density function (PDF) of the bulk molecules, $\underline{x}$ is the position vector, $\underline{v}$ is the velocity vector, $\underline{F}$ is the external force vector, t is time, and $(\partial f/\partial t)_{\text{coll}}$ is the molecular collision term. The equation models the change in the PDF over time by allowing each node of the domain to host the fluid particles represented by the PDF, which can be used to calculate the velocity field throughout the porous domain. Typically, either the Bhatnagar-Gross-Krook collision term (Bhatnagar et al., 1954) or the Multiple-Relaxation-Time collision term (d'Humires, 2002) can be applied, which are responsible for relaxing the fluid distribution functions towards the equilibrium state at each lattice node.

The simulation of fluid flow in each porous subsample was performed in the x-direction and under laminar flow conditions, restricting the Reynolds number to less than one. In addition, small constant lattice pressure boundaries were imposed. The converged LBM solution represents the steady-state flow, and the average velocity of the resulting velocity fields, as shown in Fig. 1 (bottom), can be used to solve the Darcy equation for permeability, as shown in the following equation:

$$k = \frac{\overline{v}}{\mu}\frac{\mathrm{d}p}{\mathrm{d}x} \tag{2}$$

where k is permeability, $\overline{v}$ is the average velocity in the direction of the pressure gradient $\mathrm{d}p/\mathrm{d}x$, and μ is the dynamic viscosity of the fluid. Permeability has units of length squared and is typically measured in m^2 or Darcy (D). In this equation, x represents the length of the sample, in contrast to $\underline{x}$ in Eq. (2), which is a vector representing to the spatial position in the simulation domain. The simulation was run under the same conditions for all 7,000 subsamples and their permeabilities were calculated. The distributions of the subsample permeabilities for each rock type are shown in Fig. 2(a). These rock types and subsample sizes were purposefully selected to allow a wide range of permeability values to be used as labels to build a comprehensive training dataset.

3.2 Convolutional neural networks

Deep CNNs have revolutionized artificial intelligence model capabilities for computer vision tasks, such as image classification, facial recognition (Taigman et al., 2014), speech recognition (Alsobhani et al., 2021), and natural language processing (Wang and Gang, 2018). Moreover, numerous studies have been shown CNNs capabilities to model porous media properties and serve as efficient substitutes for traditional numerical modeling approaches (Alqahtani et al., 2020, 2021; Rabbani et al., 2020; Santos et al., 2020). They operate on uniform grids, that is, with constant spacing between all adjacent grid units, which makes them suitable for processing 1D series, 2D images, and 3D volumes.

The core concept of CNNs relies on the application of convolution operations that involve sliding a small filter across the input data, known as a kernel. The filter performs element-wise multiplication with a local receptive field, as it slides over the input data, resulting in trainable feature maps. Typically, CNNs are composed of multiple convolutional layers stacked

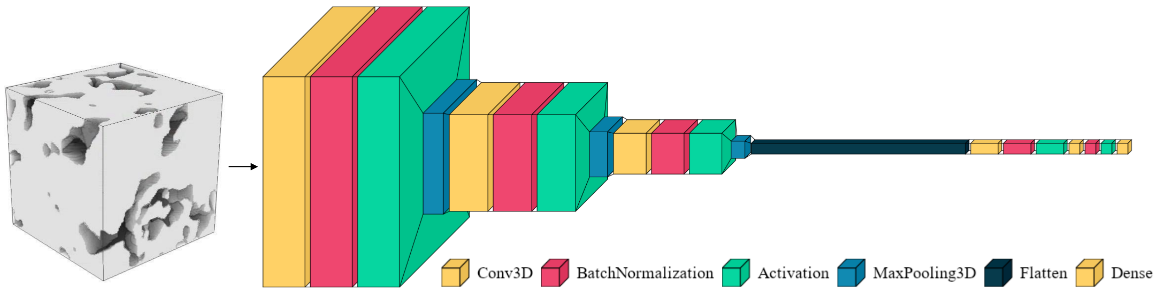

Fig. 3. Architecture visualization of baseline CNN.

Table 2. Architecture of baseline CNN used to predict permeability values.

Layer	Type	Input size	Kernel	Options	Trainable parameters
1	Input	100 × 100 × 100 × 1	/	Normalization	0
2	Convolutional	96 × 96 × 96 × 8	3 × 3 × 3	Max pool 5 × 5 × 5 + BN	1,024
3	Convolutional	42 × 42 × 42 × 16	3 × 3 × 3	Max pool 5 × 5 × 5 + BN	16,048
4	Convolutional	15 × 15 × 15 × 32	3 × 3 × 3	Adaptive max pooling	64,096
5	Fully-connected	6 × 6 × 6 × 32	/	ReLU activation + BN	3,539,456
6	Fully-connected	1 × 1 × 512	/	ReLU activation + BN	66,688
7	Output	1 × 1 × 128	/	Denormalization	129
Total	/	/	/	/	3,623,345

Notes: All layers, except the input and output layers, include ReLU activation and an optional batch normalization (BN) layer.

on top of one another. A rectified linear unit (ReLU), batch normalization, and pooling operations are commonly used to reduce possible overfitting, owing to the high model complexity as the number of layers increases. Fully connected (FC) layers can be used to flatten the resulting feature maps and feed them into a one-dimensional vector of the required size.

Fig. 3 shows the visualization of the architecture of the baseline CNN layers. Further details regarding the dimensions and the number of trainable dimensions of each layer are presented in Table 2, which was trained to predict the permeability of porous media using binary rock images as the input. The model comprises a series of convolutions, ReLU, batch normalization layers, max-pooling layers connected to an FC layer, and a prediction head that predicts a single permeability value. The convolutional layers produce an increasing number of feature maps as the network deepens, whereas the pooling layers reduce the spatial dimensionality. The last convolutional layer is followed by an adaptive max-pooling layer that pools the input feature map into the desired output size, in Table 2, the output size is [6, 6, 6, 32]. Moreover, the last FC layer before the prediction head is designed to produce output vectors of shape 128. These design choices were applied so that the trained CNN could be utilized as a feature extractor for embedding the initial node representation of the training graphs, for different regression tasks, as explained in the following section. Fig. 2(b) shows an illustration of the training dataset obtained by (1) removing the output layer from the pre-trained baseline CNN, (2) running the trained baseline CNN on the entire dataset, resulting in 128-dimensional vectors for all samples in the dataset and, (3) reducing the dimensionality of each sample (node) feature to three arbitrary dimensions using principal component analysis (Pearson, 1901; Jolliffe and Cadima, 2016). Finally, the 3D scatter points were plotted and colored based on the corresponding permeability values of the sample in mD. As shown in the figure, two linear trends can be observed, within which data exist in the reduced space. Moreover, there was a clear separation between clusters based on the permeability values, indicating nonarbitrary positioning within the reduced 3D spaces.

3.3 Training graphs construction

The novelty of this GNN-based approach lies in the alteration of the input data representation from 3D grid-structured to graph-structured data. Two approaches for constructing training graphs were examined: 1) sampling as a node and 2) sampling as a graph. This was performed to allow building of GNN models that can be trained for both node- and graph-level regression tasks, respectively. The models are explained in detail in the next section.

