Yinquan Meng[1], Jianguo Jiang[1], Jichun Wu[1], and Dong Wang[1]

[1]Key Laboratory of Surficial Geochemistry of Ministry of Education, School of Earth Sciences and Engineering, Nanjing University, Nanjing, China

Corresponding author: Jianguo Jiang (jianguo.jiang@nju.edu.cn)

Key Points:

- We introduce a hybrid neural network combining convolution with self-attention mechanism for permeability prediction with great accuracy
- The accuracy is the result of novel network architecture and incorporating physical parameters into digital images
- The generalization of the model has been reliably validated on previously unseen samples with the transfer learning method

Abstract

The direct acquisition of the permeability of porous media by digital images helps to enhance our understanding of and facilitate research into the problem of subsurface flow. A complex pore space makes the numerical simulation methods used to calculate the permeability quite time-consuming. Deep learning models represented by three-dimensional convolutional neural networks (3D CNNs), as a promising approach to improving efficiency, have made significant advances concerning predicting the permeability of porous media. However, 3D CNNs only have the ability to represent the local information of 3D images, and they cannot consider the spatial correlation between 2D slices, a significant factor in the reconstruction of porous media. This study combines a 2D CNN and a self-attention mechanism to propose a novel CNN-Transformer hybrid neural network that can make full use of the 2D slice sequences of porous media to accurately predict their permeability. In addition, we added physical information to the slice sequences and built a PhyCNN-Transformer model to reflect the impact of physical properties on permeability prediction. In terms of dataset preparation, we used the publicly available DeePore porous media dataset with the labeled permeability calculated by pore network modelling (PNM). We compared the two transformer-based models with a 3D CNN in terms of parameter number, training efficiency, prediction performance, and generalization, and the results showed significant improvement. Combined with the transfer learning method, we demonstrate the superior generalization ability of the transformer-based models to unfamiliar samples with small sample sizes.

1 Introduction

Determining the permeability of porous media such as soils and rocks is crucial for the study of natural and industrial processes. The research on subsurface flow, oil exploitation, nuclear waste leakage treatment, carbon dioxide sequestration, and other issues is inseparable from the key parameter of permeability (Djabelkhir et al., 2017; Saljooghi & Hezarkhani, 2015; Tatar et al., 2015;

Tsang et al., 2015). As a significant parameter required to measure the difficulty of fluid passing through a porous medium, permeability plays a basic control role in establishing mathematical models of porous media seepage and pollutant migration. Accurate acquisition of the degree of permeability helps us to comprehensively understand the detailed process of material transport in geological porous media, reduce the application uncertainty, and improve the application efficiency in order to more precisely evaluate and predict the dynamic characteristics of geological bodies (Bultreys et al., 2016). The traditional methods of permeability measurement require a pressure test on a core sample, and then the permeability is calculated according to Darcy's law (San Manley et al., 2020). However, the test method is cumbersome and subject to environmental disturbance, and thus it is not suitable for obtaining the permeability of porous media encountered in large quantities. In recent decades with the development of imaging technologies such as X-ray micro computed tomography (micro-CT), computational fluid dynamics (CFD) simulation of digital rocks has gradually become the main method to determine the permeability of porous media. The Lattice Boltzmann method (LBM) is the most popular approach for permeability characterization directly from pore-space images (Blunt et al., 2013; Elmorsy et al., 2022). LBM solves the Navier-Stokes equations on complex boundaries provided by the 3D pore space, but its computational complexity limits the calculable size of the porous media. Pore Network Modelling (PNM) is another effective method for simulating pore flow and calculating permeability. This method simplifies the geometry of the pore space and preserves the pore structure characteristics necessary for material migration, steps that can greatly improve computational efficiency and can be applied to large-scale geological bodies. However, the accuracy of the calculation results is not as high as that of LBM (Da Wang et al., 2021; Li et al., 2017).

In recent years with the development of machine learning (ML) and deep learning (DL), many researchers have accelerated the numerical simulation process of calculating the permeability of porous media by using neural network models in order to augment the calculation efficiency and accuracy. The permeability of a porous medium is determined by the pore geometric space, and thus a direct mapping between the permeability and digital images of porous media can be established by CNNs. CNNs are widely used in the field of computer vision, and they have many variants that perform well on tasks such as image recognition and semantic segmentation (He et al., 2016; Huang et al, 2017). The permeability and digital images of porous media are input into CNNs, and after training with a mass of samples, the models can accurately predict the permeability with minor errors. Araya-Polo et al. (2020) input digital images of sandstone from different reservoirs and cores into CNNs and predicted the permeability of multiple reservoir thin sections, obtaining an R^2 of 0.7967 on the testing dataset. Tang et al. (2022) proposed to incorporate global physical information of porous media such as porosity and tortuosity in dense blocks of densely connected convolutional networks (DenseNet) that can effectively improve the prediction accuracy. The performance of the models was evaluated

on a small dataset and a dataset with a different sample distribution from that of the training set. Elmorsy et al. (2022) designed a new end-to-end model that encompassed an inception module capable of feature extraction at multiple scales before sequential convolutional layers. The coefficients of determination R^2 of the trained model were 0.95 and 0.93 on the testing dataset and the external dataset with previously unseen samples, respectively. They believed that collecting a wider variety of porous media datasets and building intricate and ingenious neural networks could take this approach to new heights in the coming years.

For the past few years, 2D slices have been considered as the input for 3D porous media reconstruction models. Feng et al. (2020) realized the mapping of 2D slices to 3D structures by a BicycleGAN framework. Zheng et al. (2022) proposed a RockGPT method to reconstruct 3D structures based on a single 2D slice. Zhang et al. (2021) treated slices of porous structures as spatial series and proposed a 3D porous media recurrent neural network (3D-PMRNN) to generate 3D porous media from a corresponding 2D image. They used LSTM to learn the image correlation in spatial series and then predicted future layers along the z direction. The layers were then finally stacked into a 3D structure.

The above successful applications prompted us to consider the significance of the 2D spatial series of the porous medium to the 3D pore structure. Selecting an appropriate sequence model to deal with the correlations between 2D slices may help to predict the permeability of the 3D pore structure, as 3D CNNs do at present. Transformer is a sequence-to-sequence model based on the attention mechanism, meaning the use of limited attention resources to quickly focus on needed information, greatly improving guidance and decision making (Mnih et al., 2014; Vaswani et al., 2017). Compared with recurrent neural networks (RNNs), Transformer without computationally-inefficient recurrent connections can establish a longer connection between elements in the sequence (Bai & Tahmasebi, 2022). In this work, we transform the regression problem of 3D images into the regression problem of sequences of 2D images and propose a CNN-Transformer hybrid neural network. Compared with 3D CNNs, the number of parameters of the model is greatly reduced, and the prediction performance is significantly improved. In addition, we added the corresponding physical parameter matrix to each 2D digital image in the sequences to construct a PhyCNN-Transformer hybrid neural network that makes the model's prediction performance better and more robust.

The remainder of this paper is structured as follows. In Section 2, the principle and hybrid mode of the 2D CNN and Transformer are elaborated in detail. After that, we explain how to include physical parameter information into 2D digital images and describe the structure of hybrid neural networks. Then, we introduce the datasets used in this study. In Section 3 we compare the prediction results of the CNN-Transformer and PhyCNN-Transformer, and then compare the differences between the two models and a 3D CNN model in terms of prediction performance, number of parameters, and training efficiency. We

examine the generalization abilities of the models on previously unseen samples from two kinds of porous media datasets that differ from the training dataset. Section 4 summarizes and concludes the study.

