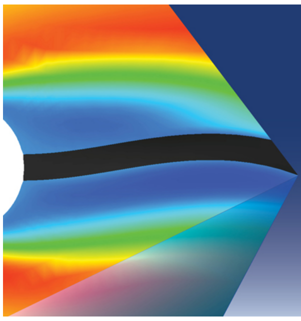
Physics of Fluids

RESEARCH ARTICLE | MAY 15 2024

A novel Fourier neural operator framework for classification of multi-sized images: Application to three dimensional digital porous media

Ali Kashefi ; Tapan Mukerji

Physics of Fluids 36, 057131 (2024)
https://doi.org/10.1063/5.0203977



AIP Publishing

A novel Fourier neural operator framework for classification of multi-sized images: Application to three dimensional digital porous media

Cite as: Phys. Fluids **36**, 057131 (2024); doi: 10.1063/5.0203977
Submitted: 18 February 2024 · Accepted: 30 April 2024 ·
Published Online: 15 May 2024

Ali Kashefi[1,a] and Tapan Mukerji[2,b]

AFFILIATIONS

[1]Department of Civil and Environmental Engineering, Stanford University, Stanford, California 94305, USA
[2]Department of Energy Science and Engineering, Stanford University, Stanford, California 94305, USA

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

ABSTRACT

Fourier neural operators (FNOs) are invariant with respect to the size of input images, and thus images with any size can be fed into FNO-based frameworks without any modification of network architectures, in contrast to traditional convolutional neural networks. Leveraging the advantage of FNOs, we propose a novel deep-learning framework for classifying images with varying sizes. Particularly, we simultaneously train the proposed network on multi-sized images. As a practical application, we consider the problem of predicting the label (e.g., permeability) of three-dimensional digital porous media. To construct the framework, an intuitive approach is to connect FNO layers to a classifier using adaptive max pooling. First, we show that this approach is only effective for porous media with fixed sizes, whereas it fails for porous media of varying sizes. To overcome this limitation, we introduce our approach: instead of using adaptive max pooling, we use static max pooling with the size of channel width of FNO layers. Since the channel width of the FNO layers is independent of the input image size, the introduced framework can handle multi-sized images during training. We show the effectiveness of the introduced framework and compare its performance with the intuitive approach through the example of the classification of three-dimensional digital porous media of varying sizes.

I. INTRODUCTION AND MOTIVATION

Since 2020, neural operators have gained extensive popularity, specifically with two versions of graph neural operators (Li *et al.*, 2020b; 2020c) and Fourier neural operators (FNOs) (Li *et al.*, 2020a; Kovachki *et al.*, 2024). In this article, our attention is on FNOs. From a computer science perspective, regular FNOs fall in the category of supervised deep learning framework, necessitating a large volume of labeled data for training. FNOs have demonstrated their proficiency in input–output mapping across various industrial and scientific applications such as incompressible flows (Li *et al.*, 2022b; Bonev *et al.*, 2023; Peng *et al.*, 2024; Choubineh *et al.*, 2023; Lyu *et al.*, 2023; Gupta and Brandstetter, 2022; and Peng *et al.*, 2023a), wave equations (Zhu *et al.*, 2023a; Zou *et al.*, 2023; and Yang *et al.*, 2023), thermal fields (Zhao *et al.*, 2024; Hao *et al.*, 2023), carbon storages and sequestration (Wen *et al.*, 2022; Jiang *et al.*, 2023b), and other areas (Peng *et al.*, 2023b; You *et al.*, 2022; Kontolati *et al.*, 2023; Zhu *et al.*, 2023b; Hua and Lu, 2023; White *et al.*, 2023a; Li *et al.*, 2021; Pathak *et al.*, 2022; Rahman *et al.*, 2022b; 2022a; Yang *et al.*, 2022; Li *et al.*, 2022a; Maust *et al.*, 2022; Zhao *et al.*, 2022; Renn *et al.*, 2023; Xiong *et al.*, 2023; Chen *et al.*, 2023; Huang *et al.*, 2023; Poels *et al.*, 2023; White *et al.*, 2023b; Thodi *et al.*, 2023; Zhao *et al.*, 2023; Tran *et al.*, 2023; Lee, 2022; Brandstetter *et al.*, 2023; Li *et al.*, 2023; Majumdar *et al.*, 2023; Jiang *et al.*, 2023a; Lehmann *et al.*, 2024; Subramanian *et al.*, 2024; Fanaskov and Oseledets, 2024; Lanthaler, 2021; and Azzizadenesheli *et al.*, 2023). From a computer vision perspective, these are framed as segmentation problems, where an input image, such as the geometry of an airfoil, is mapped to another image, for instance, the velocity field around that airfoil. An analogous area in computer vision is classification, where an input image is mapped, for example, to a name or number. While FNOs have potential in classification tasks, there exists only a limited amount of research conducted in this application as per our knowledge (Johnny *et al.*, 2022; Xi *et al.*, 2022; and Kabri *et al.*, 2023).

Johnny *et al.* (2022) used the FNO architecture for classifying images in the CIFAR-10 dataset, containing ten different classes; however, they trained the network only on images with a fixed size of 32×32 pixels. Additionally, Kabri *et al.* (2023) examined the FNO architecture for image classification. Although they tested images of various sizes (e.g., 28 × 28 pixels, 112 × 112 pixels, etc.), they trained and then tested the network separately for each size, assessing its performance on the corresponding size. Xi *et al.* (2022) utilized the FNO architecture for the hyperspectral remote sensing image classification. Their dataset comprised images of various sizes, including 512 × 614 pixels, 610 × 340 pixels, and 512 × 217 pixels. However, they adjusted all images to a fixed size by adding patches. Consequently, although they employed the FNO architecture, in practice, they limited their analysis to images of a uniform size. In the current study, we narrow our focus on classification problems. More specifically, we consider the problem of predicting the permeability of three-dimensional digital porous media, which vary in size, as a benchmark test case.

FNOs are invariant with respect to the size of input images, and this characteristic ensures that images of varying sizes can be processed by FNO-based deep learning frameworks without requiring any architectural alterations. Note that regular convolutional neural networks (CNNs) lack this feature (Goodfellow *et al.*, 2016). Building on this strength of FNOs, we introduce a deep learning framework for training the network simultaneously on images with varying sizes for a classification problem. To achieve this deep learning framework, FNO layers must be connected to a classifier, which is commonly a multilayer perceptron (MLP). An intuitive approach to set this would be to link FNO layers to a classifier via adaptive max pooling. Considering the application of permeability prediction of three-dimensional porous media, our machine-learning experiments show that this intuitive approach only works well for porous media with fixed sizes. Pivoting from this, we propose our novel approach. Rather than using adaptive max pooling, we implement static max pooling with the size of the channel width of the FNO layers. Given that the size of the channel width of FNO layers is independent of the size of input images, our proposed framework can be efficiently trained on various image sizes at once (see Figs. 1 and 2).