In the first approach, the 128-dimensional output of the last FC layer of the pre-trained CNN was generated and used as the input node (sample) features. A graph of the entire dataset is created by constructing edges between nodes in 128-dimensional space using the K-nearest neighbor (KNN)

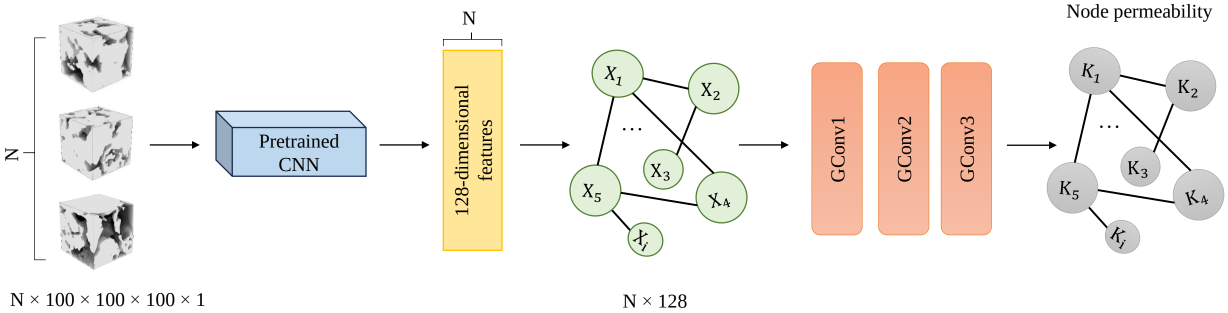

Fig. 4. Illustration of node-level Pore-GNN architecture. All binary images in the dataset of size N were fed into the pre-trained CNN model to extract 128-dimensional feature vectors for each image. A training graph was constructed by embedding 128 features as the initial node embeddings. The three GConv layers were trained to transform the initial embeddings into final node embeddings (permeability).

Table 3. Summary of the four graph convolutional layers used to build GNN-based models.

Type	Name	Update rule	Reference
Spectral	GCN	$X' = \hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}\Theta$	Kipf and Welling (2017)
	ChebNet	$X' = \sum_{k=1}^{k} Z^{(k)} \cdot \Theta^{(k)}$	Defferrard et al. (2017)
Spatial	WL-GNN	$x_i' = W_1 x_i + W_2 \sum_{j\in N(i)} e_{j,i} \cdot x_j$	Morris et al. (2021)
	GraphSAGE	$x_i' = W_1 x_i + W_2 \cdot \text{mean}_{j\in N(i)} x_j$	Hamilton et al. (2018)

algorithm (Goldberger et al., 2004), where each node has a predefined number of neighbors (edges). On the other hand, the second approach aims to model each sample as a graph, allowing more feature comprehension within each sample. In other words, instead of representing each image by a node with a feature vector of size 128, to build a single graph for each sample, the selected output shape of the adaptive pooling layer is [6, 6, 6, 32]. These graphs can have 6 × 6 × 6 nodes, each with feature vector of size 32. Similar to the first approach, edges are constructed between nodes using KNN algorithms based on the relative distance between nodes in 32-dimensional space. The size of the feature vectors and the number of nearest neighbor nodes are hyperparameters that can affect the overall model performance. The influence of these hyperparameters is discussed in the sensitivity analysis subsection of the results.

For each graph $G = \{V, E\}$, where is the set of nodes, and E is the set of edges, a feature matrix X is created. In the node-level task, G represents a graph of the entire dataset, whereas X contains node features of shape $[N, N_f]$ where N is the number of nodes and N_f is number of features. Edge matrix E is created using a coordinate format representation between neighboring nodes, which is established using the KNN algorithm. When building graphs for the graph-level regression task, each G represents a single rock sample (image); hence, more information was incorporated to describe each sample.

3.4 Graph neural network

GNNs can be classified as spatial or spectral GNNs. Spatial GNNs capitalize on local neighborhood information to aggregate and propagate features across graphs. The features of the direct neighbors are aggregated, typically using a weighted-sum operation. They are particularly effective for solving tasks that require capturing local dependencies and information diffusion, making them suitable candidates for microscale porous media modeling. Numerous spatial GNNs can be implemented for this task, of which the Weisfeiler Leman (WL-GNN) (Morris et al., 2021) and GraphSAGE (Hamilton et al., 2018) were selected for the experiment.

On the other hand, spectral GNNs leverage the spectral properties of the graph Laplacian matrix to process graph-structured data. This approach is based on graph theory and utilizes the eigenvalues and eigenvectors of the graph Laplacian matrix to transform graphs into a spectral domain. Convolution can be performed in the spectral domain by applying element-wise multiplication of the transformed data and spectral filters. This type of GNNs can capture global structural information and propagate it through a graph by applying a series of spectral convolutions. This renders spectral GNNs suitable for tasks that require global structural patterns. For the spectral layers of choice, graph convolutional network (GCN) layers (Kipf and Welling, 2017) and Chebyshev network (ChebNet) (Defferrard et al., 2017) were selected.

Table 3 summarizes of the graph convolutional layers used in this study, along with the equations used to update the node embeddings. As demonstrated in the table, spectral GNNs calculate a single updated feature matrix X' that represents all graph nodes of the corresponding layer. Each layer has a single learnable parameter matrix in spectral region, denoted by Θ. In GCN, the normalized adjacency matrix, $\hat{A}$, the normalized degree matrix, $\hat{D}$, and are both obtained by adding self-loops to nodes by adding the identity matrix, I, to the adjacency ma-

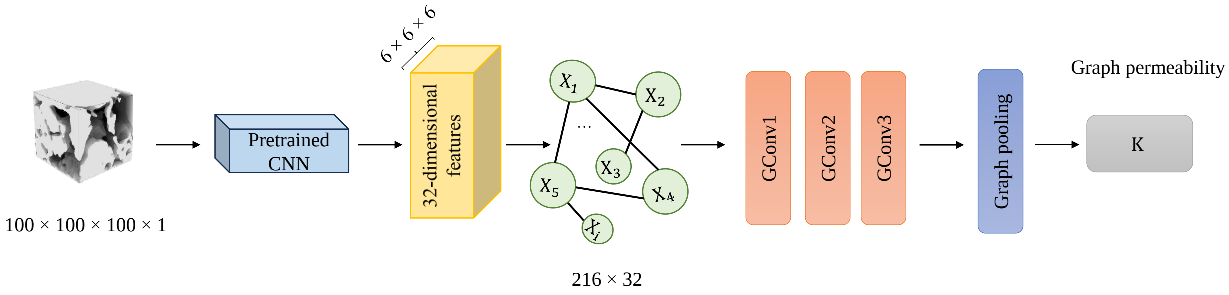

Fig. 5. Illustration of graph-level Pore-GNN architecture. The pre-trained CNN outputs 216 32-dimensional vectors. These vectors were used as the input features to construct a KNN graph for each sample. The three GConv layers update the node embeddings and a graph pooling layer is used to compute a single permeability value for each graph.

Table 4. Metrics for evaluating model regression accuracy.

Metric	Formula
MAE	$\frac{\sum_{j-1}^{N}\left\|(A_j \cdot P_j)\right\|}{N}$
RMSE	$\sqrt{\frac{\sum_{j-1}^{N}(A_j-P_j)^2}{N}}$
MSE	$\frac{\sum_{j-1}^{N}(A_j-P_j)^2}{N}$
R^2	$\frac{\sum_{j-1}^{N}(A_j-P_j)^2}{\sum_{j-1}^{N}(A_j-A)^2}$

trix, A. $Z^{(k)}$ in the ChebNet denotes the kth-order Chebyshev polynomial of the scaled and normalized graph Laplacian. In contrast to spectral convolutions, spatial methods individually compute each node embedding x_i'. In addition, these methods have a learnable node weight matrix W_1 and neighbors weight matrix W_2. Moreover, neighbor node features are denoted as x_j, whereas edge features can be incorporated into spatial GNNs, denoted as $e_{j,i}$. References are included in the table for further details.

3.4.1 Node-level Pore-GNN

The overall architecture of the node-level Pore-GNN is illustrated in Fig. 4. In node-level regression tasks, the pre-trained CNN feature extractor was designed to output a 128-dimensional feature vector for each input sample. A single dataset graph containing N nodes, where N is the number of samples in the dataset. Subsequently, the pre-trained features were used as initial node embeddings, and edges were constructed using the KNN algorithm based on the relative distance between nodes in the 128-dimensional space. Next, the resulting graph was split into 70%, 20%, and 10% for training, validation, and testing, respectively. Three GConv layers were stacked, each followed by ReLU activation, to update the node embeddings, which were the input, hidden, and output layers. The input layer accepts a graph with node features of shape $[N,128]$ and updates the embeddings to the chosen hidden layer size. Finally, the output layer updates the embeddings to a single value per node, which represents normalized permeability.