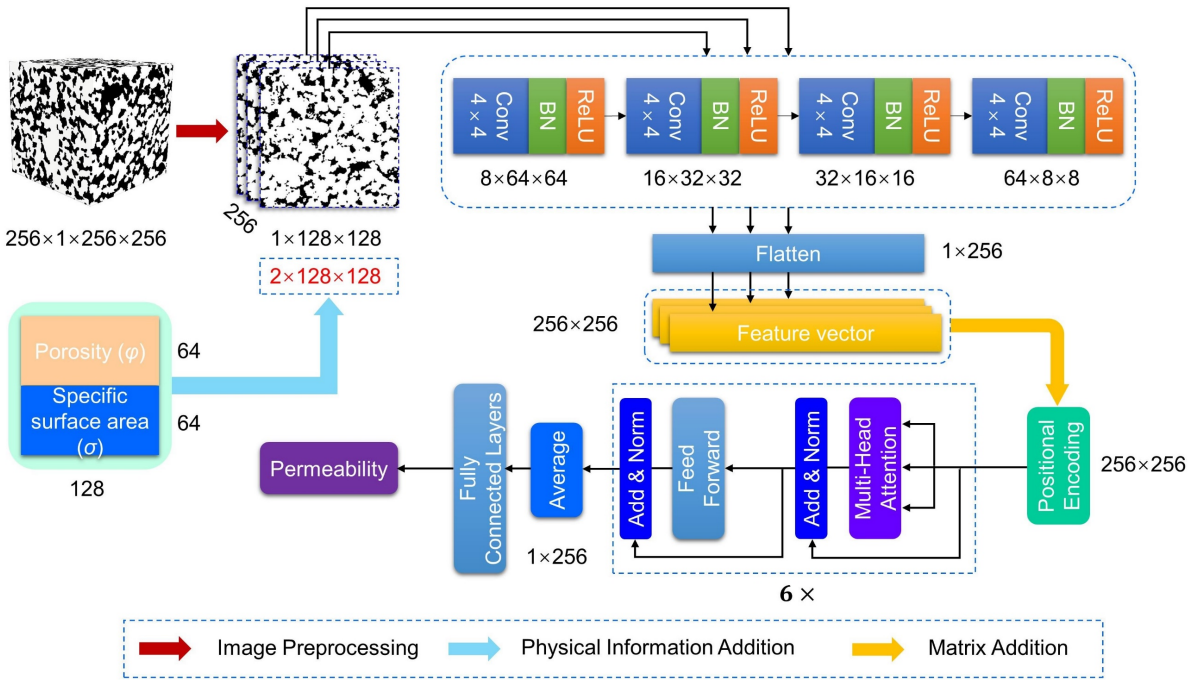

Figure 1. Schematic diagram of CNN-Transformer hybrid neural network

2 Methodology

2.1 Principles of Neural Networks

2.1.1 CNN-Transformer

Among the existing deep learning methods, CNNs perform very well in the field of computer vision. CNNs extract the local features of the image through a convolutional layer and decrease the number of features through a pooling layer, thereby reducing the number of parameters of the model and improving the training efficiency (LeCun et al., 1989). In this study, a 3D porous medium is considered as a spatial series consisting of slices of the x-y plane in the z direction. The length of the sequence is equal to the size of the 3D image in the z direction, and the thickness of each slice is equal to the size of the pixels. Each slice of the spatial series is processed by the same 2D CNN. The 2D CNN uses different convolution kernels to generate multi-channel feature maps, and then performs normalization and nonlinear transformation through Batch Norm and RuLU activation functions, respectively. CNNs generally perform downsampling through pooling layers to reduce the dimension of the feature maps. The 2D CNN used in this paper replaces pooling layers with convolutional layers. The

purpose is to make the receptive field of a single element in the feature map broader, thereby capturing large-scale features of the input image. In fact, removing pooling layers does not affect the performance of CNNs (Ruderman et al., 2018; Springenberg et al., 2014). Four convolutional layers are followed by a fully-connected layer that flattens the multi-channel feature map into a fixed-dimensional feature vector. Thus, each slice of the 3D structure is processed by the 2D CNN and is represented as a feature vector with fixed dimension. The sequence consisting of these feature vectors is used as the input to the transformer model.

Transformer has achieved huge success in the field of sequential data modeling, gradually replacing RNNs in natural language progressing (NLP), speech translation, and other applications (Devlin et al., 2018, Lüscher et al., 2019). The transformer model for computer vision, represented by the Vision Transformer (ViT), also challenges CNNs (Dosovitskiy, 2020). Bai and Tahmasebi (2022) developed a transformer-based surrogate model to provide a detailed release history of contaminants. They used the transformer model to train on the embedding space of the concentration fields, predicting the embedding features at the next time step and then reconstructing the data back to the original space. Fu et al. (2021) proposed the Stacked Auto-Encoder (SAE) network to project high-dimensional dynamical systems onto a low-dimensional nonlinear subspace and predicted the fluid dynamics based on a transformer method. Phan et al. (2022) used Image Transformer (Parmar et al., 2018) to model the distribution of rock image contents with tractable likelihood. The proposed workflow contained VQ-VAE, Image Transformer, and a size-invariant GAN to reconstruct large 3D porous structures from a single 2D rock image.

Transformer relates different positions of a single sequence in order to compute a representation of the sequence by a self-attention mechanism (Vaswani et al., 2017). Self-attention with global feature interaction outperforms RNNs such as the long short-term memory (LSTM) to effectively capture personalized patterns, since it allows feature interaction between each element in the sequence as an even stronger global inductive bias than RNNs (Jiang et al., 2022). Multi-head attention allows the model to jointly attend to information from different representation subspaces at different positions, thereby enhancing the expressive ability of the model. Since self-attention ignores the positional information of each element in the sequence, it is necessary to add positional encodings to the elements. However, transformer models can take advantage of global interaction patterns but may not be able to effectively capture local information like CNNs. For this reason, hybrids of CNNs and Transformer have been widely studied in the field of computer vision, for example in object detection (Zhu et al., 2020), visual recognition (Srinivas et al., 2021) and NLP (Chia et al., 2019). However, at present there are few relevant studies on the application of CNNs and Transformer to regression tasks.

In this work we constructed a CNN-Transformer hybrid neural network that combines the advantages of convolution with the self-attention mechanism to

capture information in order to establish the mapping relationship between the permeability of porous media and 3D digital images as a regression problem. We used a 2D CNN to compress slices of 3D porous media into fixed-dimensional feature vectors that are then composed into sequences in order. To select task-related information from a sequence, self-attention introduces a query vector and uses a scoring function to calculate the correlation between each input vector (represented by a key-value pair) and the query vector in the sequence. Key vectors are used to compute attention distributions, and value vectors are used to compute agglomerative information. The dimension of query vectors and key vectors is d_k, and the dimension of value vectors is d_v. Self-attention computes the dot products of the query vector with all key vectors, divides each by $\sqrt{d_k}$, and applies a softmax function to obtain the weights of the value vectors. When actually computing the attention function on a set of query vectors simultaneously, query vectors, key vectors, and value vectors of the input sequence are packed together into matrices Q, K, and V, respectively. The matrix of output can be written as:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (1)$$

It is beneficial to linearly project the query vectors, key vectors, and value vectors h times with different learned linear projections to d_k, d_k and d_v dimensions, respectively (Vaswani et al. 2017). The d_v-dimensional output value vectors are concatenated and once again projected, resulting in the final value vectors that are formulated as:

$$MultiHead(Q, K, V) = Concat\left(\text{head}_1, \ldots, \text{head}_h\right)W^O \quad (2)$$

$$\text{head}_i = Attention\left(QW_i^Q, KW_i^K, VW_i^V\right) \quad (3)$$

where the projections are parameter matrices $W_i^Q \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{\text{model}} \times d_k}, W_i^V \in \mathbb{R}^{d_{\text{model}} \times d_v}$ and $W^O \in \mathbb{R}^{hd_v \times d_{\text{model}}}$. In the self-attention layer, query vectors, key vectors, and value vectors can be calculated from each position in the sequence. Transformer contains encoder stacks with multiple self-attention layers, and each position in the encoder can attend to all positions in the previous layer of the encoder (Vaswani et al. 2017). In this study, the encoder of the transformer model is composed of a stack of six identical layers, and each has two sub-layers. The first is a multi-head self-attention mechanism, and the second is a simple, position-wise, fully-connected feed-forward network. The residual connection (He et al., 2016) is used for each of the two sub-layers, followed by layer normalization (Ba et al., 2016). In this paper, sinusoidal positional encodings added to the input of the transformer model can be stated by the following expressions:

$$PE_{(\mathrm{pos},2i)} = sin\left(\mathrm{pos}/10000^{2i/d_{\mathrm{model}}}\right) \quad (4)$$

$$PE_{(\mathrm{pos},2i+1)} = cos\left(\mathrm{pos}/10000^{(2i+1)/d_{\mathrm{model}}}\right) \quad (5)$$

where $2i$ and $2i$+1 denotes the $(2i)$-th and $(2i$+1)-th element of the embedding; *pos* is the global position of the embeddings in the input sequence, and d_{model} is the number of expected features in the encoder. We average the output of the encoder along the dimension of sequence length and use it as the input of an artificial neural network (ANN) to predict the permeability of 3D porous media.