To explain, at a high level, the difference between using adaptive max pooling (see Fig. 2) and static max pooling (see Fig. 1), let us consider for example a three-dimensional image being fed as an input of the deep learning framework. For both pooling methods, at the framework's outset, FNO layers lift the input image from its three-dimensional space to a higher dimensional space, determined by the size of the channel width of the FNO layers. In the case of adaptive max pooling, after FNO layer operations, the outcome eventually is dropped into the initial three-dimensional space with the same size as the input image. This array then serves as the input of adaptive max pooling. The output of the adaptive pooling is then the input of the classifier. In the case of static max pooling, before FNO layers drop the output, we implement static max pooling, which functions within the high dimensional space and pools with the size of the channel width of FNO layers. The resulting output from this pooling then becomes the classifier's input. A more detailed exploration of these concepts is provided in Sec. II.

The study of physical and geometric features of porous media is important in diverse scientific and industrial areas such as digital rock physics (Andra *et al.*, 2013a; 2013b), membrane systems (Liang and Fletcher, 2023), geological carbon storages (Blunt *et al.*, 2013), and medicine (Kumeria, 2022; Das *et al.*, 2018). Deep learning frameworks have been widely used for predicting the permeability of porous media (Meng *et al.*, 2023; Xie *et al.*, 2023; Kashefi and Mukerji, 2023; 2021; Liu *et al.*, 2023; Hong and Liu, 2020; Wu *et al.*, 2018; Tembely *et al.*, 2020; Masroor *et al.*, 2023; and Sun *et al.*, 2023), but, to the best of our knowledge, all these frameworks were trained on a fixed-size porous media. Note that training the proposed network to predict the permeability of porous media of varying sizes comes with an exclusive

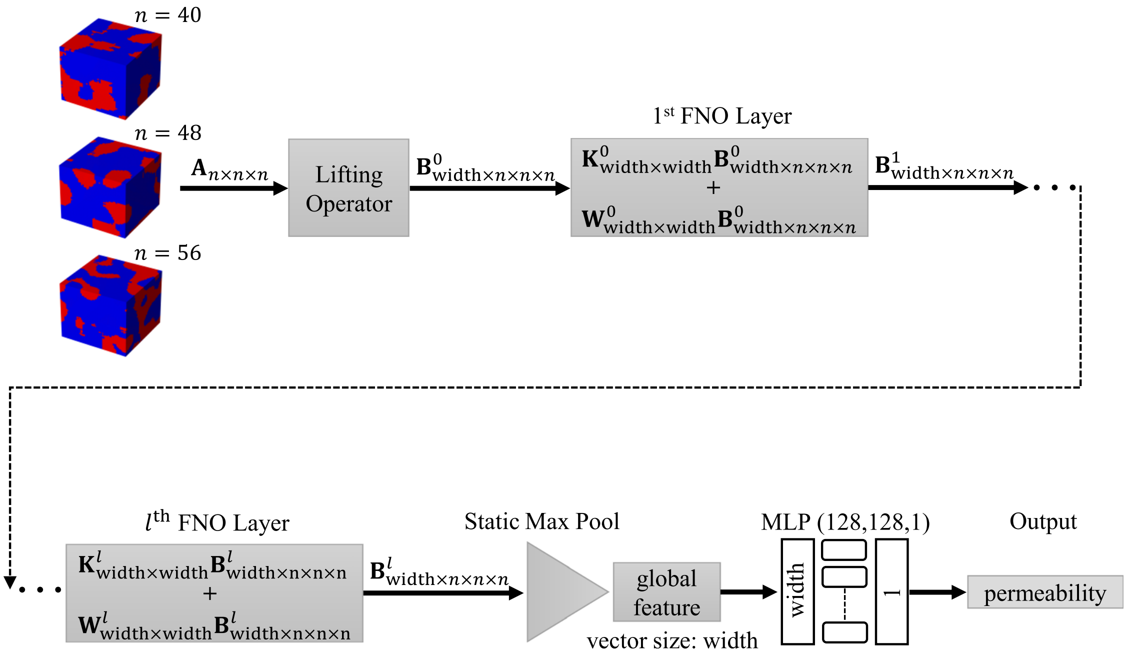

$n = 40$
$n = 48$
$n = 56$
$\mathbf{A}_{n\times n\times n}$
Lifting Operator
$\mathbf{B}^0_{\text{width}\times n\times n\times n}$
1st FNO Layer
$\mathbf{K}^0_{\text{width}\times\text{width}}\mathbf{B}^0_{\text{width}\times n\times n\times n}$
+
$\mathbf{W}^0_{\text{width}\times\text{width}}\mathbf{B}^0_{\text{width}\times n\times n\times n}$
$\mathbf{B}^1_{\text{width}\times n\times n\times n}$
l^{th} FNO Layer
$\mathbf{K}^l_{\text{width}\times\text{width}}\mathbf{B}^l_{\text{width}\times n\times n\times n}$
+
$\mathbf{W}^l_{\text{width}\times\text{width}}\mathbf{B}^l_{\text{width}\times n\times n\times n}$
$\mathbf{B}^l_{\text{width}\times n\times n\times n}$
Static Max Pool
global feature
vector size: width
MLP (128,128,1)
width
1
Output
permeability

FIG. 1. Schematic of the proposed FNO-based framework for multi-size image classification.

