RESEARCH ARTICLE | JANUARY 14 2026

Vision Mamba for permeability prediction of porous media

Ali Kashefi ; Tapan Mukerji

Physics of Fluids 38, 017111 (2026)
https://doi.org/10.1063/5.0307953

Physics of Fluids ARTICLE pubs.aip.org/aip/pof

Vision Mamba for permeability prediction of porous media

Cite as: Phys. Fluids **38**, 017111 (2026); doi: 10.1063/5.0307953
Submitted: 20 October 2025 · Accepted: 18 December 2025 ·
Published Online: 14 January 2026

Ali Kashefi[a] and Tapan Mukerji[b]

AFFILIATIONS

Stanford University, Stanford, California 94305, USA

[a]Author to whom correspondence should be addressed: kashefi@stanford.edu
[b]Electronic mail: mukerji@stanford.edu

ABSTRACT

Vision Mamba has recently received attention as an alternative to Vision Transformers (ViTs) for image classification. The network size of Vision Mamba scales linearly with input image resolution, whereas ViTs scale quadratically, a feature that improves computational and memory efficiency. Moreover, Vision Mamba requires a significantly smaller number of trainable parameters than traditional convolutional neural networks (CNNs), and thus, they can be more memory efficient. Because of these features, we introduce a neural network that uses Vision Mamba as its backbone for predicting the permeability of three-dimensional porous media. We compare the performance of Vision Mamba with ViT and CNN models across multiple aspects of permeability prediction and perform an ablation study to assess the effects of its components on accuracy. We demonstrate in practice the aforementioned advantages of Vision Mamba over ViTs and CNNs in the permeability prediction of three-dimensional porous media. We believe the proposed framework has the potential to be integrated into large vision models in which Vision Mamba is used instead of ViTs.

I. INTRODUCTION AND MOTIVATION

Porous media play a central role across diverse scientific and industrial domains, including digital rock physics,[1–3] membrane systems,[4,5] geological carbon storage,[6,7] and medicine.[8–10] Conventional investigations use numerical simulations and laboratory experiments to analyze porous media and to obtain their physical and geometric characteristics. Although both approaches are valuable, they are resource-intensive, requiring substantial computation, specialized lab instrumentation, and considerable wall-clock time. To reduce this burden, deep learning within the broader machine-learning paradigm can accelerate tasks such as segmentation of porous media[11–14] and the prediction of porous-medium properties, including permeability,[15–23] porosity,[24] elasticity,[25,26] and effective diffusivity.[27] Moreover, deep learning configurations can predict pore-scale fields such as velocity and pressure.[28–30] Additionally, generative deep learning models are used for porous-media reconstruction.[31–34] In the present work, we focus on predicting the permeability of porous media from digital rock images using supervised deep-learning frameworks.

From a computer-science perspective, a variety of deep-learning frameworks have long been applied to permeability prediction in porous media, each with its own advantages and limitations. We briefly review these approaches and then explain how our proposed deep-learning framework addresses several of their challenges while introducing new capabilities. Convolutional neural networks (CNNs) and CNN-based variants such as ResNet[35] have been widely used to predict permeability from 2D and 3D representations of porous media.[16,18,19,36] These models often achieve strong accuracy with relatively simple architectures (compared with other models that will be mentioned later in this paragraph). However, they typically require a large number of learnable parameters, often more than the alternatives we will discuss later. They also operate on fixed input resolutions; a network trained on cubes of one size generally expects test data of the same size. Point-cloud neural networks, such as PointNet[37,38] and PointNet++,[39] are another family of deep-learning frameworks used for permeability prediction.[17] In this setup, the boundary between pore and grain phases of the porous medium is represented as a point cloud. The main advantage of this approach compared with CNNs is that it dramatically reduces the dataset size, since the full volumetric cubes are no longer needed.[17] Additionally, although models are usually trained with the same number of points per point cloud within a batch (each batch can still contain point clouds with different numbers of points), at test time, the number of points can vary. However, preprocessing is required to convert volumetric images of porous media into point-cloud data. Furthermore, if the number of boundary points

varies drastically across the dataset, the training procedure may face additional challenges, both in implementation and in loss-function convergence. Fourier neural operators (FNOs) have also been used for permeability prediction.[23] FNOs are invariant to input image size;[40] leveraging this property, they can be trained on porous media of different sizes simultaneously and have shown strong generalizability to unseen sizes. However, FNOs can be prone to overfitting on the training data. Another limitation is their sensitivity to hyperparameters, especially the number of Fourier modes, which introduces additional challenges for training and fine-tuning.[23] Vision Transformers (ViTs),[41] as another deep-learning architecture, have been applied to predicting the permeability of porous media.[15,42,43] In several settings, ViTs achieve competitive or superior accuracy with comparable, or sometimes fewer, trainable parameters than CNNs.[15] Moreover, unlike standard CNNs, vanilla ViTs with full self-attention can look across the entire image from the very first layer, so they pickup long-range patterns early; CNNs usually need many layers or operations like pooling or dilated convolutions to see that much of the image.

Mamba[44] was introduced as an alternative to Transformers.[45] Building on this line of work, Vision Mamba[46] was also introduced as an alternative to Vision Transformers.[41] One of the main advantages of Vision Mamba over ViTs is that it scales linearly (rather than quadratically) with the number of tokens. This motivates us to propose a deep learning framework based on Vision Mamba for predicting the permeability of porous media. Vision Mamba has been so far used for several key applications in vision tasks[46,47] such as image detection,[48] medical image classification,[49] remote sensing,[50] ocean engineering of underwater vehicles,[51] medical image segmentation,[52] and medical video segmentation.[53] It is important to note that Mamba and its vision counterpart are emerging as alternatives to Transformers and ViTs and are increasingly used as building blocks in large language and large vision models.[54–59] Demonstrating that Vision Mamba can predict properties of three-dimensional porous media is therefore significant, as it indicates a pathway to incorporating this task into future large models that perform multiple functions, including permeability estimation from volumetric data.

The key contributions of this study are as follows.

- We introduce a deep learning framework, based on the Vision Mamba architecture, for predicting the permeability of voxelized porous media.
- The proposed network leverages Vision Mamba to achieve linear scaling with token size (or similarly patch size), whereas ViTs scale quadratically.
- Leveraging Vision Mamba significantly reduces trainable parameters compared to CNNs, improving memory efficiency.

It is worthwhile to note that the focus of the current study, from a computer science perspective, is a supervised deep learning framework in which fully labeled data (i.e., pairs of cubes and permeabilities) are available. Of course, the class of weakly supervised deep learning frameworks, such as physics-informed machine learning (see, e.g., Refs. 30, 60, and 61) which is particularly useful when only sparse data are available, represents another important research direction; however, it is not the focus of the present study. We now outline the structure of the remainder of this research paper. In Sec. II, we describe the generation and collection of three-dimensional porous media and the computation of their permeability by numerically solving the Stokes equations for the proposed supervised deep-learning framework. In Sec. III, we present the Vision Mamba architecture, adapted to predict a volumetric property, here, the permeability of 3D porous media. Training of Vision Mamba and the hyperparameter settings are explained in Sec. IV. We then discuss the results in Sec. V, including the performance of Vision Mamba, its comparison with CNNs and ViTs, and the ablation studies. Finally, Sec. VI provides a summary and potential directions for future research.

II. DATA GENERATION

To test the deep-learning framework, we synthesize voxelized porous media using the truncated-Gaussian construction.[62,63] Each sample occupies a cube of side length L discretized on an $n \times n \times n$ grid, with morphology characterized by a target porosity ϕ (i.e., pore-volume fraction) and a spatial correlation length ℓ_c. To construct each volumetric sample, we follow three stages. First, we generate a $64 \times 64 \times 64$ (i.e., $n = 64$) scalar field of white noise by drawing samples from a standard normal distribution at every voxel. Second, we impose spatial correlation by convolving the field with a three-dimensional Gaussian kernel with a standard deviation of 5.0 and a spatial correlation length $\ell_c = 17$ voxels. Third, we rescale the smoothed field to the interval [0, 1] and apply a global threshold of 0.45 so that values less than or equal to 0.45 are labeled as pore and values greater than 0.45 are labeled as grain, yielding a binary pore-grain medium. The selected threshold constrains the porosity to $\phi \in [0.125,\ 0.200]$. In the present study, the characteristic domain length l is defined as $l = n\,\Delta x$, where Δx denotes the physical length associated with each side of a single pixel in the discretized porous medium. For all simulations, Δx is set as 0.003 m, thereby setting the spatial resolution of the computational grid. We generate 1692 samples and randomly partition them into three disjoint subsets: 1353 for training, 169 for validation, and 170 for testing. A few examples of these cubic porous media are shown in Fig. 1.