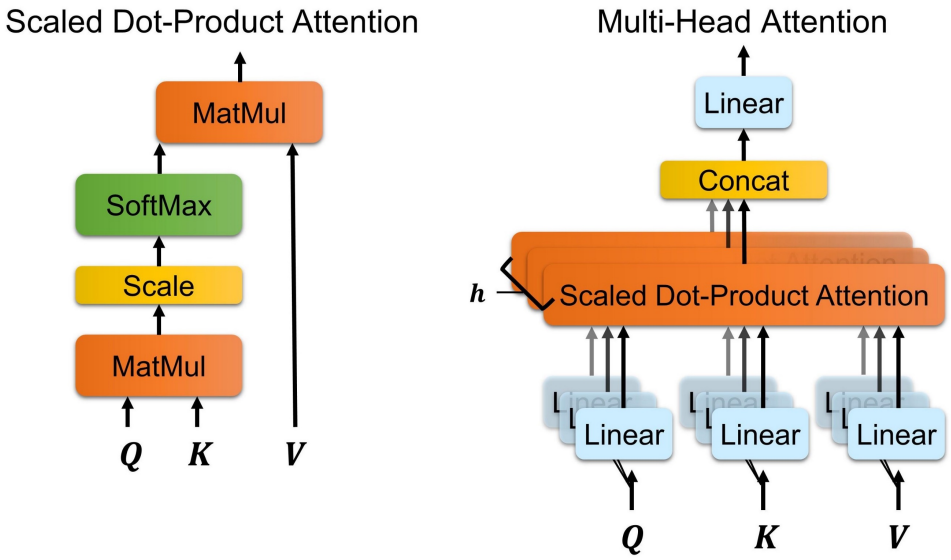

Figure 2. Scaled dot-product attention and multi-head attention

2.1.2 PhyCNN-Transformer

Adding physical parameters of porous media to neural network models is an effective method of improving the permeability prediction results (Wu et al., 2018; Yoon et al., 2020). The physics informed neural network (PINN) and the theory-guided neural network (TgNN) inspired Tang et al. (2022) to propose a method of adding global physical information to CNNs that can effectively improve the model performance. The reason is that the convolution kernels cannot implicitly extract the influential global factors of input pictures (Raissi et al., 2019, Wang et al., 2021). Tang et al. (2022) referred to the Kozeny-Carman equation (Sobieski & Zhang, 2014) that incorporates porosity , specific surface σ, and tortuosity to estimate the permeability:

$$K = \frac{c_0}{\tau^2\sigma^2}\frac{\phi^3}{(1-\phi)^2} \quad (6)$$

Tang et al. (2022) proposed that porosity and tortuosity can contribute to permeability prediction globally, and thus he included 3D matrices containing these two physical parameters into 3D digital rock images and dense blocks in DenseNet. For each 3D matrix composed of porosity and tortuosity, the upper half is porosity, and the lower half is tortuosity. The size of each physical information matrix is consistent with the original digital image size and the feature size in each dense block.

We adopted a similar approach to construct the physical information matrices. We referred to the Kozeny-Carman equation (Carman, 1939) containing porosity and specific surface area σ, which can be written as:

$$K = \frac{196}{\sigma^2} \frac{\phi^3}{(1-\phi)^2} \quad (7)$$

We needed to make two changes. First, the two global physical parameters and are expressed as sequences composed of parameters of the 2D pore structure. Each element in the sequence represents the corresponding slice of the porous medium. For example, the first element in the porosity sequence represents porosity of the first slice of the porous medium along the z direction. Second, we choose the specific surface area σ_v defined by per unit pore volume of the porous medium, not the usually adopted specific surface area σ defined by per unit bulk volume of the porous medium. Corresponding to the 2D slice, σ_v is expressed as the ratio of the total perimeters of particles to the pore area. The specific surface area σ_v in the three-dimensional state can be written as (Bear, 1988):

$$\sigma_v = A_s/U_v = A_s/\left(nU_b\right) = \sigma/n \quad (8)$$

where U_b denotes the bulk volume of the porous medium; U_v denotes the pore volume of the porous medium; A_s represents the total surface area of all particles, and n is the porosity.

In this study, digital images input to the 2D CNN are added with 2D matrices of physical parameters, the sizes of which are equivalent to those of the images. For each 2D matrix, the upper half is the porosity , and the lower half is the specific surface area σ_v. These matrices are added to the corresponding slice images along the channel direction. Specifically, the tensor of original digital images input to the 2D CNN has the size of 256 × 1 × 256 × 256. The first dimension represents the length of the sequence; the second dimension is the image channel, and the third and fourth dimensions are the sizes of the 2D image in the x and y directions, respectively. After image preprocessing in Pytorch, the digital image tensor is converted to the size of 256 × 1 × 128 × 128 to save GPU memory. When each physical parameter matrix is added

to the corresponding slice image along the dimension of the channel, the input tensor has the size of $256 \times 2 \times 128 \times 128$. There is no difference between the PhyCNN-Transformer model and the CNN-Transformer model, except that the physical parameter matrices are added to the input images.

2.2 Model Architecture of Neural Networks

In order to highlight the importance of transformer and self-attention mechanisms in this study, we constructed a relatively simple CNN structure. The CNN contains four convolutional layers with a kernel size of 4×4 plane pixels to extract features (Figure 1). The stride and padding are 2 and 1 pixels, respectively. Between each convolutional layer, we adopted a Batch Norm layer to use much higher learning rates to speed up the training and to be less careful about parameter initialization (Ioffe & Szegedy, 2015). We also applied a rectified linear unit (ReLU) as the activation function that introduces non-linearity while minimizing added computational cost (Alzubaidi et al., 2021; Elmorsy et al., 2022). The number of channels for a digital image input to the CNN is equal to 1 or 2 (when digital images are added with the physical parameter matrices). The numbers of output channels of the four convolutional layers are 8, 16, 32, and 64 in order, and finally a 64-channel, 8×8 feature map is obtained. We flattened the feature map to a feature vector with the dimension of 256 by a fully-connected layer. The procedure continues until all the 2D slice images are processed, finally forming a sequence with the length of 256. Regardless of whether physical information is added, we compress each slice image into a corresponding feature vector with the dimension of 256 to see if the PhyCNN can improve the model performance. We input such sequences to the transformer model.

In this study, the architecture of the transformer model referred to the framework of encoder stacks proposed by Vaswani et al. (2017). We adopted eight heads or parallel attention layers. We kept the dimension of the feed-forward networks in the transformer's encoder stacks to 2048 and the default dropout to 0.1. The number of expected features of the encoder stacks was 256. We averaged the output of the encoder stacks along the dimension of sequence length and fed it into four fully-connected layers that produce the output with dimensionality of 512, 1024, 512, and 1.

We set the epoch of the CNN-Transformer and PhyCNN-Transformer to 100 and saved the model parameters and prediction results of the best-performing epoch. Since transformer models requires a small learning rate, we set the initial learning rate to 1e-4. We adopted the Adam optimizer (Kingma & Ba, 2014) and set the learning rate to 0.1, 0.05, and 0.025 times the initial learning rate when the epoch was 30, 60, and 90, respectively. In neural networks, the hyperparameter batch size is limited by GPU memory and thus affects their performance (Kandel & Castelli et al., 2020; Kashefi et al., 2021; Keskar et al., 2016). We set the batch size to 8, 16, 32, and 64 while keeping the rest of the hyperparameters unchanged to observe the effect of batch size on the performance of the two models for predicting the permeability of 3D porous

media. We used the open-source machine learning library (Pytorch 1.9.0) on a computational cluster consisting of two graphics cards (NVIDIA GeForce RTX 3090 GPU) to train the models.

Table1

The architecture of the CNN-Transformer and PhyCNN-Transformer

CNN or PhyCNN	Layer Type	Filters	Kernel	Stride	Padding	Batch normalization	Activation	Total parameters
	Conv2D					Yes	ReLU	1.09M
	Conv2D					Yes	ReLU	
	Conv2D					Yes	ReLU	
	Conv2D					Yes	ReLU	
	Linear	-	-	-	-	-	ReLU	
Transformer	Layer Type	Input size	Heads	Layers	Dimension of FFN	Dropout	Layer normalization	Activation	Total parameters
	Encoder						Yes	RuLU	7.49M
	Linear		-	-	-	-	-	RuLU	
	Linear		-	-	-	-	-	-	
	Linear		-	-	-	-	-	RuLU	
	Linear		-	-	-	-	-	-	

We chose the root mean square error function (RMSE) as the loss function when training the model representing the standard deviation of prediction errors, with smaller values indicating better performance (Araya & Ghezzehei, 2019). We selected 6250 samples from the DeePore porous media dataset (Rabbani et al., 2020) that were divided into a training set and a testing set in a ratio of 8:2. Then, the training set was further divided into a training subset and a validation subset according to the ratio of 7:3. The details are listed in Section 2.3. In addition, we selected seven Fontainebleau sandstones in the shape of cubes (Berg, 2014) and generated 1000 cuboid-shaped porous media packed with spherical grains by a sedimentation-producing method (Pilotti, 1998; Vold, 1960) as the external testing set; the samples had different pore structures compared with the DeePore dataset to study the generalization performance of the models.