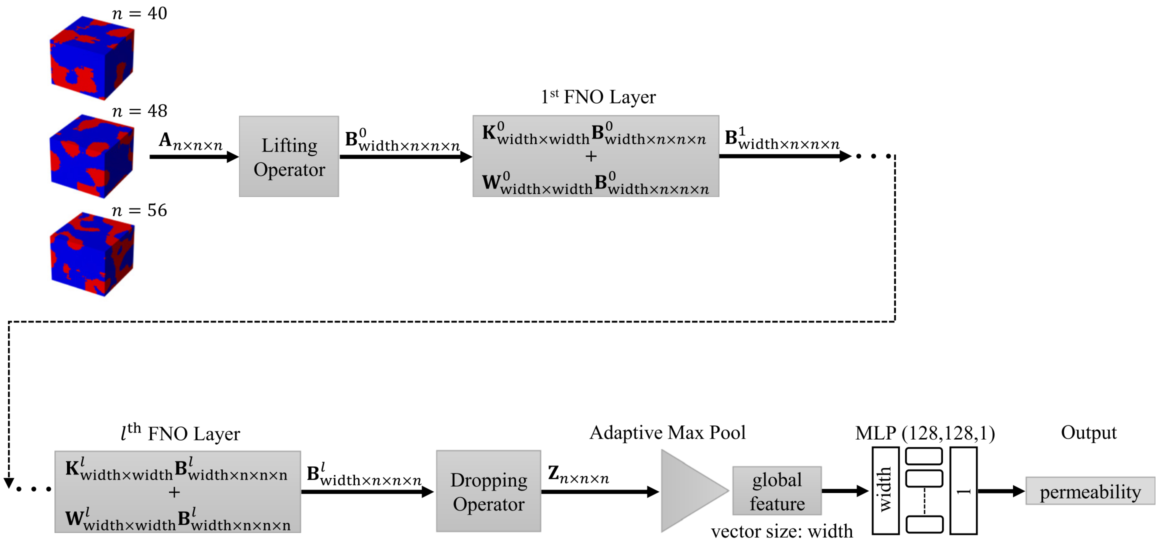

n = 40
n = 48
n = 56
$\mathbf{A}_{n\times n\times n}$
Lifting Operator
$\mathbf{B}^0_{\text{width}\times n\times n\times n}$
1st FNO Layer
$\mathbf{K}^0_{\text{width}\times\text{width}}\mathbf{B}^0_{\text{width}\times n\times n\times n}$ + $\mathbf{W}^0_{\text{width}\times\text{width}}\mathbf{B}^0_{\text{width}\times n\times n\times n}$
$\mathbf{B}^1_{\text{width}\times n\times n\times n}$
lth FNO Layer
$\mathbf{K}^l_{\text{width}\times\text{width}}\mathbf{B}^l_{\text{width}\times n\times n\times n}$ + $\mathbf{W}^l_{\text{width}\times\text{width}}\mathbf{B}^l_{\text{width}\times n\times n\times n}$
$\mathbf{B}^l_{\text{width}\times n\times n\times n}$
Dropping Operator
$\mathbf{Z}_{n\times n\times n}$
Adaptive Max Pool
global feature
vector size: width
MLP (128,128,1)
width
1
Output
permeability

FIG. 2. Schematic of the intuitive FNO-based framework for multi-size image classification.

challenge when compared to training the network on conventional images for the purpose of classifying them by their names (like those of cats and dogs). For conventional images, one possible solution to handle images with different sizes is to equalize them by adding mini patches to the smaller images. Nevertheless, this solution is inapplicable to the porous media problem. Adding mini patches to porous media can alter their physical properties such as permeability. For instance, adding mini patches around a porous medium simulates sealing it with wall boundaries, which prohibits flow within its pore spaces, resulting in a permeability of zero. Additionally, the inherently three-dimensional nature of porous media introduces another layer of complexity compared to the two-dimensional conventional images. We summarize the contributions of our work in the following bullet points:

- We propose a novel deep learning framework for image classification.
- The proposed framework leverages Fourier neural operators, which are invariant to the size of input images.
- Specifically designed to train simultaneously on images of multiple sizes, the framework can effectively classify images of varying sizes.
- This is an important feature for applications where input images naturally vary in size. We demonstrate its application specifically for three-dimensional images.

The remainder of this article is organized as follows. We introduce and discuss the concept of Fourier neural operators for image classification in Sec. II, starting with the traditional strategy of adaptive max pooling, followed by our novel approach of static max pooling in the high dimension of the Fourier space channel. A brief review of theoretical aspects of FNOs is given in Sec. II C. Data generation and the training methodologies are, respectively, presented in Secs. III and IV. In Sec. V, we provide results and discussion, including a comparison between traditional strategy and our novel approach. Moreover, we present a sensitivity analysis, covering the number of Fourier modes, the channel width of discrete Fourier space, the number of FNO units, and the effect of activation functions and average pooling. The deep learning model generalizability is discussed in this section as well. Finally, we summarize the work and present insight into future directions in Sec. VI.

II. FOURIER NEURAL OPERATORS FOR IMAGE CLASSIFICATION

A. Our novel approach: Static max pooling in channel width of FNO layers

In this subsection, we introduce the architecture of our proposed deep learning framework. Our explanation heavily uses matrix notation to ensure clarity and provide a deeper understanding. As illustrated in Fig. 1, the input of the deep learning framework is a cubic binary porous medium represented as the matrix $\mathbf{A}_{n\times n\times n}$. As a first step, the matrix $\mathbf{A}_{n\times n\times n}$ is lifted to a higher dimensional space using a fully connected network. The dimension of this space is termed the channel width of an FNO layer, shown by "width" in our matrix notation. This lifting results in a four-dimensional matrix, denoted as $\mathbf{B}^0_{\text{width}\times n\times n\times n}$. The matrix $\mathbf{B}^0_{\text{width}\times n\times n\times n}$ becomes subsequently the input of an FNO layer. Within the FNO layer, two operations are applied to $\mathbf{B}^0_{\text{width}\times n\times n\times n}$: the kernel integration operator, denoted by $\mathbf{K}^0_{\text{width}\times\text{width}}$, and the linear transformation operator, denoted by $\mathbf{W}^0_{\text{width}\times\text{width}}$. The network computes the matrix–matrix multiplication of $\mathbf{K}^0_{\text{width}\times\text{width}}\mathbf{B}^0_{\text{width}\times n\times n\times n}$ and $\mathbf{W}^0_{\text{width}\times\text{width}}\mathbf{B}^0_{\text{width}\times n\times n\times n}$ and then sums up the resulting matrices, as depicted in Fig. 1. The output undergoes element-wise operations of the Rectified Linear Unit (ReLU) activation function (Goodfellow *et al.*, 2016) defined as

$$\sigma(\gamma) = \max(0, \gamma). \tag{1}$$

Resulting in a four-dimensional matrix $\mathbf{B}^1_{\text{width}\times n\times n\times n}$. Mathematically, this procedure can be summarized as

$$\mathbf{B}^1_{\text{width}\times n\times n\times n} = \sigma\big(\mathbf{K}^0_{\text{width}\times\text{width}}\mathbf{B}^0_{\text{width}\times n\times n\times n} + \mathbf{W}^0_{\text{width}\times\text{width}}\mathbf{B}^0_{\text{width}\times n\times n\times n}\big). \tag{2}$$

In scenarios where multiple FNO layers exist in the framework, the matrix $\mathbf{B}^{1}_{\text{width}\times n\times n\times n}$ serves as the input for the succeeding FNO layers, and the same sequence of operations is applied. If we assume that there are l number of FNO layers, the output from the final FNO layer is the matrix $\mathbf{B}^{l}_{\text{width}\times n\times n\times n}$. In the next step, we implement static max pooling on the first dimension of matrix $\mathbf{B}^{l}_{\text{width}\times n\times n\times n}$. Because width is independent of the input image dimension (i.e., n), the static pooling works for input images with any desired size (e.g., $n = 40$, $n = 48$, and $n = 56$). Note that the width is a hyper parameter of FNO layers and independent of n, as all the matrix–matrix multiplication operates on the dimension with the size width, and not "n." The static max pooling produces a vector of length width, representing the global features of the input images. The vector is then connected to a classifier. The classifier is a Multilayer Perceptron (MLP) composed of three layers of sizes 128, 128, and 1. The ReLU activation function is used in the initial two layers along with a dropout with a rate of 0.3. Following the third layer, a sigmoid activation function defined as

$$\sigma(\gamma) = \frac{1}{1 + e^{-\gamma}}, \tag{3}$$

is used to ensure output values are bounded between 0 and 1.