Flow through each synthesized porous medium is driven by imposing a uniform streamwise pressure gradient $\Delta p/l$ in the x-direction. The two bounding y–z faces are assigned no-slip conditions. Within the pore space, we compute the steady incompressible motion using a lattice Boltzmann solver[64] that resolves the Stokes system,

$$\nabla \cdot \boldsymbol{u} = 0, \tag{1}$$

$$\nabla p - \mu \nabla^2 \boldsymbol{u} = \boldsymbol{0}, \tag{2}$$

where μ is the dynamic viscosity and $\boldsymbol{u}$ and p denote the velocity and pressure fields, respectively. The numerical simulation using the lattice Boltzmann solver is carried out until the L^2 norms of the residuals for both the continuity and momentum equations [i.e., Eqs. (1) and (2)] fall below 10^{-5}. From the converged solution, the intrinsic permeability (k) in the x-direction is obtained via Darcy's law,[65]

$$k = -\frac{\mu\, U\, l}{\Delta p}, \tag{3}$$

with U the superficial (volume-averaged) velocity evaluated over the entire sample (assigning zero velocity in solid voxels). Across the dataset, the resulting permeabilities lie within [20 mD, 200 mD].

III. VISION MAMBA ARCHITECTURE

In this section, we describe the architecture of the proposed neural network, whose core is Vision Mamba, a selective state-space

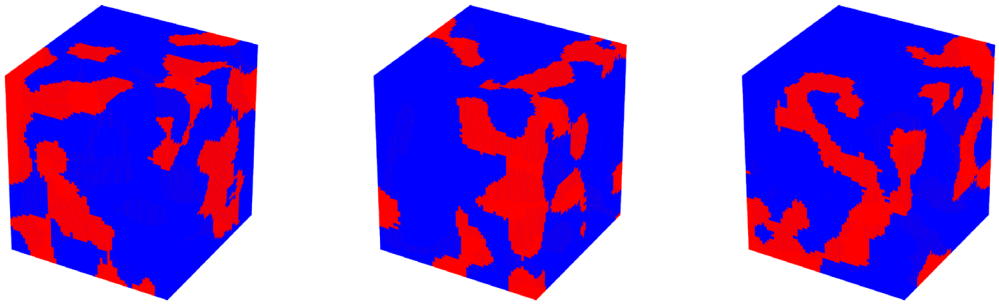

FIG. 1. Three representative examples of the synthetically generated three-dimensional digital porous media (with $n = 64$) used to train Vision Mamba are shown; phases are color-coded with blue indicating the solid grain matrix and red indicating the pore space.

model, adapted to predict permeability from voxelized porous media. Figure 2 illustrates the schematic of Vision Mamba, which serves as the core of the proposed neural network, with the 3D porous medium cube as the input. We split the cube into non-overlapping 3D patches, and each patch is embedded into a token. A stack of Vision Mamba blocks scans the embedded tokens along the depth, height, and width axes. Each scan performs bidirectional state updates through forward and backward recurrences. Finally, we take a global average over space and use a linear layer to output a single permeability value.

A. Input and patchification

The input to the network is a batch of generated porous media, i.e., a batch of cubes, which mathematically can be shown by $X \in \mathbb{R}^{\mathcal{B}\times 1\times D\times H\times W}$, where $\mathcal{B}$ is the batch size and D, H, and W are spatial dimensions. Next, we apply a patchification operator. Patchification uses a single 3D convolution with kernel size and stride equal to the patch size. In this setup, the patchification produces an $D' \times H' \times W'$ grid of patch tokens, each with $\mathcal{C}$ channels. Consequently, the output of the patchification operator is the token grid $z_{\text{tok}} \in \mathbb{R}^{\mathcal{B}\times\mathcal{C}\times D'\times H'\times W'}$.

B. Vision Mamba block

The token grid z_{tok} (obtained from the previous step) serves as the input to Vision Mamba. To elaborate on the process within Vision Mamba, we describe it in three stages: token-wise parameter generation, selective scanning along each axis, and axis fusion with residual connections.

1. Token-wise parameter generation

In the next step, a $1 \times 1 \times 1$ convolution reads $z_{\text{tok}} \in \mathbb{R}^{\mathcal{B}\times\mathcal{C}\times D'\times H'\times W'}$ and produces five fields of size $\mathcal{B} \times \mathcal{C} \times D' \times H' \times W'$: input gate ($g_{\text{in}}$), output gate ($g_{\text{out}}$), and two state-space coefficients B and C, as well as positive step size Δ. Moreover, we create a learnable vector $A \in \mathbb{R}^{\mathcal{C}}$ and a skip vector $D_{\text{skip}} \in \mathbb{R}^{\mathcal{C}}$. In addition, we define u as

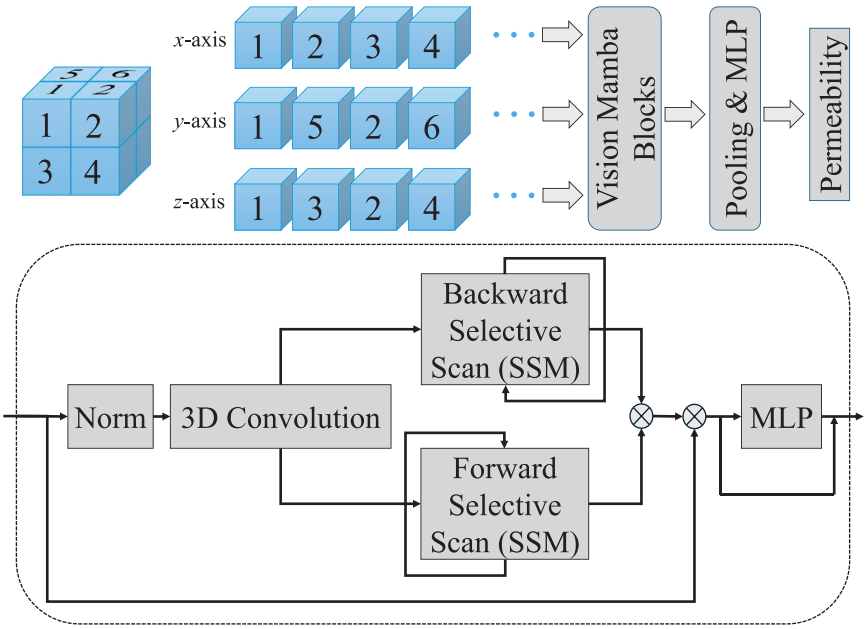

FIG. 2. Top panel: Schematic architecture of the proposed network based on Vision Mamba for deep learning of the permeability of three-dimensional porous media. If the input is a $64 \times 64 \times 64$ porous medium (i.e., $n = 64$) with a patch size of 32, there are eight subcubes, labeled 1–8. By scanning along the y-axis, the subcubes are arranged in the order 1, 5, 2, 6, 3, 7, 4, 8. Scanning along the x and z directions is defined similarly. Bottom panel: Internal structure of a Vision Mamba block.

$$u = g_{\text{in}} \odot z_{\text{tok}}, \tag{4}$$

where $\odot$ denotes elementwise product and thus $u \in \mathbb{R}^{\mathcal{B}\times\mathcal{C}\times D'\times H'\times W'}$. A_+ is introduced and computed by applying the elementwise softplus function to the vector A. The softplus function is defined as

$$\sigma(\lambda) = \ln(1 + e^{\lambda}). \tag{5}$$

Note that although $A_+ \in \mathbb{R}^{\mathcal{C}}$, it is treated as $A_+ \in \mathbb{R}^{1\times\mathcal{C}\times1\times1\times1}$ in practice from a software engineering perspective. Next, α is introduced and computed as

$$\alpha = \exp(-A_+ \odot \Delta), \tag{6}$$

where $\alpha \in \mathbb{R}^{\mathcal{B}\times\mathcal{C}\times D'\times H'\times W'}$.

2. Selective scan per axis

For each spatial axis $\eta \in \{D', H', W'\}$, the token grid z_{tok} is viewed as a collection of length-L_η sequences by treating that axis as an ordered time dimension and flattening the remaining indices into independent sequences. The forward selective state-space scan along axis η updates a channelwise hidden state s and produces an output y^{fwd} via

$$s_t = \alpha_t \odot s_{t-1} + B_t \odot u_t, \tag{7}$$

$$y_t^{\text{fwd}} = C_t \odot s_t + D_{\text{skip}} \odot u_t, \tag{8}$$

from $t = 0$ to $t = L_\eta - 1$. Note that the subscript t added to the components s, α, B, u, C, and y^{fwd} (i.e., s_t, α_t, B_t, u_t, C_t, and y_t^{fwd}) indicates that these components are reshaped to $N_{\text{seq}} \times L \times \mathcal{C}$, and therefore $\{s_t, \alpha_t, B_t, u_t, C_t, y_t^{\text{fwd}}\} \in \mathbb{R}^{N_{\text{seq}}\times\mathcal{C}}$. The value of N_{seq} depends on the scanning axis. For example, when scanning along the D' axis, $N_{\text{seq}} = \mathcal{B} \times H' \times W'$. Similarly, N_{seq} is computed when the other two axes are scanned. Finally, a corresponding backward scan runs from $t = L_\eta$ to $t = 1$, producing y^{bwd}.