After the training was completed, the model weights were saved, and predictions can be made for the testing samples. For a new sample (node), the pre-trained feature extractor was first run to obtain a 128-dimensional representation of the sample. Subsequently, a new mini-graph was constructed using the K-nearest training nodes in the dataset graph based on their positions in the 128-dimensional space. Finally, the trained GNN is executed to aggregate the neighbors' information and regress the permeability of the new node. The sensitivity of the proposed architecture to the hidden layer dimensions is discussed in the results section.

3.4.2 Graph-level Pore-GNN

A graph-level architecture is proposed to train the GNNs to perform graph regression tasks, as illustrated in Fig. 5. A single graph is constructed for each sample in the training dataset. In this scenario, the pre-trained CNN feature extractor outputs 216 32-dimensional feature vectors that can be used to construct sample graphs with 216 nodes and 32 node features. Then, the graph convolutional layers, each followed by ReLU, update the node embeddings in a manner similar to the node-level Pore-GNN approach, but with the output of the last graph convolutional layer having a size equal to the hidden layer size instead of one, which is used to predict the node-level permeability. To predict a single value for each graph, the output of the last graph convolutional layer is fed into the graph pooling layer. The pooling layer reduces graph size while retaining its essential features. In sum, product, and mean graph pooling, the node features are added, multiplied, and averaged, respectively, to obtain a single vector representation of the graph. The resulting vector is then input into a fully connected prediction head to predict the desired output. The type of pooling layer is an additional hyperparameter discussed in the sensitivity analysis section of the results.

3.5 Training, validation, and testing

First, the backbone CNN was trained to predict the permeability of the binary images. The original data, comprising 7,000 images, were split into 80% for training and 20% for testing. Validating the baseline CNN was deliberately ignored, as it is beyond the scope of this study. The CNN was trained using the mean absolute error (MAE) as the objective loss

Table 5. Summary of different hyperparameters used for conducting the experiment.

Architecture	Hyper-parameter	Values
Both	GConv layer	GCN, ChebNet, WL-GNN, GraphSAGE
	Hidden layer size	16, 32, 64, 128
	Learning rate	10^{-3}, 10^{-4}, 10^{-5}
Node-Level Pore-GNN	Pre-trained features size	36, 64, 128, 256
Graph-Level Pore-GNN	Graph pooling layer	Sum, Max, Mean

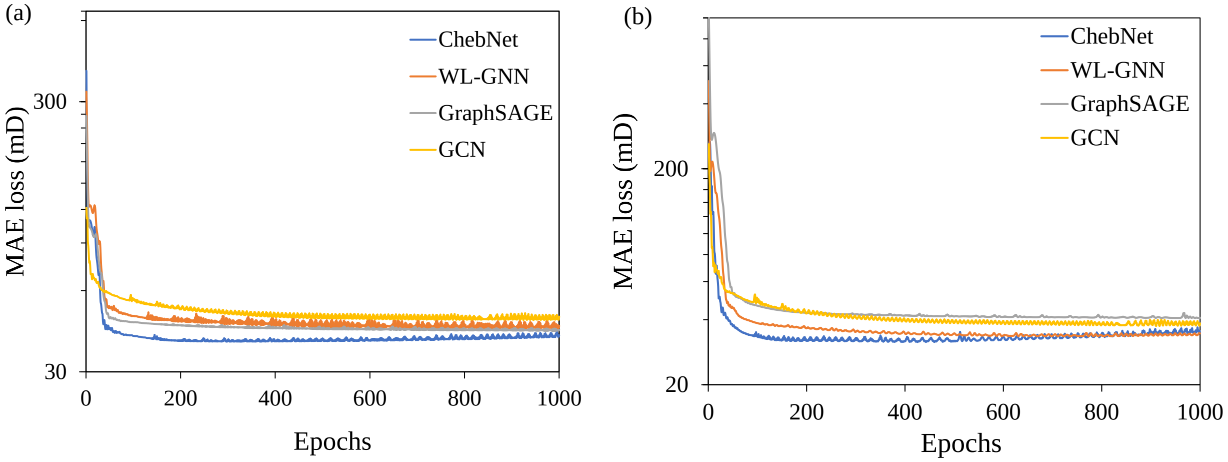

Fig. 6. Training MAE loss of (a) node-level Pore-GNN and (b) graph-level Pore-GNN obtained using different graph convolutional layers.

function. Additionally, the root mean squared error (RMSE), mean squared error (MSE), and coefficient of determination (R^2) were recorded between the model prediction and labels, that is, the ground-truth values of the normalized permeability. A summary of the mathematical formula for computing each error metric is presented in Table 4, where A_j is actual value of the j^{th} sample, P_j is the predicted value, and N is the total number of samples. During each forward propagation, the model predicts a single value of permeability based on randomly initialized weights. In the backpropagation step, the loss is calculated, and the optimizer is utilized to propagate changes (gradients) to the model weights, thereby reducing the objective loss function. The Adam optimizer was selected, which is an adaptive optimization algorithm that adjusts the learning rates for each parameter individually based on the moments of the gradients (Kingma and Ba, 2017). The model was trained for 100 epochs, that is, 100 full iterations of forward and backward propagation over the entire training dataset. At the end of each epoch, model predictions were made on the validation dataset, and validation error metrics were recorded.

After training the baseline CNN, two feature extractors were created, using the outputs of layers 7 and 5 as shown in Table 2, as the output of the feature extractor CNN for the node-level and graph-level regression tasks. The training graphs are then created using the methods described in Section 3.3. In both approaches, the dataset was split into 70%, 20%, and 10% datasets for training, validation, and testing, respectively. MAE was also used as the loss function along with the Adam optimizer. With each forward propagation, each graph convolutional layer updates the node representation by aggregation and transformation operations, which are differentiable, allowing gradients to flow through the model Because the proposed architecture significantly reduces memory and computational requirements, the models can be trained efficiently using a single graphic processing unit (GPU); thus, the models were trained for 1,000 epochs. However, only the model that achieved the lowest loss on the validation dataset was saved. The experiment was set such that the impact of several hyper-parameters can be investigated, as summarized in Table 5, such as the type of graph convolutional layers, the network hidden dimensions, the number of nearest neighbors used to construct the training graphs, and the size of the pre-trained feature vectors in the node-level Pore-GNN. For the graph-level Pore-GNN, the model was trained using various graph pooling layers, and the findings are summarized in the sensitivity analysis of the results section.

To perform CNN and GNN operations computationally, PyTorch (Paszke et al., 2019) and PyTorch Geometric (Fey and Lenssen, 2019) open-source Python libraries were utilized, which facilitate tensor and graph operations, respectively. Training was performed using a single 24 GB GeForce RTX

Table 6. Summary of training, validation, and testing results.

Architecture	GConv layer	MAE (mD)			R^2		
		Training	Validation	Testing	Training	Validation	Testing
Baseline-CNN	CNN	44.3	48.6	50.5	0.8573	0.8442	0.8420
LNode-Level Pore-GNN	GCN	42.6	51.7	50.9	0.8692	0.8175	0.8523
	ChebNet	33.4	46.4	47.2	0.9092	0.8727	0.9024
	WL-GNN	39.1	48.7	46.0	0.8959	0.8612	0.8875
	GraphSage	40.5	43.9	44.1	0.9199	0.8748	0.8969
Graph-Level Pore-GNN	GCN	40.2	48.5	50.8	0.9038	0.8971	0.8780
	ChebNet	30.2	37.7	40.2	0.9355	0.9212	0.9195
	WL-GNN	35.9	43.3	42.0	0.9140	0.9012	0.8910
	GraphSage	41.5	45.1	46.5	0.9259	0.9104	0.8964

3,090 GPU; however, the developed models can be trained using the central processing unit (CPU) within a reasonable timeframe, compared to training the baseline CNN on the CPU. In the following section, the results of the model performance during training and validation of the proposed architecture, results of testing the model on unseen samples, sensitivity analysis of the impact of hyperparameters, and a discussion of the required memory and computational resources are discussed.