B. Intuitive approach: Adaptive max pooling in 3 D spatial space

In this subsection, we explain the intuitive approach (see Fig. 2). Drawing parallels to our approach elaborated in Subsection II A, we begin by considering the input porous medium, which is a three-dimensional matrix represented by $\mathbf{A}_{n\times n\times n}$. All operations outlined in Sec. II A are applied to $\mathbf{A}_{n\times n\times n}$ until the network obtains the matrix $\mathbf{B}^{l}_{\text{width}\times n\times n\times n}$ at an intermediate step, as depicted in Fig. 2. As the next step, we drop (as an inverse of the lifting operator explained in Sec. II A) the matrix $\mathbf{B}^{l}_{\text{width}\times n\times n\times n}$ from the high dimensional space to the default space by means of a fully connected network. This transformation results in the matrix $\mathbf{Z}_{n\times n\times n}$. At this juncture, we use the adaptive three-dimensional max pooling, a functionality that is available in deep learning platforms such as PyTorch (Paszke *et al.*, 2019) or TensorFlow (Abadi *et al.*, 2015). To ensure a fair comparison between the traditional approach and our novel approach, we keep the size of the vector of the global feature consistent across both approaches. To this end, the output of the adaptive max pooling is tailored to yield a vector of size width. The resulting vector represents the global features of the input images.

Note that because the size of matrix $\mathbf{Z}_{n\times n\times n}$ depends on the size of the input image (i.e., n), the pooling must be adaptive as we plan to train the network simultaneously on input images with varying sizes (e.g., $\mathbf{A}_{40\times 40\times 40}$, $\mathbf{A}_{48\times 48\times 48}$, and $\mathbf{A}_{56\times 56\times 56}$). Subsequent to the adaptive max pooling, the global feature vector is connected to a classifier. This classifier features and architecture are precisely the same as the one elucidated in Sec. II A.

To close this subsection, it is noted that the main difference between static max pooling and adaptive max pooling can be articulated as follows. In static max pooling, the kernel size and stride are constant, whereas in adaptive max pooling, they are not constant and are computed based on the input size. For further details and formulations, one may refer to the TensorFlow (Abadi *et al.*, 2015) and PyTorch (Paszke *et al.*, 2019) handbooks.

C. A brief review of theoretical aspects of Fourier neural operators

We focused on the technical aspects and computer implementation of FNO layers in Secs. II A and II B. Theoretical aspects of FNO layers have already been vastly explained and discussed in the literature (Li *et al.*, 2020a). In this subsection, we briefly review the theory behind FNO layers and highlight some important features.

As discussed in Sec. II A, an FNO layer comprises two main operators: the integral kernel operator and the linear transformation. We overview the integral kernel operator. We consider the bounded domain D such that $D \subset \mathbb{R}^{d}$, where d indicates the physical dimensionality of the problem and is equal to 3 (i.e., $d = 3$) for the current problem since we deal with three-dimensional porous media. We further show the input of the FNO layer by $b(x)$ with $x \in D$, where b is a function representing all the operators applied to x when it arrives at the gate of the FNO layer. Moreover, we define the periodic kernel function $\tau : \mathbb{R}^{2(d+d_a)} \to \mathbb{R}^{\text{width}\times\text{width}}$, where d_a is the number of input features and is equal to 1 (i.e., $d_a = 1$) in this study, because the input of the deep learning framework is only a cubic binary array (representing a porous medium), and this array only provides one feature, which is the geometry of the porous medium. Additionally, recall that width is the channel width of the FNO layer, as illustrated in Secs. II A and II B. Following the formulation proposed by Li *et al.* (2020a), the operation of the integral kernel $\mathcal{K}$ on the function $b(x)$ in the continuous space is defined as

$$\mathcal{K}b(x) = \int_{D} \tau(x, y) b(y)\, dy, \quad \forall x \in D. \tag{4}$$

Following the original design of FNO layers by Li *et al.* (2020a), we introduce the condition $\tau(x, y) = \tau(x - y)$. By applying the convolution theorem as detailed in the literature (Li *et al.*, 2020a), the following expression for the integral kernel operator is obtained:

$$\mathcal{K}b(x) = \mathcal{F}^{-1}(\mathcal{F}(\tau) \cdot \mathcal{F}(b(x))), \quad \forall x \in D, \tag{5}$$

where the Fourier transform and its inverse are shown by $\mathcal{F}$ and $\mathcal{F}^{-1}$, respectively. We introduce $\mathcal{R}$ as the learnable Fourier transform of τ such that

$$\mathcal{R} = \mathcal{F}(\tau). \tag{6}$$

Beyond the theory, we must implement these mathematical concepts in a deep learning framework. In this way, we work with discrete spaces and consequently discrete modes of the Fourier transform. Hence, $\mathcal{R}$ is implemented as a neural network. Additionally, each porous medium is represented by n^3 discrete points such that $\{x_1, x_2, \ldots, x_{n^3}\} \subset D$. Moreover, Fourier series expansions are truncated at a maximum number of modes $m_{\max}$ computed as

$$m_{\max} = \left|\left\{ m \in \mathbb{Z}^{d} : |m_j| \leq m_{\max,j},\ \text{for}\ j = 1, \ldots, d \right\}\right|, \tag{7}$$

where $m_{\max,j}$ is the maximum number of modes taken in the dimension j, and is a hyper-parameter of the FNO layer. Note that since we work on three-dimensional problems in the current study, $d = 3$, and thus, there are only $m_{\max,1}$, $m_{\max,2}$, and $m_{\max,3}$. As a result, the components of the $\mathcal{R} \cdot \mathcal{F}(b(x))$ operation can be computed by the following formulation:

$$[\mathcal{R}\cdot\mathcal{F}(b(x))]_{m,i}=\sum_{j=1}^{\text{width}}[\mathcal{R}]_{m,i,j}[\mathcal{F}(b(x))]_{m,j},$$
$$m=1,\ldots,m_{\max},\quad i=1,\ldots,\text{width},\qquad(8)$$

where $[\mathcal{R}]\in\mathbb{C}^{m_{\max}\times\text{width}\times\text{width}}$ is the matrix representation of $\mathcal{R}$ in the discrete space. $[\mathcal{R}\cdot\mathcal{F}(b(x))]\in\mathbb{C}^{m_{\max}\times\text{width}}$ and $[\mathcal{F}(b(x))]\in\mathbb{C}^{m_{\max}\times\text{width}}$ are similarly defined. To increase computing efficiency and enable parallel computing, the operator $[\mathcal{R}]$, for the current three-dimensional problem, is better to be implemented as a five-dimensional matrix expressed as

$$\mathbf{R}_{m_{\max,1}\times m_{\max,2}\times m_{\max,3}\times\text{width}\times\text{width}}.\qquad(9)$$

As can be seen from Eq. (9), the size of matrix **R**, and thus the count of trainable parameters in the FNO layer, is a function of the number of maximum Fourier modes at each dimension and the channel width of the FNO layer. Recall that these parameters (i.e., $m_{\max,1}$, $m_{\max,2}$, $m_{\max,3}$, and width) are the hyperparameter of FNO layers and need to be tuned by potential users.