3. Axis fusion and residuals

The two directions are fused to remove directional bias,

$$\hat{y} = \frac{1}{2}(y^{\text{fwd}} + y^{\text{bwd}}), \tag{9}$$

and the result is then gated at the output

$$y_\eta = g_{\text{out}} \odot \hat{y}. \tag{10}$$

The outputs along the three axes are subsequently averaged to obtain the final output

$$y_{\text{tok}} = \frac{y_{D'} + y_{H'} + y_{W'}}{3}. \tag{11}$$

The Vision Mamba block employs a residual connection and a pointwise MLP to mix channels after the selective scan, formulated as

$$z^+ = z_{\text{tok}} + y_{\text{tok}}, \tag{12}$$

$$z_{\text{out}} = z^+ + \mathcal{M}(z^+), \tag{13}$$

where $\mathcal{M}$ denotes a pointwise MLP implemented using $1 \times 1 \times 1$ convolutions with the Gaussian error linear unit ($\mathcal{G}$) activation function, defined as

$$\mathcal{G}(\lambda) = \frac{\lambda}{2}(1 + \text{erf}(2^{-0.5}\lambda)). \tag{14}$$

C. Global pooling and head

After the Vision Mamba block, a global average pooling across spatial dimensions and a linear projection to a scalar is applied. If h denotes the globally pooled representation, the predicted permeability $\hat{k}$ is computed as

$$h = \text{mean}(z_{\text{out}}), \tag{15}$$

$$\hat{k} = w^\top h + b, \tag{16}$$

where w is the weight vector and b is a scalar bias.

Note that we described the architecture of a single Vision Mamba block. Multiple blocks can be stacked sequentially to construct deeper networks. For further details on the underlying methodology, we refer readers to the original Mamba formulation[66] and its vision-oriented adaptation in Vision Mamba.[46] Moreover, implementation-specific details are documented in our openly available GitHub repository (see the Data Availability part at the end of the article), which includes extensive inline comments.

IV. PARAMETER SETUP AND TRAINING

Since $n = 64$, the parameters D, W, and H are set to 64. We set a patch size of 8 voxels in the patchification operator on 64^3 inputs, producing an $8 \times 8 \times 8$ grid of 512 tokens with embedding width $\mathcal{C} = 64$. Since the input dimensions are 64 and the patch size is 8, it follows that D', W', and H' are 8 (because $64/8 = 8$). Moreover, because $n = 64$ and the patch size equals 8, it is concluded that the sequence length along each axis is $L_\eta = 8$ (the number of pixels divided by the patch size). Additionally, we set the block depth N_{block} to 3. The block depth of 3 ($N_{\text{block}} = 3$) means that three Vision Mamba blocks are stacked sequentially after the patch-embedding stem, each applying axiswise bidirectional selective scans and residual pointwise mixing to the output of the preceding block before the global pooling and linear regression head. In Sec. V B, we report a series of ablation studies that systematically examine how these hyperparameters affect the predictive performance of the model.

Training uses mean-squared error on a min–max normalized target. Let $k_{\min}$ and $k_{\max}$ be the minimum and maximum permeability values computed on the training split. The normalized target is

$$\tilde{k} = \frac{k - k_{\min}}{k_{\max} - k_{\min}}. \tag{17}$$

The loss over a batch is

$$\mathcal{L} = \frac{1}{\mathcal{B}}\sum_{i=1}^{\mathcal{B}} (\hat{k}_i - \tilde{k}_i)^2. \tag{18}$$

At evaluation time, predictions are mapped back to physical units by the inverse of Eq. (17). This normalization stabilizes optimization without imposing a hard output range; the regression head remains unconstrained and learns to match the normalized scale.

Model training proceeds via stochastic, mini-batch gradient optimization by adopting the Adam optimizer[67] with a constant learning rate of 0.001, and using mini-batches of 128 samples (i.e., $\mathcal{B} = 128$) for each parameter update.[68] To avoid overfitting, model performance

is continuously monitored on a held-out validation set throughout training. Convergence is typically achieved within approximately 300 epochs, at which point the final optimized model is selected. All experiments are executed on a single NVIDIA A100 (SXM4) GPU equipped with 80 GB of memory.

V. RESULTS AND DISCUSSION

A. General analysis

To assess predictive accuracy for permeability, we employ the coefficient of determination, R^2. For a test set comprising P samples with ground-truth permeabilities k_i and corresponding predictions $\tilde{k}_i$, and denoting by $\bar{k} = \frac{1}{P}\sum_{i=1}^{P} k_i$ the empirical mean of the ground truth, R^2 is defined as

$$R^2 = 1 - \frac{\sum_{i=1}^{P} (k_i - \tilde{k}_i)^2}{\sum_{i=1}^{P} (k_i - \bar{k})^2}. \tag{19}$$

Note that negative values of R^2 imply performance inferior to the trivial predictor $k_i = \bar{k}$. We also report the root mean square error (RMSE), defined as

$$\mathrm{RMSE} = \sqrt{\frac{1}{P}\sum_{i=1}^{P} (k_i - \tilde{k}_i)^2}. \tag{20}$$

We further report the maximum relative error over the test set by computing $\max\left\{\frac{|k_i - \tilde{k}_i|}{|k_i|}\right\}_{i=1}^{P}$. The minimum relative error is similarly defined.

The performance and error analysis of the Vision Mamba model in predicting the permeability of the test set (170 porous media) are summarized in Table I. As reported, the R^2 score is 0.9969, and the root mean square error is 2.6939 mD. The maximum and minimum relative errors are 0.2708 and 0.0003, respectively. The left panel of Fig. 3 further illustrates the predicted vs ground-truth permeability for all samples in the test set, highlighting the results for individual porous media. These findings demonstrate the successful training and accurate predictive capability of the proposed Vision Mamba-based neural network for applications to three-dimensional porous media.

With the available GPU, the Vision Mamba framework evaluates the permeability of the entire test set of 170 cubes in roughly 7 s. By contrast, computing the permeability of the same 170 samples with the Lattice Boltzmann solver requires, on average, about 1530 s, which is approximately 26 min, on a single Intel(R) Core CPU operating at 2.30 GHz. This corresponds to an average acceleration factor of about 218 relative to the numerical simulations. These values should be viewed as indicative rather than absolute, since the realized speedup depends strongly on the efficiency of the numerical solver implementation and on the specific GPU and CPU hardware employed.

TABLE I. R^2 score, root mean square error, and minimum/maximum relative errors of the test set (170 samples) for the comparison between Vision Mamba and CNN models. The batch size ($\mathcal{B}$) for both models is set to 128. We set $N_{\mathrm{block}} = 3$ and as well as the patch size of 8 in the Vision Mamba model.

	Vision Mamba	CNN
R^2 score	0.9969	0.9762
Root mean square error (mD)	2.6939	7.5054
Minimum relative error	0.0003	0.0004
Maximum relative error	0.2708	0.9312
Training time per epoch (s)	3.0	1.6
Number of trainable parameters	195 841	2 582 369

B. Comparison between Vision Mamba and CNNs

The next step is to compare the performance of Vision Mamba with that of a CNN. For completeness, we briefly outline the CNN architecture used in this study. Specifically, we employ the same CNN model that was adopted in our previous work published in 2021.[17] In simple terms, the CNN model consists of an encoder and a decoder. In the encoder, convolutional channels start at 16 and double at each stage. Downsampling is done with stride-2 convolutions without padding. There are no pooling layers in the encoder. We use $2 \times 2 \times 2$ kernels, except for the last layer of the encoder, which uses a $1 \times 1 \times 1$ kernel. This final layer produces a single global latent vector of length 1024. This latent vector is then passed to a decoder, implemented as a multilayer perceptron (MLP) with three layers of sizes 512, 256, and 1, respectively, which is used to predict the permeability. In both the encoder and decoder, Rectified Linear Unit (ReLU) activation function [see Eq. (7) in Ref. 17 for the mathematical expression of this function] is applied as the activation function after each layer except the final layer, which has no activation. Batch normalization[69] is applied after each layer. In the decoder, dropout[70] with a probability of 0.7 is used. Similar to the Vision Mamba model, the loss function is the mean squared error [Eq. (18)]. For additional background and implementation details on CNNs for permeability prediction in porous media, see Ref. 17.