We evaluated the performance of the models on the validation subset and the testing set by the coefficient of determination R^2, the RMSE, and the median absolute relative error (MedARE), measures that are widely used in predicting the permeability of porous media (Elmorsy et al., 2022; Kashefi et al., 2021;

Tang et al., 2022). The evaluation functions can be stated by the following expressions:

Coefficient of Determination R^2

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \overline{y})^2} \quad (9)$$

Root mean square error function (RMSE)

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2} \quad (10)$$

Median absolute relative error (MedARE)

$$MedARE = median(|y_1 - \hat{y}_1| / y_1, \ldots, |y_n - \hat{y}_n| / y_n) \times 100\% \quad (11)$$

where n is the amount of data in the validation subset or testing set. The term y_i denotes the ground truth of the permeability, and $\hat{y}_i$ denotes the permeability predicted by models. In addition, $\overline{y}$ denotes the mean of the sample set $\{y_i\}_{i=1}^{n}$.

2.3 Datasets Preparation

Deep learning methods are data-driven, and thus building models using neural networks requires a large number of samples for training. In practice, the application of imaging techniques to achieve realistic tomography data of porous material is limited by time and costs (Song, 2019). At present, simulation is a common method used to realize 3D porous media (Graczyk & Matyka, 2020; Volkhonskiy et al., 2022; Zhang et al., 2021). We selected the 3D porous media dataset implemented by Rabbani et al. (2020) based on real micro-tomography images and data augmentation methods. The 3D porous media images they generated have the size of $256 \times 256 \times 256$ and a resolution of 5 m/voxel.

Rabbani et al. (2020) used PNM techniques (Rabbani et al., 2014; Xiong et al., 2016) to simulate single-phase fluid flow on the isotropic 3D porous samples and achieved an arithmetic average of the permeability in x, y, and z directions. PNM techniques represent a complex pore space by a network of pore bodies and pore throats with idealized geometries (Blunt, 2001; Bultreys et al., 2016). The authenticity of PNM simulation depends crucially on the correspondence between the pore network of a given medium and the pore space it represents. It also depends on how the pore network represents the real pore space according to given geometric and topological characteristics (Baychev et al., 2019; Bryant & Blunt et al., 1992). In addition, Rabbani et al. (2020) adopted a dimensionless approach where the original spatial resolution of each sample was removed, and

the unit of the permeability became px^2. This means that when the physical size of each voxel is 5 m, we need to multiply the permeability by 25 in order to achieve the permeability value in the units of Darcy.

In this study, 6250 samples selected from the DeePore porous media dataset had porosity values ranging from 0.10 to 0.45, and permeability values ranging from 6.23E-5 to 6.2 px^2. If the unit is converted to Darcy, the permeability is distributed between 0.001 and 156 Darcy. The 5000 samples in the dataset were used as the training set, and the rest of the samples were used as the testing set. During the training process, 3500 samples in the training set were used as the training subset, and the remaining 1500 samples were used as the validation subset. We fixed the random parameters used for data partitioning so that each model had an identical training subset and validation subset during training. The permeability distributions of the training subset and validation subset are shown in the box plot (Figure 3).

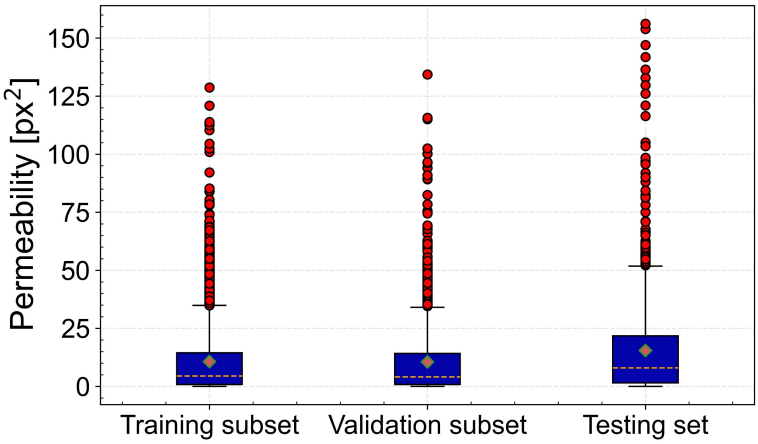

Figure 3. The permeability distribution of the sample dataset. The upper and lower limits of the box are determined by the lower quartile and the upper quartile. The dashed line in the box denotes the median, and the diamond represents the mean. The line segments at the ends of whiskers represent the minimum and maximum values, respectively, except for outliers, and the red dots represent outliers. The permeability distribution of the testing set is significantly different from that of the training subset or validation subset.

In this work, we used seven Fontainebleau sandstones that had volumes of size 480 × 480 × 480 cubic voxels, and the spatial resolution was 5.4 m/voxel. The porosity of the samples and the permeability values calculated by PNM are shown in Table 1. Moreover, we generated 1000 porous media packed with spherical grains by the sedimentation-producing method that has the advantage of retaining topological characteristics of actual media. The periodical domain boundary conditions were adopted to place grains periodically along the x-axis and y-axis, perpendicular to the z-axis, the direction of water flow. We calcu-

lated the permeability by the D3Q19 model of LBM that can conveniently solve the Navier-Stokes equation via a bounce-back scheme to deal with the non-slip boundary conditions (Qian et al., 1992; Wolf-Gladrow, 2004). The generated samples used to test the generalization abilities of the models had an average permeability of 416.3 Darcy with standard deviation of 220.4 Darcy, and an average porosity of 0.34 with standard deviation of 0.027. The size of the generated porous medium was 400 × 128 × 128 in the shape of a cuboid, and the spatial resolution was 20 m/voxel. We applied the same dimensionless approach to these samples as DeePore, expressing the permeability values in units of px^2 to fit the range of permeability values in the training set.

Table2

Seven Fontainebleau sandstones from Digital Rocks Portal to test the generalization of the models

	Porosity	Permeability	Permeability
Rock	[-]	[(m)^2] 5.7 m /voxel	[(px)^2] 10.7 m/voxel
a			
b			
c			
d			
e			
f			
g			

Note. Each rock sample was downscaled to 256^3 cubic voxels, and the spatial resolution was transformed from 5.7 m /voxel to 10.7 m /voxel.

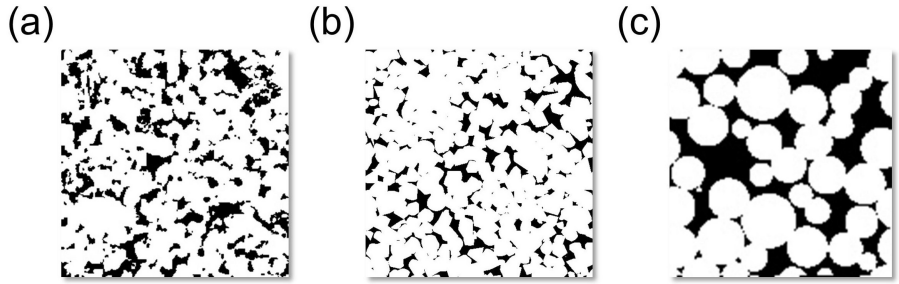

Figure 4. The slices in the middle of the 3D porous media structure, where the black indicates pores. (a) A sample with the porosity of 0.24 is selected from DeePore dataset. (b) A sample with the porosity of 0.153 is the fourth of seven Fontainebleau sandstones. (c) A sample with the porosity of 0.35 chosen from 1000 generated samples.

3 Model Performance

We introduce the results of permeability prediction on validation subset and testing set from DeePore porous media dataset using the CNN-Transformer and PhyCNN-Transformer models. The performance of the two models with different batch sizes in terms of the evaluation scores (R^2, RMSE and MedARE) and the loss curves of the validation subset are shown in Figure 5 and Figure 6. The input images of the PhyCNN-Transformer, as described in Section 2.1.2, were added with physical parameter matrices, and the structure, training method, and equipment of the PhyCNN-Transformer were no different from those of the CNN-Transformer. As the batch size increased, the R^2 score of the CNN-Transformer increased initially and then decreased, and the corresponding RMSE first decreased and then increased. When the batch size was 16, the largest R^2 and the smallest RMSE were obtained, being 0.9857 and 0.0732, respectively (Figure 7a). The MedARE also reached a minimum value of 8.41%. The PhyCNN-Transformer was not as sensitive to batch size changes as the CNN-Transformer. When the batch size was 32, the R^2 of the PhyCNN-Transformer achieved the maximum value of 0.9907; the RMSE achieved the minimum value of 0.0590 (Figure 7b), and the MedARE was 7.82%. The predictions of the two models with different batch sizes were synthesized. The average R^2 score, RMSE, and MedARE of the CNN-Transformer model were 0.9827, 0.0802, and 13.54%, respectively, while the average R^2 score, RMSE, and MedARE of the PhyCNN-Transformer model were 0.9902, 0.0606, and 8.3%, respectively, showing a significant improvement in the prediction performance.