III. DATA GENERATION

To generate synthetic data to examine the deep learning framework under investigation in this study, we consider cubic porous medium domains with length L along each side, spatial correlation length of l_c, and porosity of ϕ (the ratio of pore spaces to the total volume of a porous medium). We use the truncated Gaussian algorithm (Lantuejoul, 2002; Le Ravalec-Dupin *et al.*, 2004) to generate synthetic porous media. In practice, we create three-dimensional cubic arrays of dimension $n\times n\times n$, populated with random numbers conforming to a normal distribution with the characteristics of a mean value of 0.0 and a standard deviation of 1.0. Subsequently, we filter the arrays by a three-dimensional Gaussian smoothing kernel with a standard deviation of 5.0 and a filter size commensurate with a spatial correlation length (l_c) of 17. We then subject the arrays to a binarization process via a thresholding number such that the porosity (ϕ) of the resulting arrays lies within the range of [0.125, 0.200]. We use the MATLAB software to handle the above-described steps. We set L as $n\times\delta x$, where δx represents the length of each side of a pixel in porous media. We set δx to 0.003 m. We generate porous media with three different sizes by considering three different values for n, such that $n_1=40$, $n_2=48$, and $n_3=56$. In this way, each cubic porous medium can be characterized by its size as n^3 (e.g., 40^3, 48^3, and 56^3). For each n, we generate 1250 data. We randomly split the generated data corresponding to each size into three categories of training (80%, i.e., 1000 data), validation (10%, i.e., 125 data), and test (10%, i.e., 125 data). Hence, there are 3750 data in total, 3000 data for the training set, 375 data for the validation set, and 375 data for the test set. Figure 3 exhibits a few examples of the generated synthetic data.

To stimulate the incompressible viscous Newtonian flow within the generated porous media, we apply a constant pressure gradient in the x direction ($\Delta p/L$). Zero velocity boundary condition is applied at the top and bottom of the porous medium on the y–z planes. Given the geometry and boundary conditions illustrated above, we use a Lattice Boltzmann solver (Keehm *et al.*, 2004) to solve the continuity and steady-state Stokes equations, which are written as follows:

$$\nabla\cdot\boldsymbol{u}=0,\quad\text{in }V,\qquad(10)$$
$$-\nabla p+\mu\Delta\boldsymbol{u}=\mathbf{0},\quad\text{in }V,\qquad(11)$$

where μ is the dynamic viscosity, $\boldsymbol{u}$ and p indicate, respectively, the velocity vector and pressure fields in the pore space of the porous medium, V. In the next step, we compute the permeability in the $x-$ direction (k) using Darcy's law (Darcy, 1856)

$$k=-\frac{\mu\bar{U}}{\Delta p/L},\qquad(12)$$

where $\bar{U}$ shows the average velocity in the entire porous medium (i.e., including solid matrices). The computed permeabilities of our dataset fall in the range [20 mD, 200 mD].

IV. TRAINING

To accelerate the convergence of the training procedure, the output training data (i.e., permeability) are scaled in the range of [0, 1] using the maximum and minimum values of the training set. Note that although we train a single neural network simultaneously on porous media with three different sizes (corresponding to n_1, n_2, and n_3), we normalize the permeability of porous media of each size using

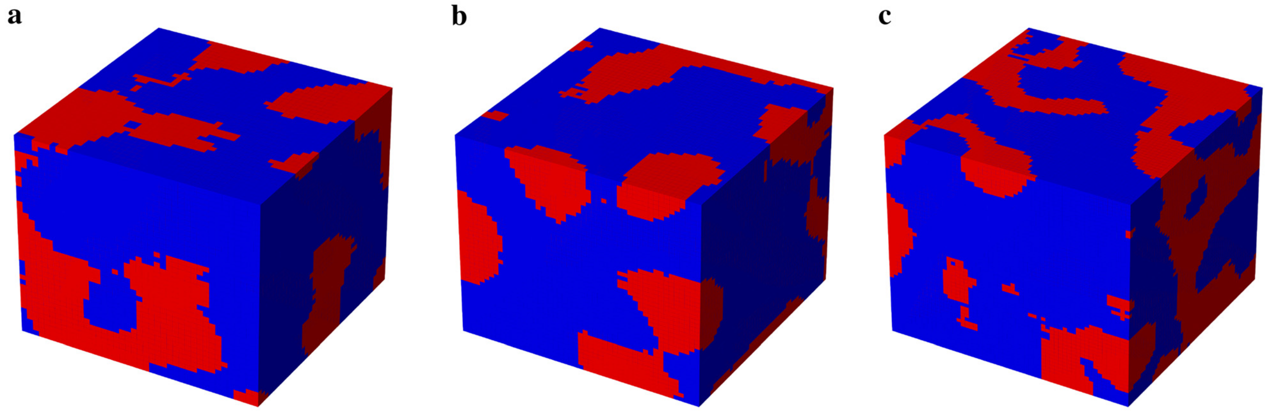


FIG. 3. A few examples of synthetically generated three-dimensional digital porous media for training the proposed neural network; (a) an image of size 40^3, (b) an image of size 48^3, and (c) an image of size 56^3. Blue represents grain space, while red indicates pore space.

the maximum and minimum values of the specific size. Mathematically, it can be written as

$$\{\hat{k}_{\text{truth}}\}_{n_j} = \frac{\{k\}_{n_j} - \min\{k\}_{n_j}}{\max\{k\}_{n_j} - \min\{k\}_{n_j}}, \quad j = 1, 2, \text{and } 3, \tag{13}$$

where $\hat{k}_{\text{truth}}$ shows the ground truth scaled permeability. Moreover, for instance, $\{k\}_{n_1}$ indicates the training data containing porous media with the size of 40^3 (because $n_1 = 40$). Note that we eventually rescale predicted permeability ($\hat{k}_{\text{prediction}}$) to the physical domain ($k_{\text{prediction}}$) for analyzing the neural network performances. Note that in the application of predicting the permeability of porous media using deep learning models, the presence of noisy data or outliers in the training set indicates that at least one sample of porous media has a permeability deviating significantly from the distribution observed in the rest of the training set. This means that data normalization, using Eq. (13), would result in the permeability values of the training set clustering near 0 or 1. Such a scenario would seriously impair the training process of any neural network, including the one proposed in this study. Therefore, data cleaning is a crucial step before proceeding with data normalization. Concerning the loss function, we use the mean squared error function defined as

$$\text{Loss} = \frac{1}{N}\sum_{i=1}^{N}(\hat{k}_{\text{prediction}} - \hat{k}_{\text{truth}})^2, \tag{14}$$