The performance and error analysis of the predicted permeability values of the test set using the CNN model are listed in Table I, with the corresponding results illustrated in the right panel of Fig. 3 for the R^2 score. In comparison with Vision Mamba, the CNN yields a lower R^2 score (0.9762 vs 0.9969), a higher root mean square error (7.5054 mD vs 2.6939 mD), a higher minimum relative error (0.0004 vs 0.0003), and a higher maximum relative error (0.9312 vs 0.2708). It is important to emphasize that our focus here is not solely on showing that Vision Mamba consistently outperforms CNN in terms of prediction accuracy of porous media permeability. In fact, we ensured that the CNN model was optimized to achieve its best possible performance. Nevertheless, as discussed earlier, CNN still achieves lower R^2 scores compared to Vision Mamba. Instead, our comparison primarily concerns the training time and the number of trainable parameters, as summarized in Table I. The training time (per epoch) of Vision Mamba is approximately 1.875 times longer than that of CNN. This can be attributed to the sequential nature of Vision Mamba, which converts each three-dimensional porous medium into a sequence of patches and processes them serially. In contrast, CNNs process three-dimensional porous media through multiple channels in parallel, where the number of channels typically increases and the kernel size decreases at deeper layers, leading to faster training. However, this parallelization comes at the cost of requiring more trainable parameters and higher GPU memory consumption. Based on Table I, the number of trainable parameters in the CNN model is 2 582 369, whereas

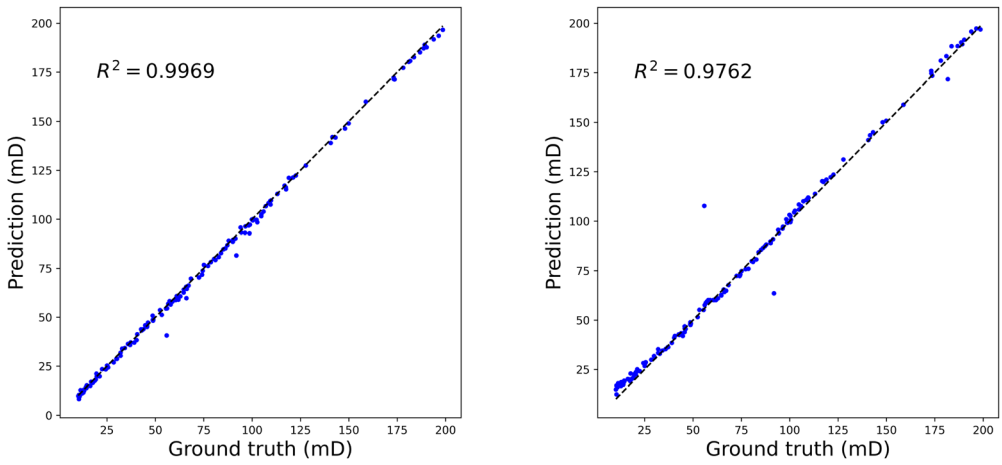

FIG. 3. Comparison of Vision Mamba and CNN performance using the R^2 score (left: Vision Mamba; right: CNN).

Vision Mamba requires only 195 841 parameters, approximately a 13.2-fold reduction, which is a substantial difference.

C. Comparison between Vision Mamba and ViT

This subsection compares the proposed model, based on Vision Mamba, with ViT for permeability prediction of porous media. A brief summary of the ViT architecture is provided at the end of this subsection; here, we focus on the results. In each machine learning experiment, the number of trainable parameters between the two models is matched as closely as possible. Once the initial ViT design is established, the only variable across experiments is the patch size (i.e., token size). The Vision Mamba configuration follows that described previously, except that the patch size is varied. For both models, the batch size is fixed at 128 ($\mathcal{B} = 128$). Other hyperparameters and training procedures are selected to achieve the best performance, and early stopping is applied to mitigate overfitting. Table II reports the results for patch sizes 4, 8, 16, and 32. In both models, reducing the patch size decreases the number of trainable parameters but increases GPU memory requirements. The character of this increase distinguishes

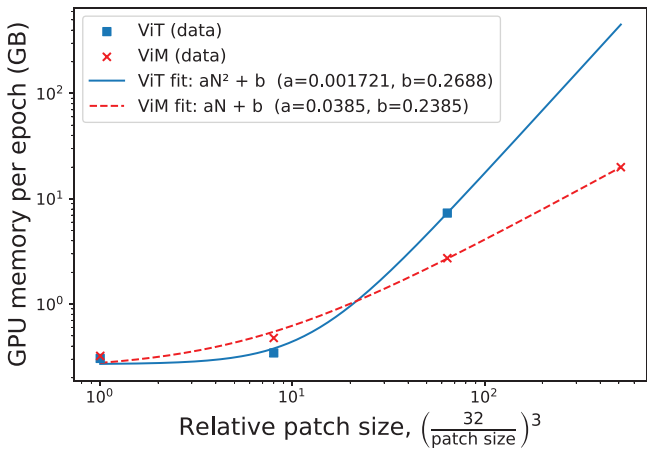

FIG. 4. GPU memory (per epoch) scaling vs relative patch size (i.e., relative token size) for Vision Mamba (ViM) and ViT on log–log axes. Least-squares fits reveal linear scaling for Vision Mamba and quadratic scaling for ViT.

TABLE II. Performance comparison between Vision Mamba (ViM) and Vision Transformer (ViT). Reported metrics include the R^2 score, root mean square error (RMSE), minimum relative error (MiRE), and maximum relative error (MaRE) on the test set (170 samples) for different patch sizes. See text for details of the setup for each architecture. The symbol × indicates that the corresponding machine learning experiment could not be run due to GPU memory limitations.

Patch size	4		8		16		32	
Model	ViM	ViT	ViM	ViT	ViM	ViT	ViM	ViT
GPU memory per epoch (GB)	19.967	×	2.732	7.318	0.477	0.344	0.322	0.305
Training time per epoch (s)	12.0	×	3.0	2.7	2.5	2.1	2.6	2.3
Trainable parameters	167 169	187 073	195 841	215 745	425 217	445 121	2 260 225	2 280 129
R^2 score	0.9934	×	0.9969	0.9838	0.9974	0.9957	0.9817	0.9491
RMSE (mD)	3.9571	×	2.6939	6.1794	2.4557	3.1903	6.5718	10.9623
MiRE	0.0001	×	0.0003	0.0001	0.0001	0.0001	0.0001	0.0001
MaRE	0.4478	×	0.2708	0.6973	0.2105	0.3843	0.4586	0.7567

Vision Mamba from ViT. Although the number of trainable parameters remains nearly constant in both models as patch size decreases, GPU memory usage grows at different rates. For ViT, the increase is so pronounced that a model with patch size 4 cannot be executed on an 80-GB NVIDIA A100 (SXM4) GPU, resulting in job failure, as reported in Table II. Figure 4 shows the GPU memory usage per epoch as a function of patch size. In this plot, the patch size is normalized by the largest patch size (i.e., 32 in the current case). As shown in Fig. 4, the required GPU memory per epoch increases linearly with decreasing patch size in Vision Mamba, whereas it increases quadratically with decreasing patch size in ViT. Our experimental results on three-dimensional porous media (i.e., a specific 3D image) confirm the theoretical design of Vision Mamba and ViT in terms of linear and quadratic scaling with token size. As illustrated in Fig. 4, we apply a least squares fit to derive linear and quadratic equations describing the experimental GPU memory requirements. Based on these equations, it is predicted that at a patch size of 4, the required GPU memory per epoch for ViT would be approximately 451 GB, which explains why this machine learning experiment could not be executed on our 80-GB GPU.

Figure 5 presents the permeability predictions vs the ground truth for patch sizes 8, 16, and 32 using the Vision Mamba and ViT models. Based on the results reported in Table II and the visualizations in Fig. 5, it can be concluded that very large patch sizes reduce the accuracy of permeability prediction. This effect is more pronounced for ViT, which attains an R^2 score of 0.9491 at a patch size of 32, whereas Vision Mamba maintains an R^2 score of 0.9817 at the same patch size. Overall, Vision Mamba achieves higher accuracy and, owing to its lower memory footprint, allows exploration of smaller patch sizes. This advantage is expected to become more significant for larger porous media (i.e., higher values of n) and for media with shorter spatial correlation lengths. According to Table II, the ViT model generally requires less time per epoch, although the difference from the Vision Mamba model is not substantial.