4. Results

4.1 Training and validation performance

The models were trained using the MAE as the objective loss function. Fig. 6 shows the training and validation performances of the node-level and graph-level Pore-GNN networks after every epoch. These results represent the best performance models built with the optimal hyperparameters, as discussed in Section 4.3. Overall, both architectures achieved an MAE of less than 50 mD on the training and validation datasets by using various graph convolutional layers. As shown in the figure, graph-level Pore-GNNs built with ChebNet layer managed to reduce the training loss to 30.2 and 37.7 mD for training and validation dataset, respectively. A summary of the resulting MAE losses at the end of the training and R^2 for all networks is presented in Table 6. Implementing the Pore-GNN architectures improved the computed R^2, compared to the baseline model, within ranges from 2%-7.9% for the node-level Pore-GNN and 5.6%-10.5% for the graph-level Pore-GNN architecture. These improvements in MAE and R^2 are indicative of higher accuracies compared to the baseline CNN and, hence, improved predictive performance. The stability of MAE throughout the training phase, as shown in Fig. 6, indicates reduced uncertainty, even when testing on the validation dataset that was used during training to compute the errors, but not used to train the model, that is, weight computations.

4.2 Models testing

To validate the trained models, the predicted normalized permeability values were compared with the actual values and the R^2 coefficient was used to evaluate the fit of the data. Figs. 7 and 8 illustrate the best achieved model accuracy based on the training, validation, and testing datasets for the node-level and graph-level regression Pore-GNN architectures, respectively. As shown in these figures, all graph convolutional layers regressed the permeability of porous media, with varying degrees of accuracy. Generally, the graph-level regression Pore-GNN performs better than the node-level architecture as well as the baseline CNN. This was expected because the network was fed with more information about each subsample in the graph-level architecture, i.e., assisting the model to capture more complex topological relationships within pore space regions. This comes at an additional memory cost, as opposed to the node-level task, where the entire dataset can be loaded and trained using a single GPU. Further discussion regarding the memory and computational requirements is included in Section 4.4.

Testing of the best model was extended to predict the permeabilities of unseen samples, i.e., rock samples that were not encountered during the training of the models. For this purpose, two sandstone samples were utilized, namely Bandera Gray (BG) and Buff Berea (BB), which were published with the rest of the samples, but not used for training. Additionally, a carbonate sample, Indiana limestone (IL), was imaged using the Tyree X-ray facility at the University of New South Wales, Sydney. A summary of the unseen testing rocks is presented in Table 7. As shown in the table, the permeabilities of the full testing core samples are spread over the entire range of permeabilities of the training samples, as shown in Table 1, where BG resembles the lower end (9 mD) and BB resembles the upper end (275 mD), while IL falls somewhere in between them. Additionally, the IL sample was included to test the robustness of different models when encountering varying and more complex microstructures, namely carbonates. Moreover, because the IL sample was imaged separately, it had a different

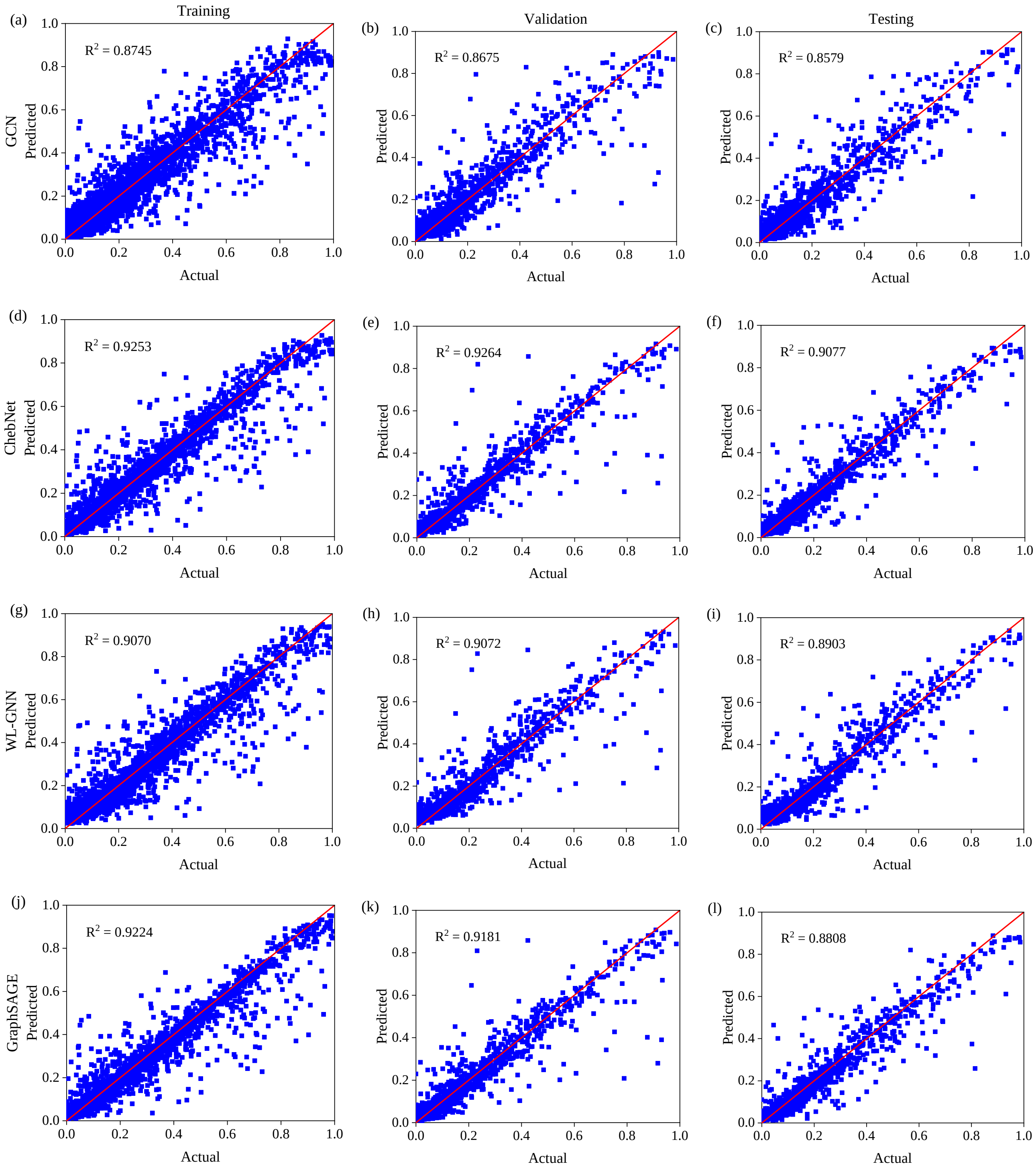

Fig. 7. Illustrations of the accuracies node-level Pore-GNN architectures based on (a), (d), (g), (j) training, (b), (e), (h), (k) validation, and (c), (f), (i), (l) testing. The x-axes show the actual normalized permeabilities obtained from LBM simulation, while the y-axes represent the normalized permeability predicted using different graph convolutional layers.