where N is the number of data in the training set (i.e., $N = 3000$). Note that using the relative mean squared error as the loss function does not lead to a significant difference in the results based on our experiments. We set the number of modes in each dimension to 2 (i.e., set $m_{\max,1} = 2$, $m_{\max,2} = 2$, and $m_{\max,3} = 2$). The channel width of the discrete Fourier space is set to 64 (i.e., width = 64). It is worth noting that both the number of modes and the channel width play pivotal roles in the network performance. Detailed discussions on their significance and implications are provided in Secs. V B and V C, respectively. Additionally, we implement three units of FNOs in the network. The Adam optimizer (Kingma and Ba, 2014) is used. A constant learning rate of 0.001 is selected. We use the stochastic gradient descent (Goodfellow *et al.*, 2016) with a mini-batch size of 50. As discussed in Sec. II, the architecture of FNOs is designed to be independent of the spatial resolution of input images. During the training process, however, all the input images within a mini-batch must be the same size. In practice, each epoch of training is characterized by an inner loop that iterates through mini-batches of differing porous medium sizes (i.e., 40^3, 48^3, and 56^3). Within this loop, the training process starts with a mini-batch of data of size 40^3, followed by one of size 48^3, and then continues to 56^3, in sequence until all the data in the training set are covered within the epoch. Note that the trainable parameters of the network are updated only at the end of each epoch. Our deep learning experiments show that the order in which these differently sized porous media are fed within an epoch has no significant influence on the result accuracy and convergence speed, whether starting with the porous media of size 40^3, followed by 48^3 and 56^3, or any other permutation. From a software perspective, we employ the NVIDIA A100 (SXM4) graphic card with 80 GB of RAM for training the networks.

In the last paragraph of this subsection, we address the metric used for assessing the effectiveness of permeability prediction. We use the coefficient of determination, also known as the R^2 score, which can be calculated using the following formula:

$$R^2 = 1 - \frac{\sum_{i=1}^{Q}(k_{\text{truth}_i} - k_{\text{prediction}_i})^2}{\sum_{i=1}^{Q}(k_{\text{truth}_i} - \bar{k})^2}, \tag{15}$$

where Q represents the number of the data in a set (e.g., training, test, etc.) and $\bar{k}$ is the average value of the set $\{k_{\text{truth}_i}\}_{i=1}^{Q}$.

V. RESULTS AND DISCUSSION

A. General analysis

As illustrated in Fig. 4, the success of our approach is evident in the R^2 score, 0.968 09, obtained for the test set (e.g., 375 data). Additionally, Fig. 5 specifically showcases the R^2 scores for the test set but individualized for each cubic size (i.e., 40^3, 48^3, and 56^3). As can be seen in Fig. 5, the R^2 scores obtained are equal to 0.968 30, 0.969 78, and 0.966 07, respectively, for the cubic digital porous media of sizes 40^3, 48^3, and 56^3. The range of R^2 scores for the three different sizes remains at an excellent level, demonstrating that our FNO-based framework is robust and not overfitted to any specific size. Regarding the speedup achieved using the proposed deep learning framework, it predicts the permeability of the test set approximately in 18 s using our graphics processing unit (GPU) machine. In contrast, computing the permeability of the same data with our in-house Lattice Boltzmann Method code, developed in C++ programing language, needs approximately 27 min on a single Intel(R) Core processor with a clock rate of 2.30 GHz. As a result, the average speedup factor we have accomplished is approximately 90 times faster compared to our conventional numerical solver. It is important to mention that the reported speedup factor is highly dependent on the efficiency of the numerical solver and the computing power. For instance, our numerical solver is a custom that operates on a single central processing unit. Modern and commercial applications (e.g., COMSOL and GeoDict) are significantly faster compared to our C++ code.

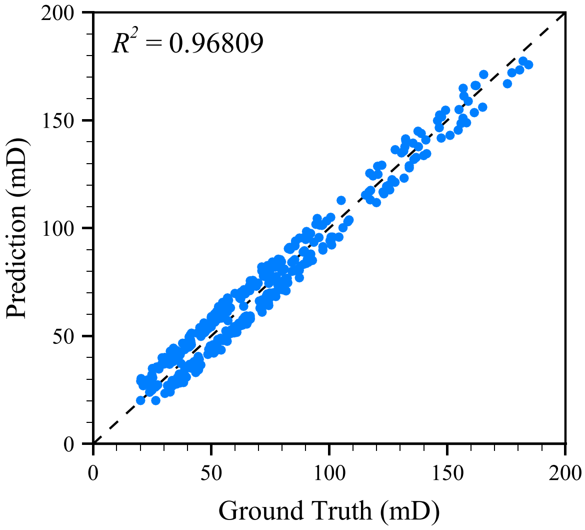


FIG. 4. R^2 plots for the test set (375 data) using the proposed approach for classification of multi-sized images.

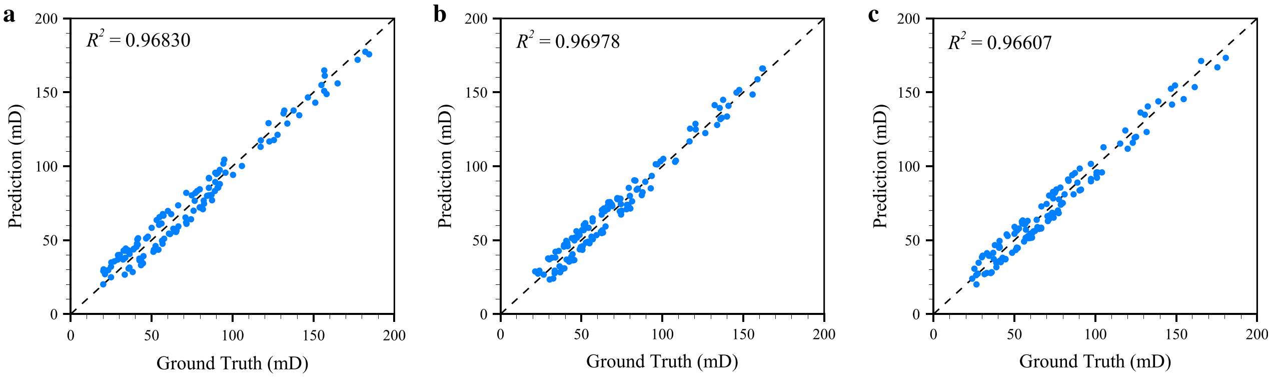


FIG. 5. R^2 plots for the test set (375 data) using the proposed approach for the classification of multi-sized images. The results are individually shown for (a) images of size 40^3 (125 data), (b) images of size 48^3 (125 data), and (c) images of size 56^3 (125 data).