At the end of this subsection, we provide a brief explanation of the ViT architecture implemented in this study. The network partitions each $64 \times 64 \times 64$ porous medium into non-overlapping patches (e.g., $8 \times 8 \times 8$ patches when the patch size is 8) and maps each patch to a token via a three-dimensional convolutional stem whose kernel and stride are equal to the patch size, producing embeddings of dimension 64. A learned absolute three-dimensional positional embedding, defined on a base token grid (e.g., $8 \times 8 \times 8$ grid for a patch size of 8), is trilinearly interpolated to the current token grid and added to the tokens. The encoder consists of three pre-normalized Transformer blocks; in each block, tokens are normalized, processed by multi-head self-attention with eight heads, and merged back through a residual connection. This is followed by another normalization, a two-layer MLP with Gaussian error unit activation and dropout, and a second residual addition. After the block stack, tokens are layer-normalized and aggregated by global average pooling, and a

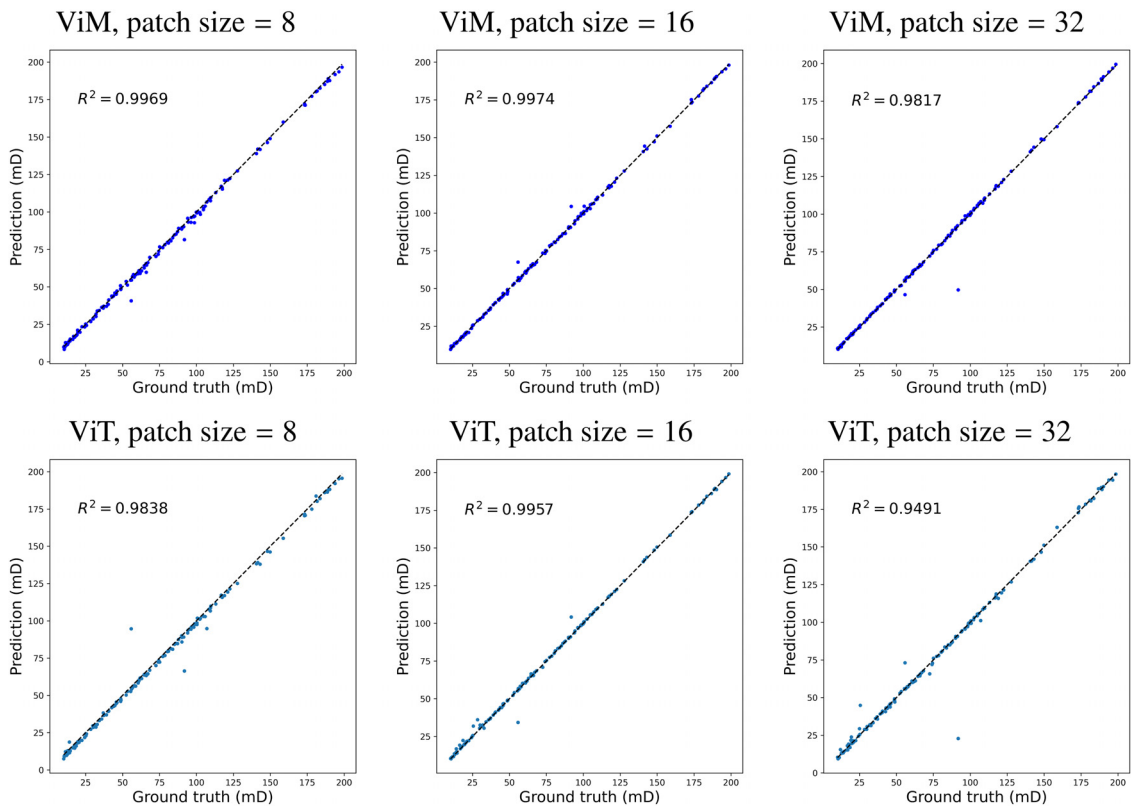

FIG. 5. Comparison of Vision Mamba (ViM) and ViT performance based on the R^2 score. The first row corresponds to Vision Mamba (ViM), and the second row corresponds to ViT.

TABLE III. R^2 score, root mean square error, and minimum/maximum relative errors of the test set (170 samples) for different numbers of Vision Mamba blocks in the proposed neural network. The patch size is fixed at 8.

Number of Vision Mamba blocks (N_{block})	1	2	3	4	5
R^2 score	0.9562	0.9803	0.9969	0.9590	0.9517
Root mean square error (mD)	10.1734	6.8171	2.6939	9.8449	10.6828
Minimum relative error	0.0030	0.0001	0.0003	0.0001	0.0001
Maximum relative error	1.4327	0.7685	0.2708	0.6538	0.6875

linear head maps the pooled representation to a single scalar permeability prediction. Further details can be found in the original Transformer[45] and Vision Transformer[41] articles, as well as in our open-source code, the link to which is provided at the end of this article.

D. Ablation studies

In this section, our objective is to examine the influence of several key hyperparameters of the neural network on its performance in predicting the permeability of porous media. While designing a neural network, it is essential to perform hyperparameter fine-tuning to

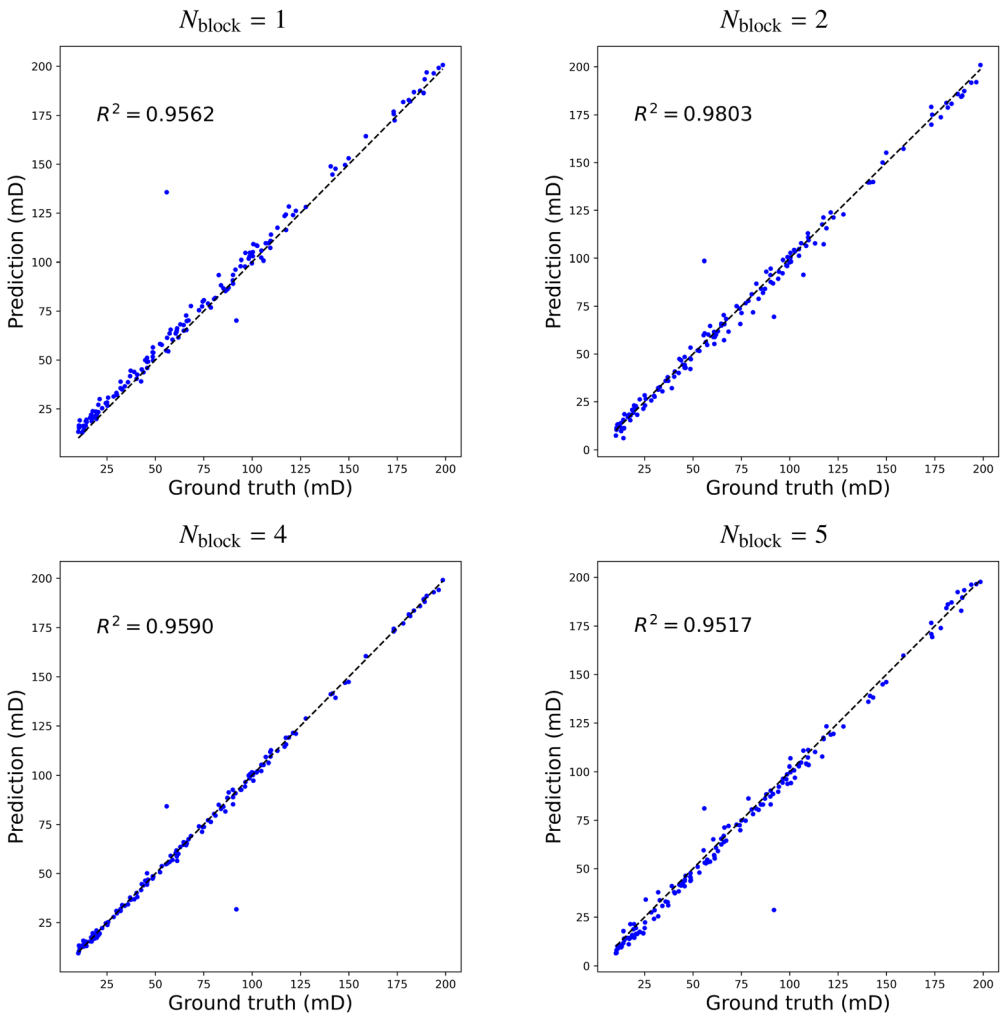

FIG. 6. Performance of Vision Mamba for different numbers of blocks.

achieve the best possible performance, which is referred to as ablation studies. Interpreting the obtained results provides valuable insights into the behavior of the network for the current specific application in this article. For illustration, we focus on three important parameters. The first parameter is the number of Vision Mamba blocks (N_{block}). As explained in Sec. IV, each block can be connected to the subsequent one, thereby progressively deepening the network. The results of this investigation are summarized in Table III and Fig. 6, where the number of Vision Mamba blocks (N_{block}) was varied from 1 to 5. For each configuration, we report the coefficient of determination (i.e., R^2 score), the root mean square error, as well as the maximum and minimum relative errors. As observed from Table III, the best performance is obtained with a network consisting of three blocks (i.e., $N_{\text{block}} = 3$), based on maximizing the R^2 score and minimizing the root mean square error on the test set (170 data). When the number of blocks is reduced, the network becomes shallower and the number of trainable parameters decreases, which leads to a decline in performance. However, this reduction is not particularly severe. For example, when the number of Vision Mamba blocks is reduced from 3 to 2, the R^2 score decreases only slightly from 0.9969 to 0.9803. Even with just a single Vision Mamba block (i.e., $N_{\text{block}} = 1$), the R^2 score remains at 0.9562, indicating that the network maintains reasonable performance. Conversely, increasing the number of blocks from 3 to 4 and 5 leads to R^2 scores of 0.9590 and 0.9517, respectively. Thus, no performance improvement is observed beyond 3 Vision Mamba blocks; instead, a slight reduction in the R^2 score occurs. This decline may be attributed to a slight overfitting on the training data set, as the number of trainable parameters increases.