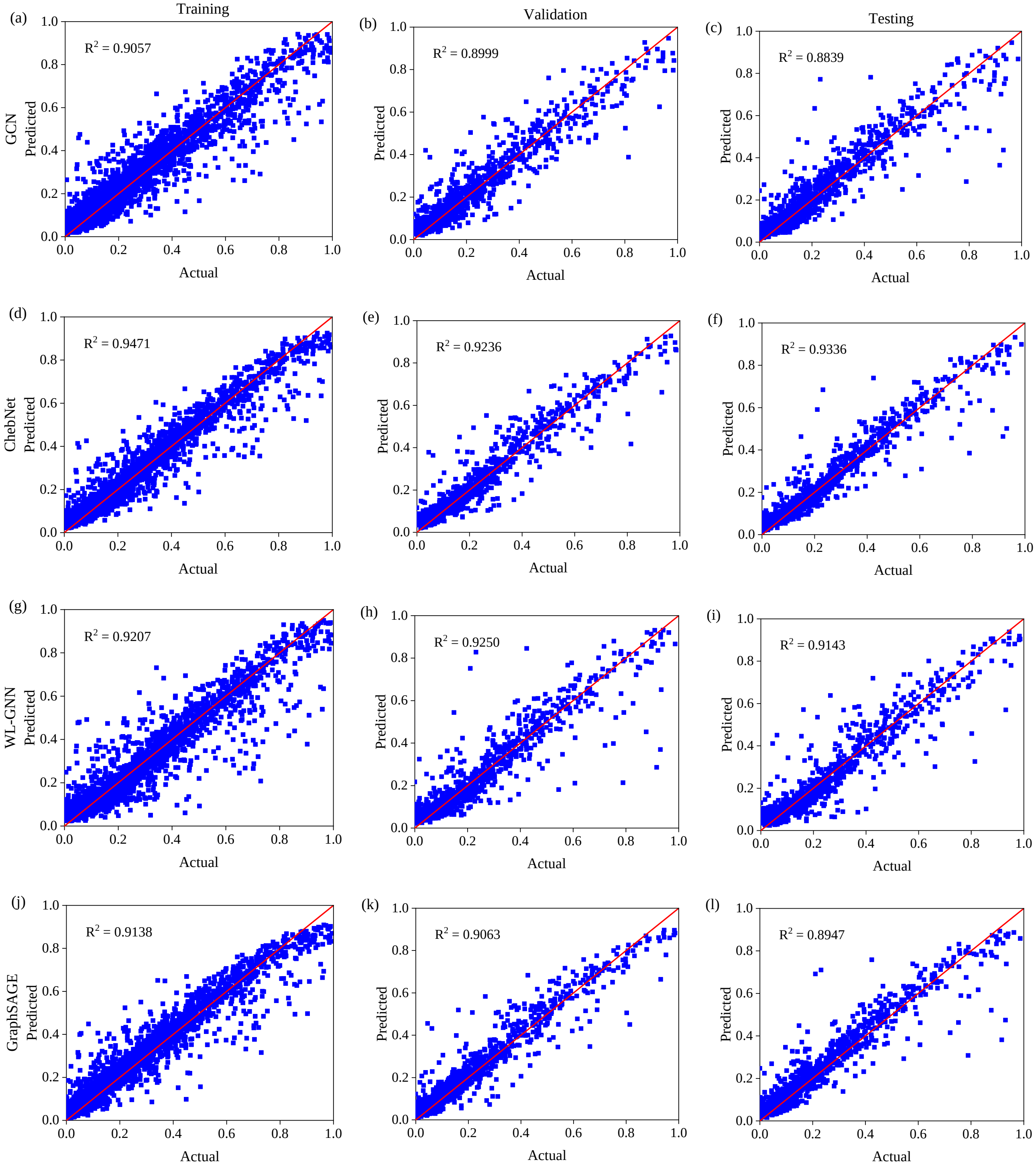

Fig. 8. Illustrations of accuracies of graph-level Pore-GNN architectures based on (a), (d), (g), (j) training, (b), (e), (h), (k) validation, and (c), (f), (i), (l) testing. The x-axes show the actual normalized permeabilities obtained from LBM simulation, while the y-axes represent the normalized permeability predicted using different graph convolutional layers.

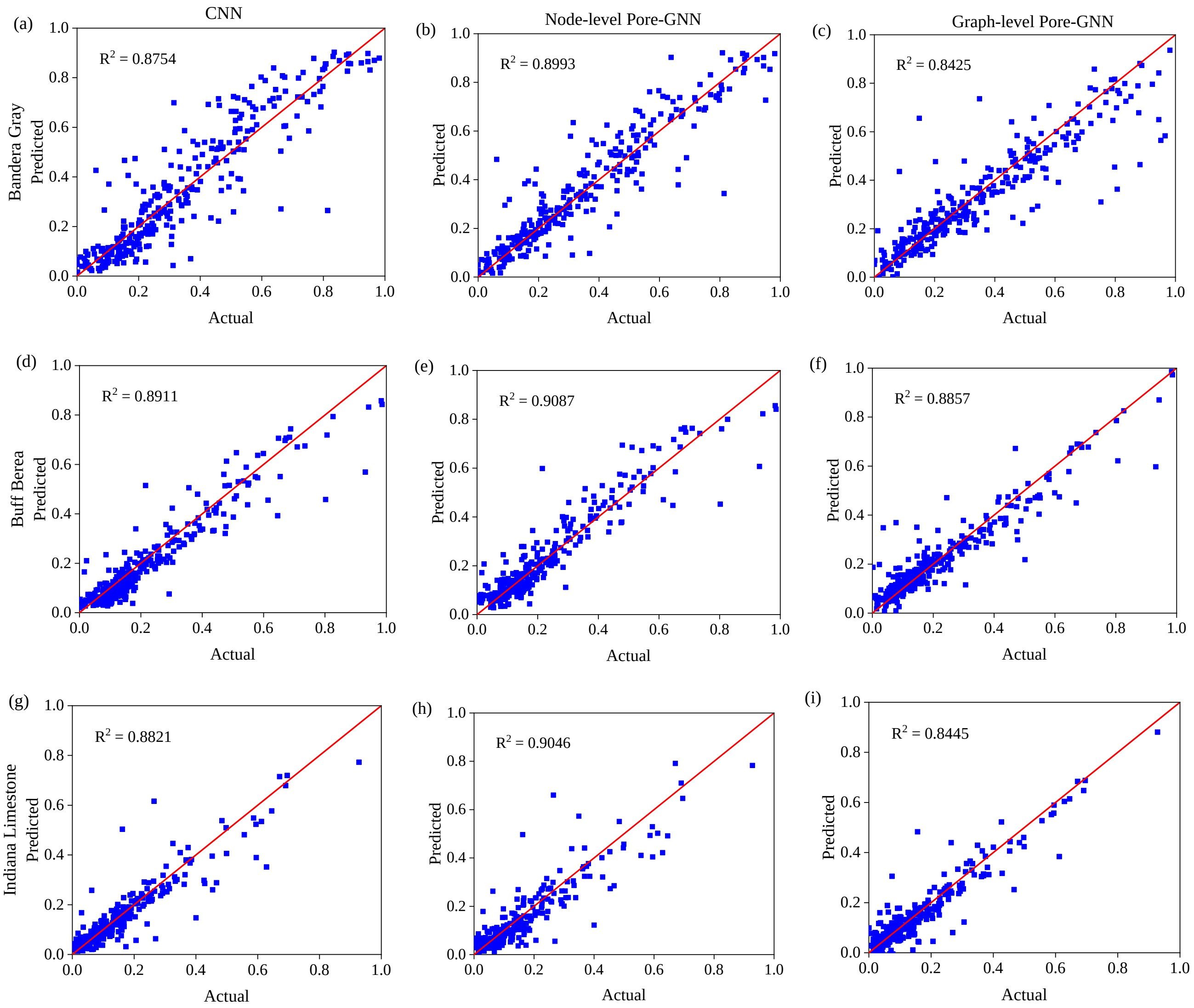

Fig. 9. Testing results on unseen Bandera Gray (a), (b), (c) and Buff Berea (d), (e), (f) sandstones samples in addition to Indiana Limestone (g), (h), (i) samples. These results are obtained from (a), (d),(g) Baseline CNN, (b), (e), (h) Node-level Pore-GNN, and (c), (f), (i) Graph-level Pore-GNN.

Table 7. Summary of the properties of the two unseen sandstone and carbonate rock samples used for testing.