Different batch sizes may lead to different convergence behavior for the same model, as illustrated by the RMSE loss curve of the validation subset (Figure 6). The loss curves of the CNN-Transformer (indicated by red lines in the figure) had a large degree of oscillation and did not easily converge. This was most obvious when the batch size was 8, and the RMSE loss of the model was almost greater than 0.1 during the training process. When the batch size was 16, although the RMSE loss curve failed to completely converge, the loss value reached the minimum with the epoch equal to 83. When the batch size was 32 and the epoch was greater than 90, the RMSE loss curve gradually converged to 0.08, while when the batch size was 64 and the epoch was greater than 72, the RMSE loss curve gradually converged to 0.09. These results show that increasing the batch size makes the CNN-Transformer model converge faster, while the final convergence value of the loss curve also increases.

The loss curve of the PhyCNN-Transformer was insensitive to changes in batch size with little oscillation and rapid convergence. When the batch size was 32 and the epoch was 83, the loss curve achieved the smallest RMSE. The loss curve results of the models with four batch sizes on the validation subset showed that the PhyCNN-Transformer makes significant progress in training stability compared to the CNN-Transformer. The average training time of the CNN-Transformer was 45.3 hours, and each epoch required an average of 27.2 minutes, while the average training time of the PhyCNN-Transformer was 48.3 hours, and

each epoch lasted an average of 29.0 minutes. Considering that adding physical parameter matrices improved the training stability and prediction performance of the model, the training efficiency of the PhyCNN-Transformer is acceptable.

We selected the two models achieving the best prediction results on the validation subset, CNN-Transformer with the batch size of 16 and PhyCNN-Transformer with the batch size of 32, to predict 1250 samples of the testing set. The two models also achieved satisfactory prediction results for the samples with the permeability distribution different from that of the training set (Figure 7d and 7e). The CNN-Transformer obtained an R^2 score of 0.9709, a RMSE of 0.1390, and a MedARE of 15.3%. The PhyCNN-Transformer provided an R^2 score of 0.9750, a RMSE of 0.1287 and a MedARE of 10.9%.

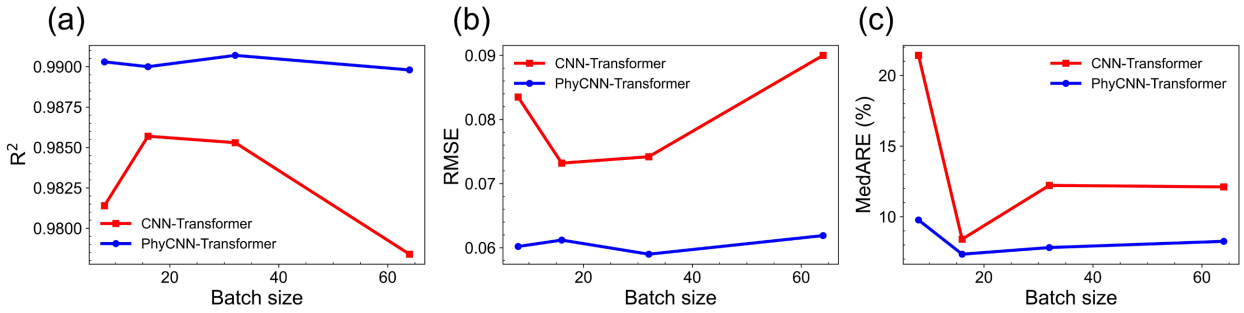

Figure 5. Effect of the batch size on the performance of the models for predicting the permeability of the validation subset. (a), (b) and (c) show how the respective R^2, RMSE, and MedARE of the models change with different batch sizes.

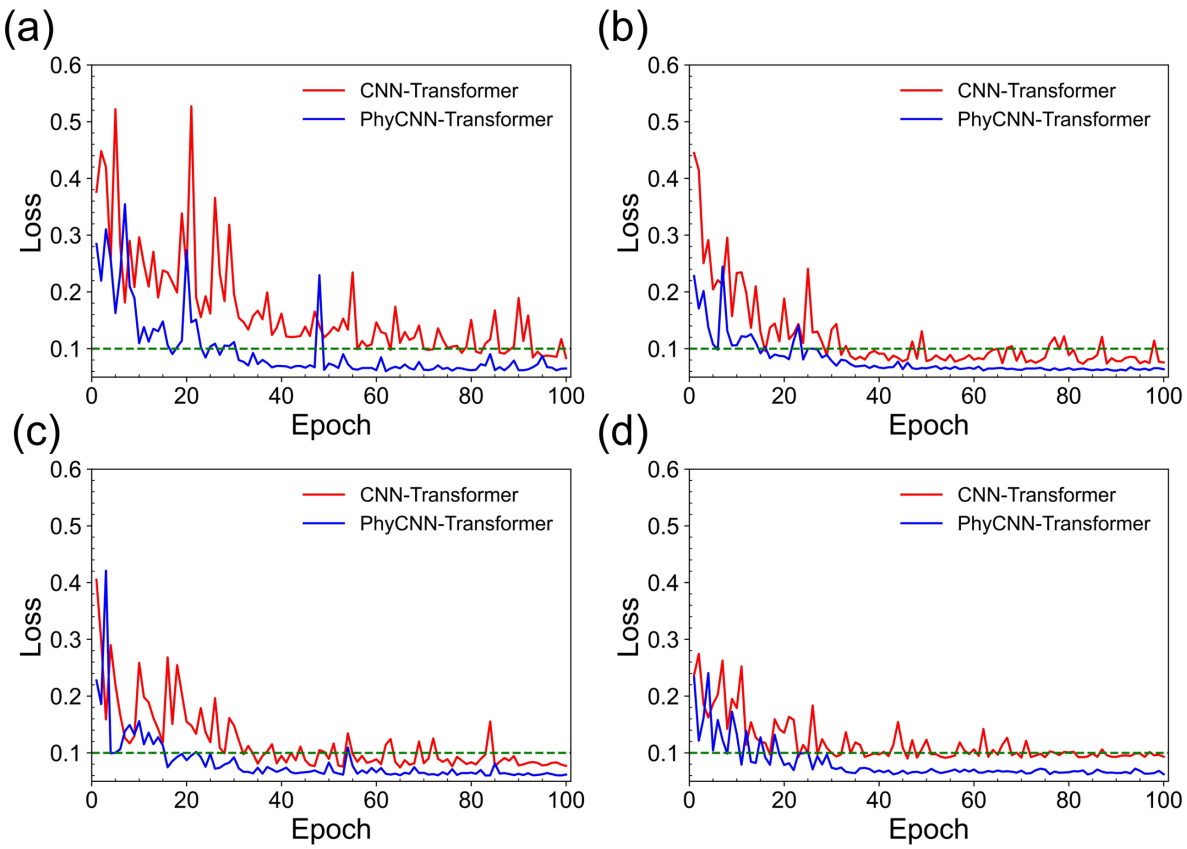

Figure 6. Loss curves of the CNN-Transformer and PhyCNN-Transformer on the validation subset: (a), (b), (c) and (d) for the batch sizes of 8, 16, 32, and 64, respectively.

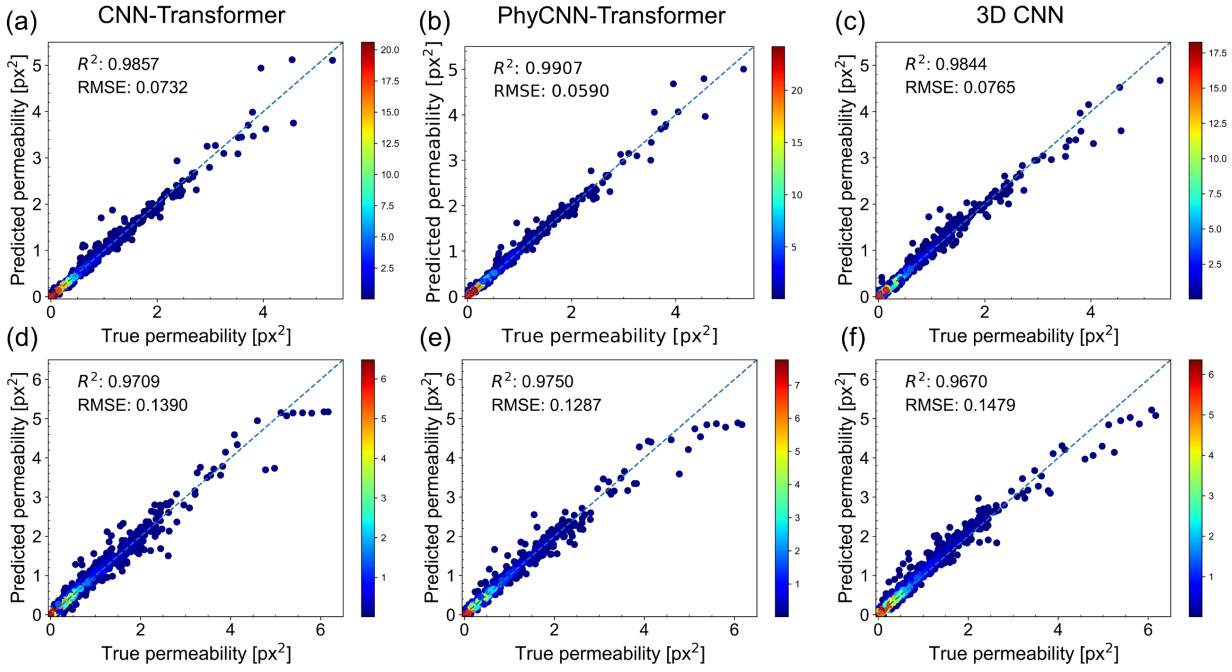

Figure 7. The prediction performance of different models on the validation subset and testing set from DeePore**:** (a), (b) and (c) for the CNN-Transformer, PhyCNN-Transformer, and 3D CNN on the validation subset; (d), (e) and (f) for the CNN-Transformer, PhyCNN-Transformer, and 3D CNN on the testing set. More samples are distributed in the range of values where the red dots are located. As the true permeability increases, the predicted permeability does not exactly agree with the true permeability.