The next hyperparameter we investigate is the patch size, the concept of which was explained in Sec. III. The outcomes of this investigation are listed in Table IV. Accordingly, we consider five different patch sizes: 4, 8, 16, 32, and 64. Similar to the previous case, we take the R^2 score and the root mean square error as benchmarks. Consequently, the best performance is obtained with a patch size of 16. However, the difference in R^2 scores across the different patch sizes is relatively small, with the highest being 0.9974 for the patch size of 16 and the lowest 0.9805 for the patch size of 64.

As shown in Table IV, the overall performance of the network decreases slightly as the patch size increases beyond 16. This trend can be explained as follows. The patch size determines how each three-dimensional input is divided into smaller cubes and transformed into a sequence of patches, allowing the network to learn both their features and their relationships with neighboring patches. As discussed in Sec. III, these sequences are constructed by scanning the input along three spatial directions. In this sense, when the patch size is 64 and the porous media samples are also of size $64 \times 64 \times 64$ (i.e., $n = 64$), no subdivision occurs; the entire cube is treated as a single patch, and fine-scale details are lost. In contrast, with a patch size of 8, a $64 \times 64 \times 64$ cube (i.e., $n = 64$) is divided into 512 smaller patches, which are then sequentially processed in three spatial directions (e.g., length, width, and height). Hence, this representation enables the network to better capture local features, leading to more accurate permeability predictions in porous media. Additionally, we observe from Table IV that using the patch size of 8 does not improve performance and yields results nearly identical to those with a patch size of 16. It is conjectured that this behavior is related to the spatial correlation length of the dataset, which is 17 voxels (i.e., $\ell_c = 17$). This may suggest that the optimal patch size should be close to the spatial correlation length (if known), since it encapsulates the dominant information embedded at that scale.

As described in Sec. III, the proposed Vision Mamba-based network processes 3D porous-media cubes along the three spatial axes (x, y, and z) and aggregates the resulting features by averaging, as in Eq. (11). The network's output is the permeability in the x-direction [see Sec. II and Eq. (3)]. To test whether scanning along the other two axes helps predict $x-$ direction permeability, we conduct an ablation in which, instead of scanning along all three axes and averaging, we scan exclusively along a single axis (x only, y only, or z only). The results in Table V show that, while x-only scanning is more accurate than y-only or z-only (as expected, given that the target is x-permeability), aggregating features from all three axes yields the best performance, with higher R^2 and lower root mean squared error. This outcome is consistent with the underlying physics, which indicates that the average velocity in Eq. (3) and the permeability in the x-direction depend on the full three-dimensional pore geometry. Therefore, solving the Stokes equations [see Eqs. (1) and (2)] in three dimensions is required even when estimating a directional permeability.

To evaluate the stability of the proposed Vision Mamba framework with respect to random initialization, we repeat the Vision Mamba training procedure five times using different random seeds. The resulting R^2 score is expressed as 0.9943 ± 0.0036, where 0.9943 is the mean and 0.0036 is the standard deviation across the five runs. The small variance observed in these repeated machine learning experiments indicates that the performance of the proposed Vision Mamba model for permeability prediction is robust to initialization.

The final hyperparameter examined is the batch size ($\mathcal{B}$) during training. As shown in Table VI, a batch size of 128 ($\mathcal{B} = 128$) yields the highest R^2 score and the lowest root mean square error when the patch size is fixed at 8 and the number of Vision Mamba blocks is set to 3. Larger batch sizes, such as 256, accelerate training but reduce

TABLE IV. R^2 score, root mean square error, and minimum/maximum relative errors of the test set (170 samples) for different patch sizes in the Vision Mamba model. The number of Vision Mamba blocks is fixed at three ($N_{\text{block}} = 3$).

Patch size	4	8	16	32	64
R^2 score	0.9934	0.9969	0.9974	0.9817	0.9805
Root mean square error (mD)	3.9571	2.6939	2.4557	6.5718	6.7914
Minimum relative error	0.0001	0.0003	0.0001	0.0001	0.0001
Maximum relative error	0.4478	0.2708	0.2105	0.4586	0.8878

TABLE V. Comparison of the R^2 score, root mean square error, and minimum/maximum relative errors on the test set (170 samples) for different scan directions in Vision Mamba [see Eq. (11)]. We set $N_{\text{block}} = 3$ and the patch size to 8.

Scan direction	All three axes	$x-$ axis	$y-$ axis	$z-$ axis
R^2 score	0.9969	0.9945	0.9829	0.9704
Root mean square error (mD)	2.6939	3.5980	6.3660	8.3650
Minimum relative error	0.0003	0.0001	0.0001	0.0001
Maximum relative error	0.2708	0.3503	0.4187	0.9440

TABLE VI. R^2 score, root mean square error, and minimum/maximum relative errors of the test set (170 samples) for different Batch sizes in the Vision Mamba model. The number of Vision Mamba blocks is fixed at three ($N_{block} = 3$). The patch size is set to 8.

Batch size ($\mathcal{B}$)	4	16	32	128	256
R^2 score	0.9750	0.9895	0.9911	0.9969	0.9700
Root mean square error (mD)	7.6883	4.9847	4.5980	2.6939	8.4248
Minimum relative error	0.0004	0.0008	0.0001	0.0003	0.0001
Maximum relative error	0.5114	0.6885	0.3712	0.2708	0.5845

performance, with the R^2 score dropping from 0.9969 to 0.9700, indicating decreased accuracy in predicting porous media permeability.

E. Permeability prediction of natural porous media

In this part of the article, we investigate the capability of the proposed Vision Mamba architecture to be trained on natural porous media and, consequently, to predict the permeability of these rocks. For this purpose, we consider Berea sandstone samples, specifically, a dataset containing 1186 cubes of size $64 \times 64 \times 64$ (i.e., $n = 64$), with permeabilities ranging approximately from 10 mD to 200 mD. We split this dataset into 90% for training, 5% for validation, and 5% for testing. An example of one such sample is shown in the left panel of Fig. 7. The porosity ranges from about 0.08 to 0.19, which differs from the porosity range considered for the synthetic dataset discussed in Secs. V A–V D, and the spatial correlation lengths vary from 5 to 12 voxels. To compute the spatial correlation length, we used the Fast Fourier Transform-based two-point correlation method described in Ref. 71. Unlike the synthetic data generation process, the spatial correlation length in this dataset is not fixed. The outcome of this investigation is presented in the right panel of Fig. 7, where we observe an R^2 score of 0.9833. The root mean squared error is 6.7418 mD. Hence, this experiment demonstrates that the Vision Mamba framework can be effectively applied to natural porous media for real digital rocks and realistic applications, where the dataset is complex and exhibits a range of spatial correlation lengths and porosities.

VI. SUMMARY AND FUTURE RESEARCH PROJECTS

In this article, we presented a neural network based on Vision Mamba for predicting the permeability of three-dimensional porous media. We demonstrated the effectiveness of the proposed model using evaluation criteria such as the coefficient of determination, mean square error, and maximum and minimum relative errors. We discussed the advantages of Vision Mamba compared to CNNs and ViTs for permeability prediction. In particular, we showed that, relative to CNNs, Vision Mamba requires far fewer trainable parameters while achieving superior performance. Furthermore, we demonstrated that GPU memory usage in Vision Mamba scales linearly with patch size, whereas in ViTs it scales quadratically. As a result, under limited GPU memory, Vision Mamba was able to successfully execute the machine learning experiments with small patch sizes, whereas ViTs could not be trained due to insufficient memory when using the same small batch sizes. Finally, we explored the impact of key hyperparameters, including the number of Vision Mamba blocks, patch size within each block, and batch size, to highlight their influence on model performance.