B. Number of Fourier modes in each dimension

Our deep learning experiments demonstrate that there is a critical interplay between the number of modes (i.e., $m_{\max,1}$, $m_{\max,2}$, and $m_{\max,3}$) set in the proposed FNO framework and the tendency for overfitting during the training procedure. Accordingly, setting the number of modes beyond 2 leads to a severe divergence between the training and validation loss. This fact can be observed in Fig. 6 when we set $m_{\max,1} = 7$, $m_{\max,2} = 7$, and $m_{\max,3} = 7$ or $m_{\max,1} = 10$, $m_{\max,2} = 10$, and $m_{\max,3} = 10$. The reported results indicate that the number of modes plays a critical role in the FNO model generalization. A further survey of the influence of the number of modes in the FNO configuration is performed by varying the number of modes in all three principal directions, from 2 to 10, and the obtained R^2 scores are tabulated in Table I. Accordingly, the optimal mode configuration for avoiding overfitting is 2, as the divergence between the validation and training loss is minimized. Consequently, a careful selection of the number of modes in the FNO units is necessary to make the deep learning framework robust and reliable for the image classification application. The consequence of this scenario is observable in Fig. 7, where we plot the R^2 score for the test sets, for example, for the choice of $m_{\max,1} = m_{\max,2} = m_{\max,3} = 4$, $m_{\max,1} = m_{\max,2} = m_{\max,3} = 7$, and $m_{\max,1} = m_{\max,2} = m_{\max,3} = 10$. In all of these cases, the R^2 scores obtained for the prediction of the permeability of the porous media in the test set are less than 0.4.

We perform two other experiments. In the first one, we set only one mode (e.g., $m_{\max,3}$) to 10 ($m_{\max,3} = 10$) and the other two modes to 2 (i.e., $m_{\max,1} = 2$ and $m_{\max,2} = 2$). In the second one, we set only two modes (e.g., $m_{\max,2}$ and $m_{\max,3}$) to 10 and the reminder mode to 2 (i.e., $m_{\max,1} = 2$). The outputs of these two experiments are illustrated in Fig. 8. As can be seen in Fig. 8, the resulting R^2 scores of the test set are equal to 0.222 98 and 0.347 28, respectively, for the first and second experiments. Accordingly, we conclude that even increasing one mode beyond 2 drastically negatively affects the performance of the proposed FNO framework for the current application. Hence, the main challenge of working with the proposed network is its high sensitivity to the number of modes. As discussed in this subsection, changing even the number of modes in one dimension leads to overfitting of the network on the training data and a lack of efficiency in predicting the test data, consequently resulting in a lack of generalizability.

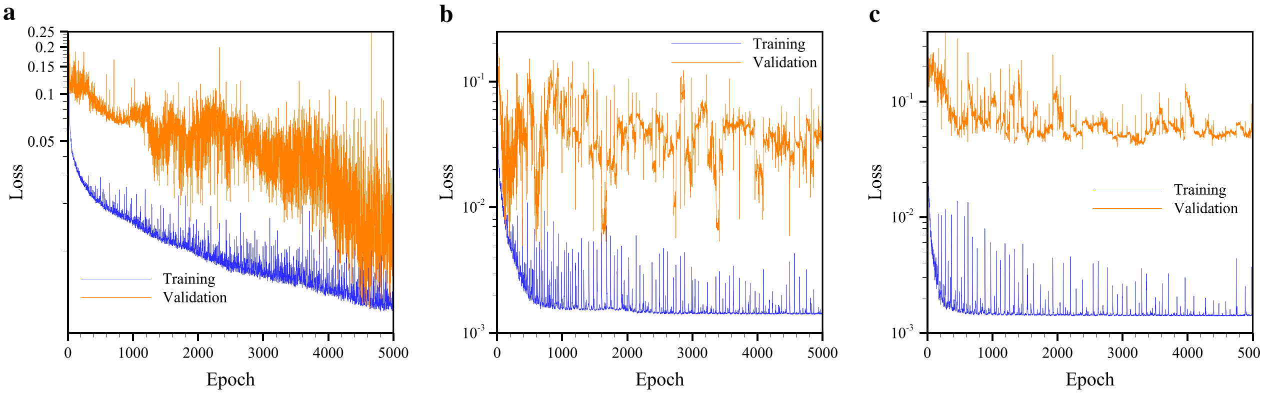


FIG. 6. Evolution of the loss function for the validation and training sets for the choice of (a) $m_{\max,1} = m_{\max,2} = m_{\max,3} = 2$, (b) $m_{\max,1} = m_{\max,2} = m_{\max,3} = 7$, and (c) $m_{\max,1} = m_{\max,2} = m_{\max,3} = 10$.

TABLE I. R^2 score of the test set for different mode numbers of the proposed FNO-based framework.

Number of modes in each dimension	2	3	4	5	6	7	8	9	10
R^2 score	0.968 09	0.154 16	0.267 57	0.263 61	0.233 25	0.387 89	0.404 33	0.317 73	0.288 39

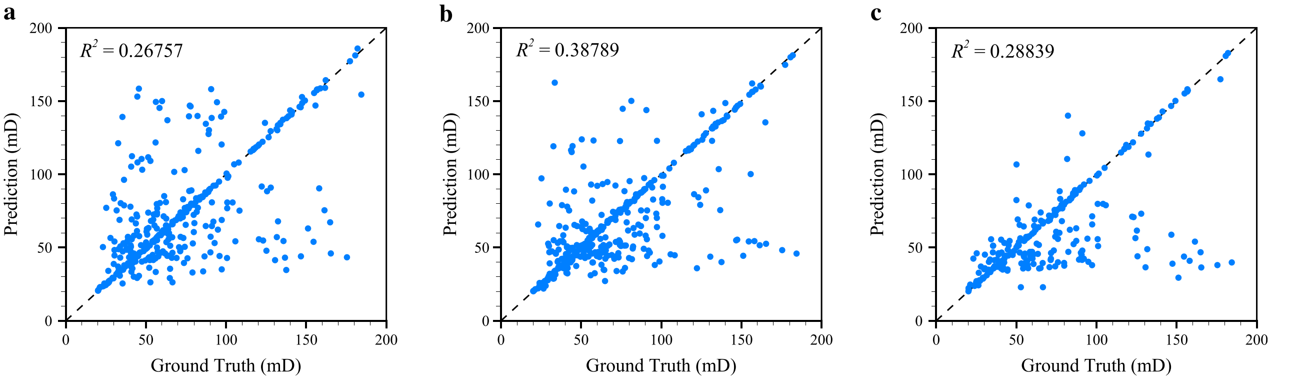


FIG. 7. R^2 plots for the test set (375 data) using the proposed approach for the classification of multi-sized images for the choice of (a) $m_{max,1} = m_{max,2} = m_{max,3} = 4$, (b) $m_{max,1} = m_{max,2} = m_{max,3} = 7$, and (c) $m_{max,1} = m_{max,2} = m_{max,3} = 10$.