In order to compare the differences between the CNN-Transformer and PhyCNN-Transformer proposed in this study and the common 3D CNN model in terms of prediction performance and generalization ability, we introduced a 3D CNN model with four convolutional layers, kernel sizes of $4 \times 4 \times 4$ cubic voxels, a stride size of 2 voxels, and a padding size of 1 voxel; this was a 3D version of the CNN used in this study. There were also Batch Normalization and ReLU activation functions after each layer of convolution. The 3D CNN model compressed 3D digital images with the size of $256 \times 256 \times 256$ into feature vectors with the dimension of 256 and then fed them into five fully-connected layers that had output dimensions of 2048, 1024, 512, 256, and 1. The model adopted the same training and testing sets as the CNN-Transformer and PhyCNN-Transformer. The epoch of the 3D CNN for training was increased to 150, and the initial learning rate was 1e-3. With epochs of 20, 50, 80, and 110, the learning rate was set to 0.1 times, 0.05 times, 0.025 times, and 0.01 times the initial learning rate in order. When the batch size was equal to 8, the 3D CNN achieved an R^2 of 0.9844, a RMSE of 0.0765, and a MedARE of 13.61% on the validation

subset. The trained 3D CNN yielded an R^2 of 0.9670, a RMSE of 0.1479, and a MedARE of 13.99% on the testing set with 1250 samples. The prediction performance of the 3D CNN on both the validation subset and testing set was inferior to those of the CNN-Transformer and PhyCNN-Transformer. The training parameters of the CNN-Transformer and PhyCNN-Transformer were 8.59 M, while the training parameters of the 3D CNN model were 20.2 M, a value 2.35 times those of the two transformer-based models. In terms of the training efficiency, the 3D CNN used an average of 29.6 minutes per epoch, increases of 8.82% and 2.09% compared to the CNN-Transformer and PhyCNN-Transformer, respectively.

The prediction performance of the models on previously unseen samples is crucial to evaluating their generalization, and it is also the key to comparing the differences between the proposed transformer-based models and the 3D CNN. First, we use seven Fontainebleau sandstones to test the prediction performance of the models on real samples. The porosity of the samples ranged from 8% to 26%, and the permeability distribution ranged from 0.043 to 8.040 Darcy. For the Fontainebleau sandstones with the size of 480^3 cubic voxels and a voxel length of 5.7 m, we downscaled their size to 256^3, so the corresponding voxel length became 10.7 m, and then we converted the unit of the permeability to px^2. The transformer-based models and the 3D CNN predicted the permeability of these rock samples. The 3D CNN model obtained a prediction with an R^2 of 0.5484, a RMSE of 0.0158, and a MedARE of 45.5%. The R^2 values of the CNN-Transformer and PhyCNN-Transformer were 0.9559 and 0.9197, respectively. The RMSE of the CNN-Transformer was 0.0049, while the RMSE of the PhyCNN-Transformer was 0.0067. The MedARE of the PhyCNN-Transformer was 22.7%, while the MedARE of the CNN-Transformer was 47.0%. The results showed that the generalization performance of the transformer-based models was much better than that of the 3D CNN. Although the 3D CNN achieved fine scores on the testing set with 1250 samples, its performance was not satisfactory for previously unseen samples. Compared with the CNN-Transformer model, the MedARE of the PhyCNN-Transformer model was lower but its RMSE was higher, indicating that the overall robustness of the PhyCNN-Transformer model was better, but the prediction on individual samples had a larger error. For example, for rock (g) with the porosity of 0.245, the true permeability was 0.069 px^2, while the predicted permeability was 0.086 px^2, an overestimate by 24.64% (Figure 8). The large porosity of the sample makes its pore structure complex, which may cause the porosity and the specific surface area $_v$ to fail to represent the pore structure well, resulting in a large error.

Secondly, we tested the prediction performance of the models on cuboid-shaped porous media whose pore structure was different from that of the training samples. These samples differed significantly from the training samples in terms of shape, size, solid particle shape, and permeability calculation method, prompting us to consider the transfer learning method to predict their permeability. Transfer learning takes the trained model as the basis, regards its network weights as the initial weights, and uses the back propagation algorithm to re-

train the new samples. The 3D CNN model represents the training samples of size 256^3 as 64-channel feature maps with the size of $16 \times 8 \times 8$, and then flattens them into feature vectors with the dimension of 256. However, the trained 3D CNN model cannot adapt to the size change of the 3D pore structure in the z direction, while using an image-downscaling approach to a cuboid-shaped porous medium to adapt the network will result in a large distortion of the digital image. The hybrid neural networks proposed in this paper solve this difficulty by transfer learning. We increased the length of the positional encodings of these two transformer-based models to 400 and divided the 1000 samples into training sets and testing sets at ratios of 7:3, 5:5, and 3:7. The predicted permeability of the CNN-Transformer and PhyCNN-Transformer with the testing set versus the true permeability calculated by the LBM method is shown in Figure 9a-f. The models maintain the satisfactory prediction performance when the number of training samples is gradually reduced and the number of testing samples is continuously increased. After training a large number of samples, the transformer-based models have been able to learn the mapping relationship between the pore structure and permeability, and then with the help of transfer learning, the prediction ability of the models can be generalized to different types of porous media.

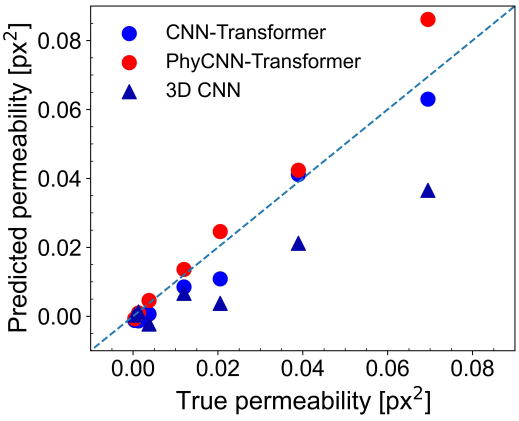

Figure 8. Permeability prediction performance of different models on seven Fontainebleau sandstones.