In the current article, we focus on predicting the permeability in the x-direction, denoted by k, which corresponds to the k_{xx} component of the anisotropic permeability tensor. Extending Vision Mamba to predict the full anisotropic permeability tensor is conceptually straightforward. The primary requirement is the availability of training labels for all tensor components, which can be generated by solving the continuity and Stokes equations [see Eqs. (1) and (2)] under three independent pressure-gradient directions (e.g., x, y, and z). The proposed Vision Mamba model can then be adapted by employing a single multi-output regression head that predicts all tensor entries simultaneously. The Vision Mamba backbone remains unchanged and only the output layer and the loss function become multi-dimensional. From a data-generation perspective, computing accurate off diagonal components and ensuring sufficiently diverse training samples to capture a broad range of anisotropy are required.

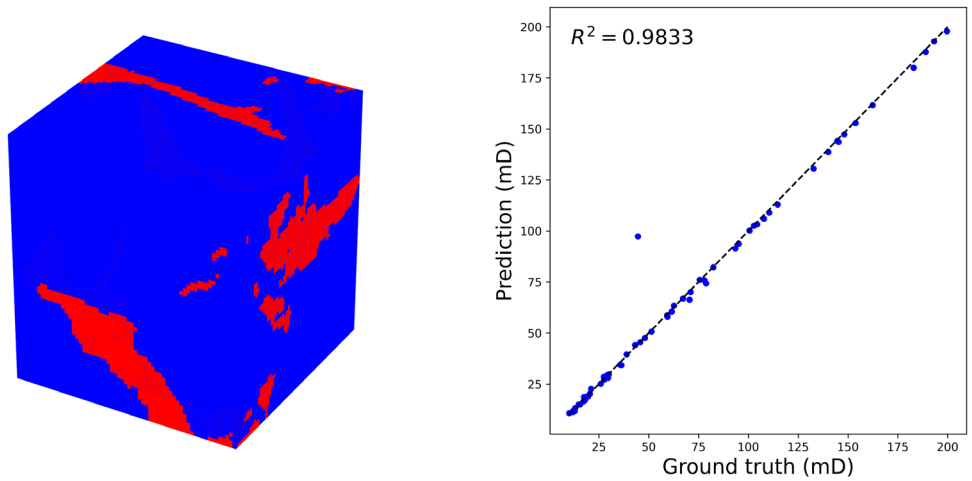

FIG. 7. A Berea sandstone sample as a natural porous medium (left), and the performance of Vision Mamba in predicting its permeability (right).

In the present article, we used the classification branch of Vision Mamba, where the neural network input represents the geometry of the porous medium and the output is the permeability as a scalar value. As one idea for future projects, one could use the segmentation branch of Vision Mamba such that, although the input remains the same three-dimensional cube describing the porous-medium geometry, the output is the predicted velocity field within the pore space. Of course, after obtaining the velocity field, the permeability can also be computed. However, access to the full velocity field provides more information. This can be done in the form of fully supervised deep learning or in the form of weakly supervised deep learning, if only sparse observations from the velocity field are available, by enforcing the governing equations [e.g., Eqs. (1) and (2)] as a loss function for the Vision Mamba network.

Another promising research direction could be the development of large language and vision models for porous media based on Vision Mamba rather than transformer architectures. Such foundation models could handle variable-size and multimineral porous media, enabling the prediction of physical and geometrical features, and offering interactive environments to integrate images, codes, texts, and mathematical formulations within a unified framework (see, e.g., Ref. 72).

ACKNOWLEDGMENTS

The authors of this research article gratefully acknowledge the sponsors of the Stanford Center for Earth Resources Forecasting (SCERF). The authors thank the reviewers for their helpful comments and suggestions, which have contributed to improving the quality of this work.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Ali Kashefi: Conceptualization (lead); Data curation (lead); Formal analysis (lead); Investigation (lead); Methodology (lead); Software (lead); Validation (lead); Visualization (lead); Writing – original draft (lead); Writing – review & editing (equal). **Tapan Mukerji:** Conceptualization (supporting); Formal analysis (supporting); Funding acquisition (lead); Methodology (supporting); Resources (lead); Writing – review & editing (equal).

DATA AVAILABILITY

The data that support the findings of this study are openly available in GitHub at https://github.com/Ali-Stanford/Vision_Mamba_3D_Porous_Media, Ref. 73.

REFERENCES

[1]H. Andra, N. Combaret, J. Dvorkin *et al.*, "Digital rock physics benchmarks—Part I: Imaging and segmentation," Comput. Geosci. **50**, 25–32 (2013).
[2]H. Andra, N. Combaret, J. Dvorkin *et al.*, "Digital rock physics benchmarks—Part II: Computing effective properties," Comput. Geosci. **50**, 33–43 (2013).
[3]J. Zhu, L. Zhao, W. Zhu, and J. Geng, "Joint use of multiscale digital rock physics and effective medium theory to model elastic properties of shale reservoir," Geophysics **90**, MR307–MR318 (2025).
[4]Y. Liang and D. Fletcher, "Computational fluid dynamics simulation of forward osmosis (FO) membrane systems: Methodology, state of art, challenges and opportunities," Desalination **549**, 116359 (2023).
[5]F. S. Ferro and B. S. Carmo, "Numerical modelling and simulation of hollow fiber dense membranes for CO_2/CH_4 separation using CFD," J. Membr. Sci. **729**, 124134 (2025).
[6]M. J. Blunt, B. Bijeljic, H. Dong *et al.*, "Pore-scale imaging and modelling," Adv. Water Resour. **51**, 197–216 (2013).
[7]M. Yang, S. Huang, F. Zhao, and C. Yang, "A novel hybrid finite-infinite diffusion model for determining CO_2 diffusion coefficient in oil-saturated porous media: Applications for enhanced oil recovery and geological carbon storage," Energy **316**, 134621 (2025).
[8]T. Kumeria, "Advances on porous nanomaterials for biomedical application (drug delivery, sensing, and tissue engineering)," ACS Biomater. Sci. Eng. **8**, 4025–4027 (2022).
[9]M. K. Das, P. P. Mukherjee, and K. Muralidhar, *Porous Media Applications: Biological Systems* (Springer International Publishing, 2018).
[10]U. Farooq, T. Liu, and A. Jan, "Boundary layer analysis of second-order magnetic nanofluid flow with carbon nanotubes and gyrotactic microorganisms for medical diagnostics," BioNanoScience **15**, 113 (2025).
[11]A. Bihani, H. Daigle, J. E. Santos *et al.*, "MudrockNet: Semantic segmentation of mudrock SEM images through deep learning," Comput. Geosci. **158**, 104952 (2022).
[12]H.-B. Lee, M.-H. Jung, Y.-H. Kim *et al.*, "Deep learning image segmentation for the reliable porosity measurement of high-capacity Ni-based oxide cathode secondary particles," J. Anal. Sci. Technol. **14**, 47 (2023).
[13]Y. Han and Y. Liu, "Advanced petrographic thin section segmentation through deep learning-integrated adaptive GLFIF," Comput. Geosci. **193**, 105713 (2024).
[14]C. Wang, H. Luo, J. Wang, and D. Groom, "ReUNet: Efficient deep learning for precise ore segmentation in mineral processing," Comput. Geosci. **195**, 105773 (2025).
[15]Y. Meng, J. Jiang, J. Wu, and D. Wang, "Transformer-based deep learning models for predicting permeability of porous media," Adv. Water Resour. **179**, 104520 (2023).
[16]C. Xie, J. Zhu, H. Yang *et al.*, "Relative permeability curve prediction from digital rocks with variable sizes using deep learning," Phys. Fluids **35**, 096605 (2023).
[17]A. Kashefi and T. Mukerji, "Point-cloud deep learning of porous media for permeability prediction," Phys. Fluids **33**, 097109 (2021).
[18]M. Liu, R. Ahmad, W. Cai, and T. Mukerji, "Hierarchical homogenization with deep-learning-based surrogate model for rapid estimation of effective permeability from digital rocks," J. Geophys. Res.: Solid Earth **128**, e2022JB025378, https://doi.org/10.1029/2022JB025378 (2023).
[19]J. Hong and J. Liu, "Rapid estimation of permeability from digital rock using 3D convolutional neural network," Comput. Geosci. **24**, 1523–1539 (2020).
[20]J. Wu, X. Yin, and H. Xiao, "Seeing permeability from images: Fast prediction with convolutional neural networks," Sci. Bull. **63**, 1215–1222 (2018).
[21]M. Masroor, M. Emami Niri, and M. H. Sharifinasab, "A multiple-input deep residual convolutional neural network for reservoir permeability prediction," Geoenergy Sci. Eng. **222**, 211420 (2023).
[22]H. Sun, L. Zhou, D. Fan *et al.*, "Permeability prediction of considering organic matter distribution based on deep learning," Phys. Fluids **35**, 032014 (2023).
[23]A. Kashefi and T. Mukerji, "A novel Fourier neural operator framework for classification of multi-sized images: Application to three dimensional digital porous media," Phys. Fluids **36**, 057131 (2024).
[24]K. M. Graczyk and M. Matyka, "Predicting porosity, permeability, and tortuosity of porous media from images by deep learning," Sci. Rep. **10**, 21488 (2020).
[25]J. Chung, R. Ahmad, W. Sun *et al.*, "Prediction of effective elastic moduli of rocks using graph neural networks," Comput. Methods Appl. Mech. Eng. **421**, 116780 (2024).
[26]C. Liu, R. Guo, and Y. Su, "A deep learning based prediction model for effective elastic properties of porous materials," Sci. Rep. **15**, 6707 (2025).
[27]H. Wu, W.-Z. Fang, Q. Kang *et al.*, "Predicting effective diffusivity of porous media from images by deep learning," Sci. Rep. **9**, 20387 (2019).
[28]J. E. Santos, D. Xu, H. Jo *et al.*, "PoreFlow-Net: A 3D convolutional neural network to predict fluid flow through porous media," Adv. Water Resour. **138**, 103539 (2020).