Rock	Label	Porosity (%)	Tortuosity	SSA (1/µm)	APS (µm^3)	K_{exp}(mD)
Bandera Gray	BG	18.10	1.95	0.0375	14,838	9
Buff Berea	BB	24.02	1.55	0.0621	32,159	275
Indiana Limestone	IL	18.00	1.77	0.0589	28,500	113

resolution of 3.4 µm^3/voxel, meaning that extracting cubes of size 100^3 voxels would capture more than the length scale used for training, measured in µm. Therefore, cubes of size 66^3 voxels were extracted from the IL sample and rescaled (coarsened) to 100^3. Thus, the input length scale for testing is ensured to match the length scale used for training.

After processing the sandstone and carbonate images, 300 random sub-samples of size 100^3 voxels were extracted, and then, CNN-based feature extractors were applied to compute the pre-trained features. Next, the models were tested, and predictions were made in a manner akin to that described in Section 3.4, for both node-level and graph-level Pore-GNN. The results of the best-performing models for node-level and graph-level Pore-GNN architectures on the unseen samples are shown in Fig. 9. The best model was able to predict the permeabilities BG sandstone samples with R^2 of 0.87 and 0.89 for the node-level and graph-level Pore-GNN, respectively, outperforming the baseline CNN with 4%-6.7%

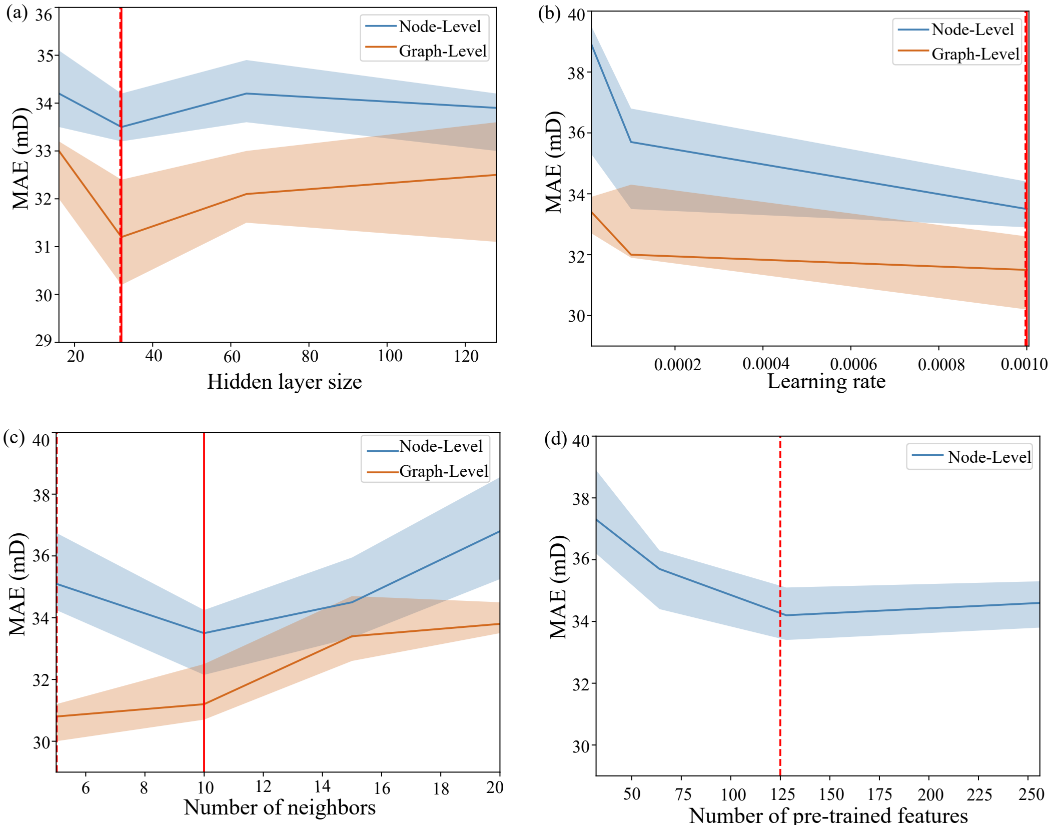

Fig. 10. Sensitivity analysis results showing the effect of (a) hidden layers size, (b) learning rate, (c) number of nearest neighbors, and (d) size of the pre-trained feature vectors in node-level Pore-GNN. The red solid and dashed lines represent the hyper-parameter values that yield the lowest MAE for the node-level and graph-level Pore-GNN, respectively.

improvement. Similarly, the model predicts BB sandstone samples permeabilities with R^2 of 0.89 and 0.91 for the node-level and graph-level Pore-GNN, respectively. On the other hand, the IL carbonate samples permeability were predicted with R^2 of 0.88 and 0.91 achieved by node-level and graph-level Pore-GNN, respectively, using ChebNet as the graph convolutional layer. In addition, the performance of the models was compared with that of the baseline CNN for IL permeability prediction. The proposed models were shown to improve R^2 by 7% for predicting carbonate sample permeability. In summary, these results highlight the capabilities of GNNs to further comprehend pre-trained features and map them to the corresponding permeability labels, even when encountered by unseen samples with different structures from those existing in the training dataset.

4.3 Sensitivity analysis on the effect of hyper-parameters

The robustness of the proposed models towards the hyper-parameters tested was evaluated by performing a sensitivity analysis on both node-level and graph-level regression tasks. In this analysis, model performance was tested using different hidden layer sizes, numbers of nearest neighbors to construct the input graphs, and learning rates, as shown in Fig. 10. Optimal hyperparameters are selected based on the combinations that yield the lowest MAE, indicated by the solid and dashed red lines in the figure, for node-level and graph-level approaches, respectively. The results in the figure show the best model performance, with Pore-GNN with ChebNet layers, under variable hyperparameters. In addition, for node-level tasks, the sizes of the pre-trained input features were tested. The best models were achieved with 32 as the hidden layers size, 0.001 as the learning rate, and five nearest neighbors. In addition, the feature vector size that yielded the lowest MAE is 128. Moreover, the influence of various graph pooling functions was investigated. Overall, as shown in the figure, the developed frameworks are not highly sensitive to the studied hyperparameters, i.e., the highest standard deviation caused by varying these parameters is 7.3 mD. With respect to the selected pooling function, implementing the summation pooling operation on top of graph convolutional layers yielded the lowest MAE of 30.5 mD, while max pooling operation yielded the lowest MAE of 38.7 mD.

4.4 Memory and computational requirements

Owing to the large size of the training data (7,000 3D images), a custom data loader was implemented to pipeline the data into the models during the training phase. These loaders allow data to be loaded to the memory on a batch-by-batch basis. The maximum batch size that was able to fit into a 24

Table 8. A summary of the training and interference time of the trained models.

Device	Model	Training time (minutes)	Inference time (ms/sample)
GPU	Baseline-CNN	153	48
	Node-level Pore-GNN	13	39
	Graph-level Pore-GNN	17	42
CPU	Baseline-CNN	501	242
	Node-level Pore-GNN	53	170
	Graph-level Pore-GNN	65	162

GB GPU was 16 samples per batch for the baseline CNN. On the other hand, the single training graph for the node-level Pore-GNN could fit into the memory all at once without mini-batching. However, training graph-level Pore-GNN requires batch training. Similar to training the baseline CNN, the maximum allowable batch size, using 128 GB of memory, was 16 samples per batch. Further reduction of the batch size to 8 was attempted; however, the model started overfitting, increasing MAE after approximately 300 epochs of training. Table 8 presents the training and inference times for each architecture are shown. The training time is computed based on the total time required to train the model over the entire training epoch, whereas the inference time refers to the average time required to the model on the entire dataset. In contrast, the reported inference times do not include the feature extraction and graph construction steps; instead, they represent the duration required to generate predictions from pre-processed inputs. Regardless, the average end-to-end inference times, including feature extraction and graph construction, were 140 and 96 ms for the node-level and graph-level Pore-GNN architectures, respectively. As shown in the table, the proposed architecture significantly reduced the training time; however, it required high memory availability compared to the CNN, especially for the graph-level regression task. Once the models were trained, both Pore-GNN architectures could efficiently predict the permeabilities of the new samples, enabling accelerated porous media characterization and reduced computational resources. Notably, improvements of 407 and 527 times were achieved for the node-level and graph-level Pore-GNN, respectively, compared with the numerical LBM solver, which required an average of 72.8 seconds to perform the simulations. These improvements are computed based on the total prediction time, including the feature extraction, graph construction time and inference time for each method.