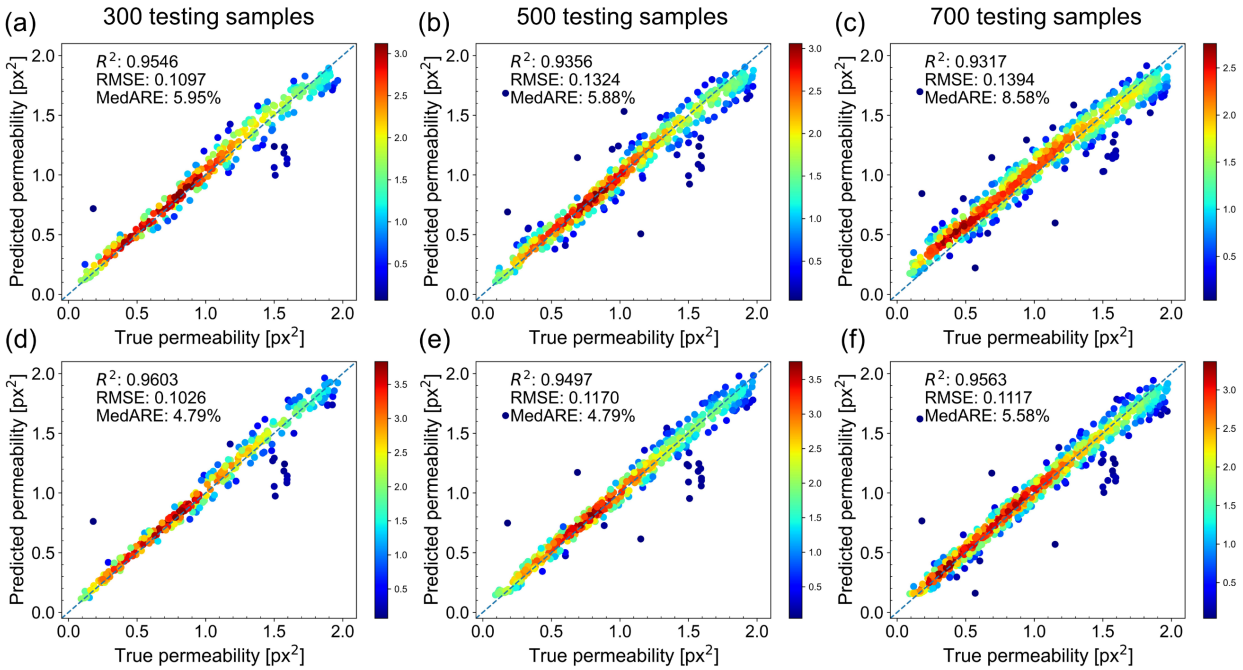

Figure 9. The performance of permeability (calculated by LBM) prediction of the CNN-Transformer and PhyCNN-Transformer on cuboid-shaped porous media packed with spherical grains by transfer learning: (a), (b), and (c) for 300, 500, and 700 testing samples with the CNN-Transformer; (d), (e), and (f) for 300, 500, and 700 testing samples with the PhyCNN-Transformer.

4 Conclusions

In this study, we developed a CNN-Transformer hybrid neural network that combines convolution with self-attention mechanism to predict the permeability of porous media. We treated 3D porous media structures as sequences composed of spatially continuous 2D slices and built a deep learning model connecting a 2D CNN with a transformer model. We constructed a 2D CNN with four convolutional layers by using the same convolution kernel size and doubling the number of channels to extract features of each slice of a porous medium, and then used these to form a sequence. The transformer model then used the self-attention mechanism to learn correlations between features of slices in the sequence, eventually predicting the permeability. Based on the CNN-Transformer hybrid neural network, we added porosity and specific surface area matrices to 3D digital images of the porous media, providing a PhyCNN-Transformer model with physical information. We used the DeePore porous media dataset (Rabbani et al., 2020) to adequately train and test the models. The true permeability of the selected samples was calculated by the PNM method with values ranging from 0.001 to 156 Darcy (Figure 3). We also selected seven real rocks (Figure 4b)

from *Digital Rocks Portal* and generated 1000 porous media packed with spherical grains (Figure 4c) with the permeability calculated by LBM to examine the generalization abilities of the models.

The trained CNN-Transformer (Figures 7a and 7d) and PhyCNN-Transformer models (Figures 7b and 7e) achieved better prediction performance than the 3D CNN model (Figures 7c and 7f) on both validation and testing sets, while the PhyCNN-Transformer model outperformed the CNN-Transformer model. The two transformer-based models exhibited excellent generalization when testing the two previously unseen types of samples. For the seven real rock samples, the R^2 value of the CNN-Transformer was 0.9559, a value that was much higher than the 0.5484 achieved by the 3D CNN model (Figure 8). With the help of transfer learning, we accurately predicted the permeability of samples calculated by LBM with a small number of the training samples (Figure 9), a result that could not be realized by the 3D CNN.

In general, our work provides a novel and potential deep learning method to quickly and accurately obtain the permeability of porous media. The model has certain inspiration for the characterization and application of a subsurface. Obtaining a broader variety of porous media with labeled properties would promote the predictive ability of our models. Increasing the complexity and adopting a more advanced transformer model could also contribute to the improvement of the method.

Data Available Statement

The data is publicly available on Zenodo, a general-purpose open-access repository (Meng, 2022).

Acknowledgments

The work was supported by the National Natural Science Foundation of China (Grant Nos. 41730856 and 41877177). We also thank the High-Performance Computing Center of Collaborative Innovation Center of Advanced Microstructures of Nanjing University for the help.

References

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., ... & Farhan, L. (2021). Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *Journal of big Data*, *8*(1), 1-74. doi:10.1186/s40537-021-00444-8

Araya, S. N., & Ghezzehei, T. A. (2019). Using machine learning for prediction of saturated hydraulic conductivity and its sensitivity to soil structural perturbations. *Water Resources Research*, *55*(7), 5715-5737. doi:10.1029/2018WR024357

Araya-Polo, M., Alpak, F. O., Hunter, S., Hofmann, R., & Saxena, N. (2020). Deep learning–driven permeability estimation from 2D images. *Computational Geosciences*, *24*(2), 571-580. doi:10.1007/s10596-019-09886-9

Ba, J. L., Kiros, J. R., & Hinton, G. E. (2016). Layer normalization. *arXiv preprint arXiv:1607.06450*. doi:10.48550/arXiv.1607.06450

Bai, T., & Tahmasebi, P. (2022). Characterization of groundwater contamination: A transformer-based deep learning model. *Advances in Water Resources*, *164*, 104217. doi:10.1016/j.advwatres.2022.104217

Baychev, T. G., Jivkov, A. P., Rabbani, A., Raeini, A. Q., Xiong, Q., Lowe, T., & Withers, P. J. (2019). Reliability of algorithms interpreting topological and geometric properties of porous media for pore network modelling. *Transport in Porous Media*, *128*(1), 271-301. doi:10.1007/s11242-019-01244-8

Bear, J. (1988). *Dynamics of fluids in porous media*. Courier Corporation.

Berg, C. F. (2014). Permeability description by characteristic length, tortuosity, constriction and porosity. *Transport in porous media*, *103*(3), 381-400. doi:10.1007/s11242-014-0307-6

Blunt, M. J. (2001). Flow in porous media—pore-network models and multiphase flow. *Current opinion in colloid & interface science*, *6*(3), 197-207. doi:10.1016/S1359-0294(01)00084-X

Blunt, M. J., Bijeljic, B., Dong, H., Gharbi, O., Iglauer, S., Mostaghimi, P., ... & Pentland, C. (2013). Pore-scale imaging and modelling. *Advances in Water resources*, *51*, 197-216. doi:10.1016/j.advwatres.2012.03.003

Bryant, S., & Blunt, M. (1992). Prediction of relative permeability in simple porous media. *Physical review A*, *46*(4), 2004. doi:10.1103/physreva.46.2004

Bultreys, T., De Boever, W., & Cnudde, V. (2016). Imaging and image-based fluid transport modeling at the pore scale in geological materials: A practical introduction to the current state-of-the-art. *Earth-Science Reviews*, *155*, 93-128. doi:10.1016/j.earscirev.2016.02.001

Bultreys, T., De Boever, W., & Cnudde, V. (2016). Imaging and image-based fluid transport modeling at the pore scale in geological materials: A practical introduction to the current state-of-the-art. *Earth-Science Reviews*, *155*, 93-128. doi:10.1016/j.earscirev.2016.02.001

Carman, P. C. (1939). Permeability of saturated sands, soils and clays. *The Journal of Agricultural Science*, *29*(2), 262-273. doi:10.1017/S0021859600051789

Chia, Y. K., Witteveen, S., & Andrews, M. (2019). Transformer to CNN: Label-scarce distillation for efficient text classification. *arXiv preprint arXiv:1909.03508*. doi:10.48550/arXiv.1909.03508

Da Wang, Y., Blunt, M. J., Armstrong, R. T., & Mostaghimi, P. (2021). Deep learning in pore scale imaging and modeling. *Earth-Science Reviews*, *215*, 103555. doi:10.1016/j.earscirev.2021.103555

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint*

arXiv:1810.04805. doi:10.48550/arXiv.1810.04805

Djabelkhir, K., Lauvernet, C., Kraft, P., & Carluer, N. (2017). Development of a dual permeability model within a hydrological catchment modeling framework: 1D application. *Science of the Total Environment*, *575*, 1429-1437. doi:10.1016/j.scitotenv.2016.10.012

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... & Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*. doi:10.48550/arXiv.2010.11929

Elmorsy, M., El-Dakhakhni, W., & Zhao, B. (2022). Generalizable Permeability Prediction of Digital Porous Media via a Novel Multi-Scale 3D Convolutional Neural Network. *Water Resources Research*, *58*(3), e2021WR031454. doi:10.1029/2021WR031454

Feng, J., Teng, Q., Li, B., He, X., Chen, H., & Li, Y. (2020). An end-to-end three-dimensional reconstruction framework of porous media from a single two-dimensional image based on deep learning. *Computer Methods in Applied Mechanics and Engineering*, *368*, 113043. doi:10.1016/j.cma.2020.113043

Fu, R., Xiao, D., Navon, I. M., & Wang, C. (2021). A data driven reduced order model of fluid flow by auto-encoder and self-attention deep learning methods. *arXiv preprint arXiv:2109.02126*. doi:10.48550/arXiv.2109.02126

Graczyk, K. M., & Matyka, M. (2020). Predicting porosity, permeability, and tortuosity of porous media from images by deep learning. *Scientific reports*, *10*(1), 1-11. doi:10.1038/s41598-020-78415-x

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778). doi:10.1109/cvpr.2016.90

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4700-4708). doi:10.1109/cvpr.2017.243

Ioffe, S., & Szegedy, C. (2015, June). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning* (pp. 448-456). PMLR.