[29]S. Kamrava, M. Sahimi, and P. Tahmasebi, "Simulating fluid flow in complex porous materials by integrating the governing equations with deep-layered machines," npj Comput. Mater. **7**, 127 (2021).
[30]A. Kashefi and T. Mukerji, "Prediction of fluid flow in porous media by sparse observations and physics-informed PointNet," Neural Networks **167**, 80–91 (2023).
[31]M. Liu and T. Mukerji, "Multiscale fusion of digital rock images based on deep generative adversarial networks," Geophys. Res. Lett. **49**, e2022GL098342, https://doi.org/10.1029/2022GL098342 (2022).
[32]K. M. Guan, T. I. Anderson, P. Creux, and A. R. Kovscek, "Reconstructing porous media using generative flow networks," Comput. Geosci. **156**, 104905 (2021).
[33]J. Phan, M. Sarmad, L. Ruspini *et al.*, "Generating 3D images of material microstructures from a single 2D image: A denoising diffusion approach," Sci. Rep. **14**, 6498 (2024).
[34]N. Baishnab, E. Herron, A. Balu *et al.*, "3D multiphase heterogeneous microstructure generation using conditional latent diffusion models," arXiv:2503.10711 (2025).
[35]K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'16)* (IEEE, 2016), pp. 770–778.
[36]P. Tang, D. Zhang, and H. Li, "Predicting permeability from 3D rock images based on CNN with physical information," J. Hydrol. **606**, 127473 (2022).
[37]C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2017), pp. 652–660.
[38]A. Kashefi, "PointNet with KAN versus PointNet with MLP for 3D classification and segmentation of point sets," Comput. Graphics **131**, 104319 (2025).
[39]C. R. Qi, L. Yi, H. Su, L. J. Guibas "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in Neural Information Processing Systems*, edited by I. Guyon, U. V. Luxburg, S. Bengio *et al.* (Curran Associates, Inc., 2017), Vol. 30.
[40]Z. Li, N. Kovachki, K. Azizzadenesheli *et al.*, "Fourier neural operator for parametric partial differential equations," arXiv:2010.08895 (2020).
[41]A. Dosovitskiy, L. Beyer, A. Kolesnikov *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," arXiv:2010.11929 (2021).
[42]S. Geng, S. Zhai, and C. Li, "Swin transformer based transfer learning model for predicting porous media permeability from 2D images," Comput. Geotech. **168**, 106177 (2024).
[43]C. Temizel, U. Odi, K. Li *et al.*, "Permeability prediction using vision transformers," Math. Comput. Appl. **30**, 71 (2025).
[44]A. Gu and T. Dao, "Mamba: Linear-time sequence modeling with selective state spaces," arXiv:2312.00752v2 (2024).
[45]A. Vaswani, N. Shazeer, N. Parmar *et al.*, "Attention is all you need," arXiv:1706.03762 (2023).
[46]L. Zhu, B. Liao, Q. Zhang *et al.*, "Vision Mamba: Efficient visual representation learning with bidirectional state space model," arXiv:2401.09417 (2024).
[47]R. Xu, S. Yang, Y. Wang *et al.*, "A survey on Vision Mamba: Models, applications and challenges," CoRR abs/2404.18861 (2024).
[48]Y. Liu, Y. Tian, Y. Zhao *et al.*, "VMamba: Visual state space model," arXiv:2401.10166 (2024).
[49]Y. Yue and Z. Li, "MedMamba: Vision Mamba for medical image classification," arXiv:2403.03849 (2024).
[50]M. Bao, S. Lyu, Z. Xu *et al.*, "Vision Mamba in remote sensing: A comprehensive survey of techniques, applications and outlook," arXiv:2505.00630 (2025).
[51]J. Liu, J. Li, X. Wang *et al.*, "Mamba-augmented residual network for rapid wake field prediction of underwater vehicles," Ocean Eng. **341**, 122474 (2025).
[52]C. Wang, Y. Xie, Q. Chen *et al.*, "A comprehensive analysis of Mamba for 3D volumetric medical image segmentation," arXiv:2503.19308 (2025).
[53]Y. Yang, Z. Xing, L. Yu *et al.*, "Vivim: A video Vision Mamba for medical video segmentation," arXiv:2401.14168 (2024).
[54]OpenAI, J. Achiam, S. Adler *et al.*, "GPT-4 technical report," arXiv:2303.08774 (2024).
[55]Y. Chang, X. Wang, J. Wang *et al.*, "A survey on evaluation of large language models," arXiv:2307.03109 (2023).
[56]G. Team, P. Georgiev, V. I. Lei *et al.*, "Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context," arXiv:2403.05530 (2024).
[57]A. Kashefi and T. Mukerji, "ChatGPT for programming numerical methods," J. Mach. Learn. Model. Comput. **4**, 1–74 (2023).
[58]A. Kashefi, "A misleading gallery of fluid motion by generative artificial intelligence," J. Mach. Learn. Model. Comput. **5**, 113–144 (2024).
[59]A. Basant, A. Khairnar, A. Paithankar *et al.*, "NVIDIA Nemotron Nano 2: An accurate and efficient hybrid Mamba-transformer reasoning model," arXiv:2508.14444 (2025).
[60]J. Xu, H. Wei, and H. Bao, "Physics-informed neural networks for studying heat transfer in porous media," Int. J. Heat Mass Transfer **217**, 124671 (2023).
[61]M. Sahimi, "Physics-informed and data-driven discovery of governing equations for complex phenomena in heterogeneous media," Phys. Rev. E **109**, 041001 (2024).
[62]C. Lantuejoul, *Geostatistical Simulation: Models and Algorithms* (Springer, 2002).
[63]M. Le Ravalec-Dupin, F. Roggero, and R. Froidevaux, "Conditioning truncated gaussian realizations to static and dynamic data," SPE J. **9**, 475–480 (2004).
[64]Y. Keehm, T. Mukerji, and A. Nur, "Permeability prediction from thin sections: 3D reconstruction and lattice-Boltzmann flow simulation," Geophys. Res. Lett. **31**, L04606, https://doi.org/10.1029/2003GL018761 (2004).
[65]H. Darcy, *Les Fontaines Publiques de la Ville de Dijon: Exposition et Application Des Principes à Suivre et Des Formules à Employer Dans Les Questions de Distribution D'eau* (Victor Dalmont, 1856), Vol. 1.
[66]A. Gu and T. Dao, "Mamba: Linear-time sequence modeling with selective state spaces," arXiv:2312.00752 (2023).
[67]D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv:1412.6980 (2014).
[68]I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, 2016). http://www.deeplearningbook.org.
[69]S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning* (PMLR, 2015), pp. 448–456.
[70]N. Srivastava, G. Hinton, A. Krizhevsky *et al.*, "Dropout: A simple way to prevent neural networks from overfitting," J. Mach. Learn. Res. **15**, 1929–1958 (2014).
[71]S. Torquato *et al.*, *Random Heterogeneous Materials: Microstructure and Macroscopic Properties* (Springer, 2002), Vol. 16.
[72]Y. Qiao, Z. Yu, L. Guo *et al.*, "VL-Mamba: Exploring state space models for multimodal learning," arXiv:2403.13600 (2024).
[73]A. Kashefi and T. Mukerji (2025). "Vision Mamba for permeability prediction of porous media," GitHub. https://github.com/Ali-Stanford/Vision_Mamba_3D_Porous_Media