5. Conclusions

This paper presented Pore-GNN framework, a deep learning approach for characterizing the properties of porous media from 3D micro-CT images. The proposed approach capitalizes on CNN as the backbone feature extractor. Then, two strategies were examined for embedding the extracted features into the training graphs: (1) sampling as a node for node-level regression tasks, and (2) sampling as a graph for graph-level regression tasks. The results highlight the capabilities of GNNs to improve the predictive performance of the baseline CNN by 1.2%-7.2% for the node-level Pore-GNN and 4.3%-9.2% for the graph-level Pore-GNN architecture. Moreover, the developed architecture either matched or outperformed the baseline CNN in predicting the permeability of unseen samples of varying structures, namely BB and BG sandstones in addition to IL carbonate sample.

Although the proposed architectures enhanced the predictive capabilities and outperformed the baseline CNN in terms of efficiency and accuracy, they are still bound by the same main limitation of CNNs, which is the inability to process raw inputs of arbitrary dimensionality. This is because the proposed models are hybrid models that depend on CNN-based feature extractors. Additionally, increasing the dimensions of the feature maps and number of features used to build the training graphs can cause a significant increase in computational requirements to a point where a single GPU might not be able to load and process the entire dataset, which might necessitate mini-batch training techniques. Future work should investigate the feasibility of techniques such as adaptive feature extraction or multi-resolution architectures to accommodate varying input dimensions. In terms of computational requirements, optimization through parallelization strategies, distributed training methods, or hardware acceleration can be considered. Moreover, it would be interesting to exploit physics-based graph learning by embedding pre-computed physical properties of the training samples as node or graph features for training GNNs.

Graph-structured data representations are routinely used for pore-scale characterization. For future work, it is recommended that three main areas of interest be investigated: (1) PNMs, (2) modelling of surface and interfacial porous media properties, and (3) mesh-based numerical fluid flow simulations. GNN-based approaches can provide a unique road map for processing graph-structured data by capturing the complex dependencies between the interconnected graph nodes. Furthermore, a comprehensive comparative analysis of unstructured deep learning techniques can provide valuable insights into their strengths and weaknesses within the context of porous media characterization problems.

Acknowledgements

The original dataset used for training the model was made available on Digital Rocks Portal (Prodanovic et al., 2015) and published by Lucas-Oliveira et al. (2020) and Neumann et al. (2021). The images of unseen IL carbonate sample used for testing were obtained using the Tyree X-ray Lab at the University of New South Wales, Sydney.

Conflict of interest

The authors declare no competing interest.

Open Access This article is distributed under the terms and conditions of the Creative Commons Attribution (CC BY-NC-ND) license, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

References

Akai, T., Blunt, M. J., Bijeljic, B. Pore-scale numerical simulation of low salinity water flooding using the lattice boltzmann method. Journal of Colloid and Interface Science, 2020, 566: 444-453.

Al-Hashimi, O., Hashim, K., Loffill, E., et al. A comprehensive review for groundwater contamination and remediation: Occurrence, migration and adsorption modelling. Molecules (Basel, Switzerland), 2021, 26(19): 5913.

Alqahtani, N., Alzubaidi, F., Armstrong, R. T., et al. Machine learning for predicting properties of porous media from 2D X-ray images. Journal of Petroleum Science and Engineering, 2020, 184: 106514.

Alqahtani, N., Armstrong, R. T., Mostaghimi, P. Deep learning convolutional neural networks to predict porous media properties. Paper SPE 191906 Presented at the SPE Asia Pacific Oil and Gas Conference and Exhibition, Australia, 23-25 October, 2018.

Alqahtani, N. J., Chung, T., Wang, Y., et al. Flow-based characterization of digital rock images using deep learning. SPE Journal, 2021, 26(4): 1800-1811.

Alsobhani, A., ALabboodi, H. M. A., Mahdi, H. Speech recognition using convolution deep neural networks. Journal of Physics: Conference Series, 2021, 1973: 012166.

Ashual, O., Wolf, L. Specifying object attributes and relations in interactive scene generation. arXiv, 2019, 1909.05379.

Bhatnagar, P. L., Gross, E. P., Krook, M. A model for collision processes in gases. I. small amplitude processes in charged and neutral one-component systems. Physical Review, 1954, 94: 511-525.

Brantson, E. T., Ju, B., Appau, P. O., et al. Development of hybrid low salinity water polymer flooding numerical reservoir simulator and smart proxy model for chemical enhanced oil recovery (CEOR). Journal of Petroleum Science and Engineering, 2020, 187: 106751.

Cai, C., Vlassis, N., Magee, L., et al. Equivariant geometric learning for digital rock physics: Estimating formation factor and effective permeability tensors from morse graph. International Journal for Multiscale Computational Engineering, 2023, 21: 1-24.

Cheng, C., Herrmann, J., Wagner, B., et al. Long-term evolution of fracture permeability in slate: An experimental study with implications for enhanced geothermal systems (EGS). Geosciences, 2021, 11(11): 443.

Defferrard, M., Bresson, X., Vandergheynst, P. Convolutional neural networks on graphs with fast localized spectral filtering. arXiv, 2017,1606.09375.

Deng, J., Dong, W., Socher, R., et al. ImageNet: A large-scale hierarchical image database. Paper Presented at 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20-25 June, 2009.

d'Humires, D. Multiple-relaxation-time lattice boltzmann models in three dimensions. Philosophical Transactions of the Royal Society of London, 2002, 360(1792): 437-451.

Ding, K., Wang, S., Luo, Y. Supervised biological network alignment with graph neural networks. Bioinformatics, 2023, 39: i465-i474.

Ding, H., Wang, Y., Shapoval, A., et al. Macro- and microscopic studies of "smart water" flooding in carbonate rocks: An image-based wettability examination. Energy & Fuels, 2019, 33(8): 6961-6970.

Ershadnia, R., Wallace, C. D., Soltanian, M. R. CO_2 geological sequestration in heterogeneous binary media: Effects of geological and operational conditions. Advances in Geo-Energy Research, 2020, 4(4): 392-405.

Essaid, H. I., Bekins, B. A., Cozzarelli, I. M. Organic contaminant transport and fate in the subsurface: Evolution of knowledge and understanding. Water Resources Research, 2015, 51: 4861-4902.

Fey, M., Lenssen, J. E. Fast graph representation learning with pyTorch geometric. arXiv, 2019, 1903.02428.

Gärttner, S., Alpak, F. O., Meier, A., et al. Estimating permeability of 3D micro-CT images by physics-informed CNNs based on DNS. Computational Geosciences, 2023, 27: 245-262.

Gerke, K. M., Sizonenko, T. O., Karsanina, M. V., et al. Improving watershed-based pore-network extraction method using maximum inscribed ball pore-body positioning. Advances in Water Resources, 2020, 140: 103576.

Gilmer, J., Schoenholz, S. S., Riley, P. F., et al. Neural message passing for quantum chemistry. arXiv, 2017, 1704.01212.

Goldberger, J., Hinton, G. E., Roweis, S., et al. Neighbourhood components analysis. Paper Presented at Advances in Neural Information Processing Systems 17, Columbia, Canada, December, 2004.

Gostick, J., Khan, Z., Tranter, T., et al. PoreSpy: A python toolkit for quantitative analysis of porous media images. Journal of Open Source Software, 2019, 4: 1296.

Graczyk, K. M., Matyka, M. Predicting porosity, permeability, and tortuosity of porous media from images by deep learning. Scientific Reports, 2020, 10: 21488.

Guo, Y., Zhang, L., Zhu, G., et al. A pore-scale investigation of residual oil distributions and enhanced oil recovery methods. Energies, 2019, 12(19): 3732.