Jiang, J., Kim, J. B., Luo, Y., Zhang, K., & Kim, S. (2022). AdaMCT: Adaptive Mixture of CNN-Transformer for Sequential Recommendation. *arXiv preprint arXiv:2205.08776*. doi:10.48550/arXiv.2205.08776

Kandel, I., & Castelli, M. (2020). The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset. *ICT express*, *6*(4), 312-315. doi:10.1016/j.icte.2020.04.010

Kashefi, A., & Mukerji, T. (2021). Point-cloud deep learning of porous media for permeability prediction. *Physics of Fluids*, *33*(9), 097109. doi:10.1063/5.0063904

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., & Tang, P. T. P. (2016). On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*. doi:10.48550/arXiv.1609.04836

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural computation*, *1*(4), 541-551. doi:10.1162/neco.1989.1.4.541

Li, J., McDougall, S. R., & Sorbie, K. S. (2017). Dynamic pore-scale network model (PNM) of water imbibition in porous media. *Advances in Water Resources*, *107*, 191-211. doi:10.1016/j.advwatres.2017.06.017

Lüscher, C., Beck, E., Irie, K., Kitza, M., Michel, W., Zeyer, A., ... & Ney, H. (2019). RWTH ASR Systems for LibriSpeech: Hybrid vs Attention--w/o Data Augmentation. *arXiv preprint arXiv:1905.03072*. doi:10.48550/arXiv.1905.03072

Meng. (2022). Permeability-Prediction-Via-CNN-Transformer (v1.0). Zenodo. https://doi.org/10.5281/zenodo.7233567

Mnih, V., Heess, N., & Graves, A. (2014). Recurrent models of visual attention. *Advances in neural information processing systems*, *27*.

Parmar, N., Vaswani, A., Uszkoreit, J., Kaiser, L., Shazeer, N., Ku, A., & Tran, D. (2018, July). Image transformer. In *International conference on machine learning* (pp. 4055-4064). PMLR.

Phan, J., Ruspini, L., Kiss, G., & Lindseth, F. (2022). Size-invariant 3D generation from a single 2D rock image. *Journal of Petroleum Science and Engineering*, 110648. doi:10.1016/j.petrol.2022.110648

Pilotti, M. (1998). Generation of realistic porous media by grains sedimentation. *Transport in Porous Media*, *33*(3), 257-278. doi:10.1023/A:1006598029153

Qian, Y. H., d'Humières, D., & Lallemand, P. (1992). Lattice BGK models for Navier-Stokes equation. *EPL (Europhysics Letters)*, *17*(6), 479. doi:10.1209/0295-5075/17/6/001

Rabbani, A., Babaei, M., Shams, R., Da Wang, Y., & Chung, T. (2020). DeePore: A deep learning workflow for rapid and comprehensive characterization of porous materials. *Advances in Water Resources*, *146*, 103787. doi:10.1016/j.advwatres.2020.103787

Rabbani, A., Jamshidi, S., & Salehi, S. (2014). An automated simple algorithm for realistic pore network extraction from micro-tomography

images. *Journal of Petroleum Science and Engineering*, *123*, 164-171. doi:10.1016/j.petrol.2014.08.020

Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, *378*, 686-707. doi:10.1016/j.jcp.2018.10.045

Ruderman, A., Rabinowitz, N. C., Morcos, A. S., & Zoran, D. (2018). Pooling is neither necessary nor sufficient for appropriate deformation stability in CNNs. *arXiv preprint arXiv:1804.04438*. doi:10.48550/arXiv.1804.04438

Saljooghi, B. S., & Hezarkhani, A. (2015). A new approach to improve permeability prediction of petroleum reservoirs using neural network adaptive wavelet (wavenet). *Journal of Petroleum Science and Engineering*, *133*, 851-861. doi:10.1016/j.petrol.2015.04.002

San Manley, S., Steindl, P., Hewitt, G. F., & Bismarck, A. (2020). An integrated method for measuring gas permeability and diffusivity of porous solids. *Chemical Engineering Science*, *223*, 115725. doi:10.1016/j.ces.2020.115725

Sobieski, W., & Zhang, Q. (2014). Sensitivity analysis of Kozeny-Carman and Ergun equations. *Technical Sciences/University of Warmia and Mazury in Olsztyn*, (17 (3)), 235-248.

Song, S. (2019). An improved simulated annealing algorithm for reconstructing 3D large-scale porous media. *Journal of Petroleum Science and Engineering*, *182*, 106343. doi:10.1016/j.petrol.2019.106343

Springenberg, J. T., Dosovitskiy, A., Brox, T., & Riedmiller, M. (2014). Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*. doi:10.48550/arXiv.1412.6806

Srinivas, A., Lin, T. Y., Parmar, N., Shlens, J., Abbeel, P., & Vaswani, A. (2021). Bottleneck transformers for visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 16519-16529).

Tang, P., Zhang, D., & Li, H. (2022). Predicting permeability from 3D rock images based on CNN with physical information. *Journal of Hydrology*, *606*, 127473. doi:10.1016/j.jhydrol.2022.127473

Tatar, A., Shokrollahi, A., Lee, M., Kashiwao, T., & Bahadori, A. (2015). Prediction of supercritical CO2/brine relative permeability in sedimentary basins during carbon dioxide sequestration. *Greenhouse Gases: Science and Technology*, *5*(6), 756-771. doi:10.1002/ghg.1524

Tsang, C. F., Neretnieks, I., & Tsang, Y. (2015). Hydrologic issues associated with nuclear waste repositories. *Water Resources Research*, *51*(9), 6923-6972. doi:10.1002/2015wr017641

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in neural*

information processing systems, *30*.

Vold, M. J. (1960). The sediment volume in dilute dispersions of spherical particles. *The Journal of Physical Chemistry*, *64*(11), 1616-1619. doi:10.1021/j100840a004

Volkhonskiy, D., Muravleva, E., Sudakov, O., Orlov, D., Burnaev, E., Koroteev, D., ... & Krutko, V. (2022). Generative adversarial networks for reconstruction of three-dimensional porous media from two-dimensional slices. *Physical Review E*, *105*(2), 025304. doi:10.1103/PhysRevE.105.025304

Wang, K., Chen, Y., Mehana, M., Lubbers, N., Bennett, K. C., Kang, Q., ... & Germann, T. C. (2021). A physics-informed and hierarchically regularized data-driven model for predicting fluid flow through porous media. *Journal of Computational Physics*, *443*, 110526. doi:10.1016/j.jcp.2021.110526

Wolf-Gladrow, D. A. (2004). *Lattice-gas cellular automata and lattice Boltzmann models: an introduction*. Springer.

Wu, J., Yin, X., & Xiao, H. (2018). Seeing permeability from images: fast prediction with convolutional neural networks. *Science bulletin*, *63*(18), 1215-1222. doi:10.1016/j.scib.2018.08.006

Xiong, Q., Baychev, T. G., & Jivkov, A. P. (2016). Review of pore network modelling of porous media: Experimental characterisations, network constructions and applications to reactive transport. *Journal of contaminant hydrology*, *192*, 101-117. doi:10.1016/j.jconhyd.2016.07.002

Yoon, H., Melander, D. J., & Verzi, S. J. (2020). *Permeability Prediction of Porous Media using Convolutional Neural Networks with Physical Properties* (No. SAND2020-3557C). Sandia National Lab.(SNL-NM), Albuquerque, NM (United States).

Zhang, F., Teng, Q., Chen, H., He, X., & Dong, X. (2021). Slice-to-voxel stochastic reconstructions on porous media with hybrid deep generative model. *Computational Materials Science*, *186*, 110018. doi:10.1016/j.commatsci.2020.110018

Zhang, F., Teng, Q., Chen, H., He, X., & Dong, X. (2021). Slice-to-voxel stochastic reconstructions on porous media with hybrid deep generative model. *Computational Materials Science*, *186*, 110018. doi:10.1016/j.commatsci.2020.110018

Zheng, Q., & Zhang, D. (2022). RockGPT: reconstructing three-dimensional digital rocks from single two-dimensional slice with deep learning. *Computational Geosciences*, *26*(3), 677-696. doi:10.1007/s10596-022-10144-8

Zhu, X., Su, W., Lu, L., Li, B., Wang, X., & Dai, J. (2020). Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*. doi:10.48550/arXiv.2010.04159