Hamilton, W. L., Ying, R., Leskovec, J. Inductive representation learning on large graphs. arXiv, 2018, 1706.02216.

Hussain, S. T., Rahman, S. S., Azim, R. A., et al. Multiphase fluid flow through fractured porous media supported by innovative laboratory and numerical methods for estimating relative permeability. Energy & Fuels, 2021, 35(21): 17372-17388.

Hussain, S. T., Regenauer-Lieb, K., Zhuravljov, A., et al. Asymptotic hydrodynamic homogenization and thermodynamic bounds for upscaling multiphase flow in porous media. Advances in Geo-Energy Research, 2023, 9(1): 38-53.

Iglauer, S., Pentland, C. H., Busch, A. CO_2 wettability of seal and reservoir rocks and the implications for carbon geo-sequestration. Water Resources Research, 2015, 51(1): 729-774.

Ijeje, J. J., Gan, Q., Cai, J. Influence of permeability anisotropy on heat transfer and permeability evolution in geothermal reservoir. Advances in Geo-Energy Research, 2019, 3(1): 43-51.

Jiang, J., Guo, B. Graph convolutional networks for simulating multi-phase flow and transport in porous media. arXiv, 2023, 2307.04449.

Jolliffe, I. T., Cadima, J. Principal component analysis: A review and recent developments. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 2016, 374(2065): 20150202.

Kashefi, A., Mukerji, T. Point-cloud deep learning of porous media for permeability prediction. Physics of Fluids, 2021, 33(9): 097109.

Kingma, D. P., Ba, J. Adam: A method for stochastic optimization. arXiv, 2017, 1412.6980.

Kipf, T. N., Welling, M. Semi-supervised classification with graph convolutional networks. arXiv, 2017, 1609.02907.

Li, X., Li, B., Liu, F., et al. Advances in the application of deep learning methods to digital rock technology. Advances in Geo-Energy Research, 2023, 8(1): 5-18.

Li, B., Nie, X., Cai, J., et al. U-Net model for multi-component digital rock modeling of shales based on CT and QEMSCAN images. Journal of Petroleum Science and Engineering, 2022, 216: 110734.

Liang, J., Deng, Y., Zeng, D. A deep neural network combined CNN and GCN for remote sensing scene classification. Earth Observations and Remote Sensing, 2020, 13: 4325-4338.

Liu, Q., Kampffmeyer, M., Jenssen, R., et al. SCG-Net: Self-constructing graph neural networks for semantic segmentation. arXiv, 2021, 2009.01599.

Lucas-Oliveira, E., Araujo-Ferreira, A. G., Trevizan, W. A., et al. Sandstone surface relaxivity determined by NMR T_2 distribution and digital rock simulation for permeability evaluation. Journal of Petroleum Science and Engineering, 2020, 193: 107400.

Mohammadi, M., Riazi, M. Applicable investigation of SPH in characterization of fluid flow in uniform and non-uniform periodic porous media. Sustainability, 2022, 14(21): 14320.

Morris, C., Ritzert, M., Fey, M., et al. Weisfeiler and leman go neural: Higher-order graph neural networks. arXiv, 2021, 1810.02244.

Neumann, R. F., Barsi-Andreeta, M., Lucas-Oliveira, E., et al. High accuracy capillary network representation in digital rock reveals permeability scaling functions. Scientific Reports, 2021, 11(1): 11370.

Paszke, A., Gross, S., Massa, F., et al. PyTorch: An imperative style, high-performance deep learning library. arXiv, 2019, 1912.01703.

Pearson, K. LIII. On lines and planes of closest fit to systems of points in space. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, 1901, 2(11): 559-572.

Perrin, J. C., Krause, M., Kuo, C. W., et al. Core-scale experimental study of relative permeability properties of CO_2 and brine in reservoir rocks. Energy Procedia, 2009, 1(1): 3515-3522.

Pfaff, T., Fortunato, M., Sanchez-Gonzalez, A., et al. Learning mesh-based simulation with graph networks. arXiv, 2021, 2010.03409.

Prodanovic, M., Esteva, M., Hanlon, M. Digital rocks portal: A sustainable platform for imaged dataset sharing, translation and automated analysis. AGU Fall Meeting Abstracts, 2015, 2015: MR43A-02.

Qi, C. R., Su, H., Mo, K., et al. PointNet: Deep learning on point sets for 3D classification and segmentation. arXiv, 2017, 1612.00593.

Rabbani, A., Babaei, M., Shams, R., et al. DeePore: A deep learning workflow for rapid and comprehensive characterization of porous materials. Advances in Water Resources, 2020, 146: 103787.

Redmon, J., Divvala, S., Girshick, R., et al. You only look once: Unified, real-time object detection. arXiv, 2016, 1506.02640.

Ronneberger, O., Fischer, P., Brox, T. U-Net: Convolutional networks for biomedical image segmentation. arXiv, 2015, 1505.04597.

Ryazanov, A. V., van Dijke, M. I., Sorbie, K. S. Pore-network prediction of residual oil saturation based on oil layer drainage in mixed-wet systems. Paper SPE 129919 Presented at the SPE Improved Oil Recovery Symposium, Tulsa Oklahoma, 24-28 April, 2010.

Santos, J. E., Gigliotti, A., Bihani, A., et al. MPLBM-UT: Multiphase LBM library for permeable media analysis. SoftwareX, 2022, 18: 101097.

Santos, J. E., Xu, D., Jo, H., et al. PoreFlow-Net: A 3D convolutional neural network to predict fluid flow through porous media. Advances in Water Resources, 2020, 138: 103539.

Shapoval, A., Alzahrani, M., Xue, W., et al. Oil-water interactions in porous media during fluid displacement: Effect of potential determining ions (PDI) on the formation of in-situ emulsions and oil recovery. Journal of Petroleum Science and Engineering, 2022, 210: 110079.

Shapoval, A., Zhuravljov, A., Lanetc, Z., et al. Pore-scale evaluation of physicochemical interactions by engineered water injections. Transport in Porous Media, 2023, 148: 605-625.

Taigman, Y., Yang, M., Ranzato, M., et al. DeepFace: Closing the gap to human-level performance in face verification. Paper Presented at 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23-28 June, 2014.

Tran, D., Bourdev, L., Fergus, R., et al. Learning spatiotemporal features with 3D convolutional networks. Paper Presented at 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7-13 December, 2015.

Wang, W., Gang, J. Application of convolutional neural network in natural language processing. Paper Presented at 2018 International Conference on Information Systems and Computer Aided Education (ICISCAE), Changchun, China, 6-8 July, 2018.

Wang, Z., Li, H., Lan, X., et al. Formation damage mechanism of a sandstone reservoir based on micro-computed tomography. Advances in Geo-Energy Research, 2021, 5(1): 25-38.

Wang, Y., Shabaninejad, M., Armstrong, R. T., et al. Physical accuracy of deep neural networks for 2D and 3D multi-mineral segmentation of rock micro-CT images. arXiv, 2020, 2002.05322.

Wang, Y., Song, R., Liu, J., et al. Pore scale investigation on scaling-up micro-macro capillary number and wettability on trapping and mobilization of residual fluid. Journal of Contaminant Hydrology, 2019, 225: 103499.

Wu, C., Pfrommer, J., Beyerer, J., et al. Object detection in 3D point clouds via local correlation-aware point embedding. Paper Presented at the 2020 Joint 9th International Conference on Informatics, Electronics & Vision (ICIEV) and 2020 4th International Conference on Imaging, Vision & Pattern Recognition (icIVPR), Kitakyushu, Japan, 26-29 August, 2020.

Ying, R., He, R., Chen, K., et al. Graph convolutional neural networks for web-scale recommender systems. Paper Presented at the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, United Kingdom, 19-23 August, 2018.

Yun, W., Liu, Y., Kovscek, A. R. Deep learning for automated characterization of pore-scale wettability. Advances in Water Resources, 2020, 144: 103708.