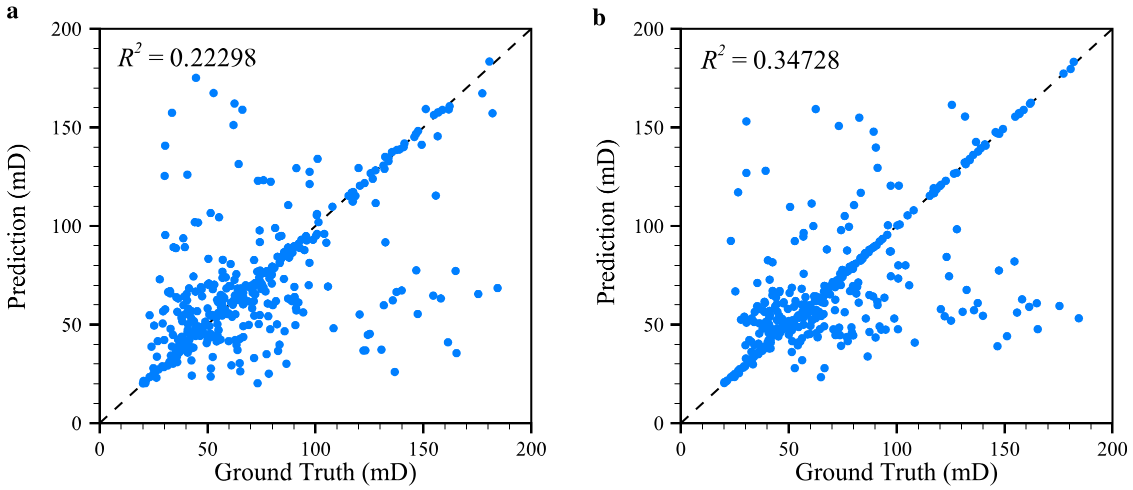


FIG. 8. R^2 plots for the test set (375 data) using the proposed approach for the classification of multi-sized images for the choice of (a) $m_{max,1} = m_{max,2} = 2$ and $m_{max,3} = 10$, and the choice of (b) $m_{max,1} = 2$ and $m_{max,2} = m_{max,3} = 10$.

C. Channel width of FNOs

We further analyze the impact of different channel widths on the performance of the introduced deep learning framework. Based on our machine learning experiments, R^2 scores obtained for the channel width of 8, 27, 64, and 125 are 0.499 04, 0.816 18, 0.968 15, and 0.944 57, respectively. When the channel width decreases from 64 to 27 or to 8, a significant drop in the R^2 score is observed. Notably, increasing the channel width beyond 64 to 125 also leads to a slight decrease in the precision of permeability predictions.

As discussed in Sec. II C, the choice of channel width is directly related to the number of trainable parameters, which are 30 897, 163 783, 828 673, and 3 096 531 for each respective channel width. Moreover, the channel width also determines the size of the max pooling, representing the size of the global feature vector. Hence, optimizing channel width is critical. Small channel width leads to poor performance, whereas large channel width imposes high computational costs and memory allocation without necessarily a significant performance improvement.

D. Number of FNO units

We investigate the effect of varying the number of FNO units (see Fig. 1). Deep learning experiments are conducted using one, two, three, four, and five units to assess the impact on the introduced FNO performance. By computing the R^2 score across the test set, we realize that there is no significant improvement in the prediction accuracy. R^2 score for the FNO configuration with one, two, three, four, and five units are, respectively, 0.827 67, 0.917 03, 0.968 13, 0.967 59, and 0.978 18. Hence, adding more units (beyond 3) and making the network deeper does not have a remarkable effect on the prediction accuracy. However, the number of trainable parameters and consequently, the computational cost and required GPU memory (e.g., RAM) escalated by adding FNO units. For example, 820 353, 824 513, 828 673, 832 833, and 836 993 are, respectively, the number of trainable parameters of the model with one, two, three, four, and five layers of FNOs.

E. Activation functions

We give particular attention to the effect of choosing an activation function on the prediction ability of our FNO model. In the primary setup, we configure all layers to employ the ReLU activation function, except the last layer of the classifier, where we utilize a sigmoid function. We implement two alternative setups. In the first one, we alter the activation function of the last layer to ReLU, this configuration results in a drastic reduction in the R^2 score of the test set, regardless of if the output permeability is normalized between 0 and 1. In the second setup, we replace the activation function in all layers with sigmoid. As a consequence of this setup, a slight decrease in performance is indicated, as R^2 score of 0.914 78 is obtained for the test set. Note that the training procedure becomes slower in this setup, as the derivative of the sigmoid function results in a more complicated computation graph compared to that one output by the derivation of the ReLU function.

F. Static max pooling vs static average pooling

Within the context of capturing global features in the proposed FNO-based framework, we explore the efficacy of implementing static average pooling as an alternative to static max pooling. Our machine learning experiment yields a R^2 score of 0.944 78 in this case, demonstrating a marginal diminishment in the network performance compared to the presence of static max pooling. As supported by the literature (Qi *et al.*, 2017a; 2017b; Kashefi *et al.*, 2021; Kashefi and Mukerji, 2022; and Kashefi *et al.*, 2023), max pooling is a preferred technique for classification tasks compared to average pooling. Our finding shows a similar pattern for the introduced FNO-based framework.

G. Generalizability

In this subsection, we assess the generalizability ability of the proposed FNO-based framework. Note that the concept of generalizability in the context of the present work extends to the network's performance to predict the permeability of cubic porous media with unseen sizes. As discussed in Sec. IV, the network was initially trained using porous media with cubic geometries of sizes 40^3, 48^3, and 56^3. To examine the network capacity to generalize, we predict the permeability of porous media with sizes 36^3, 44^3, 52^3, and 60^3 with 375 cubes for each of these sizes using our pretrained FNO-based framework. Figure 9 shows a few examples of these synthetic data, generated for the purpose of examining the network generalizability. As shown in Fig. 10, a slight decline is observed in the accuracy of permeability predictions for porous media with unseen sizes. However, the obtained R^2 scores remain in an excellent range. These scores are 0.931 85, 0.911 24, 0.915 00, and 0.908 44 for the porous media sizes of 36^3, 44^3, 52^3, and 60^3, respectively. As another observation, the performance of our approach is marginally higher in predicting the permeability of unseen porous media with smaller cubic sizes. As highlighted in Fig. 10, R^2 scores of porous media with sizes of 36^3 are greater than ones with a size of 44^3. A similar scenario occurs when we compare porous media of sizes 52^3 and 60^3. This can be attributed to the fact that, for smaller sizes, the fixed-size vector of the global feature encodes the features of smaller cubes more effectively. Moreover, note that the vector size is the same as the width channel. As a last comment in this subsection, to enhance the network's generalizability, a potential strategy could involve expanding the training dataset to include more than the initial three sets of geometry sizes.

H. Comparison with intuitive approach

1. Classification of fixed-sized image

For the comparison between the proposed approach (see Fig. 1) and the intuitive approach (see Fig. 2), we consider the problem of predicting the permeability of porous media with fixed cubic sizes. Specifically, we consider a size of 48^3. Similar outputs are observed for other sizes. To ensure a fair comparison, both methodologies are investigated under similar conditions. Specifically, both methods are set to have an approximately equal number of trainable parameters (i.e., 828 738 for the intuitive strategy and 828 673 for our approach). Accordingly, the size of the vector representing the global feature is 64 in both methods. All other parameters such as the number of modes in each direction, the number of FNO units, the classifier architecture,

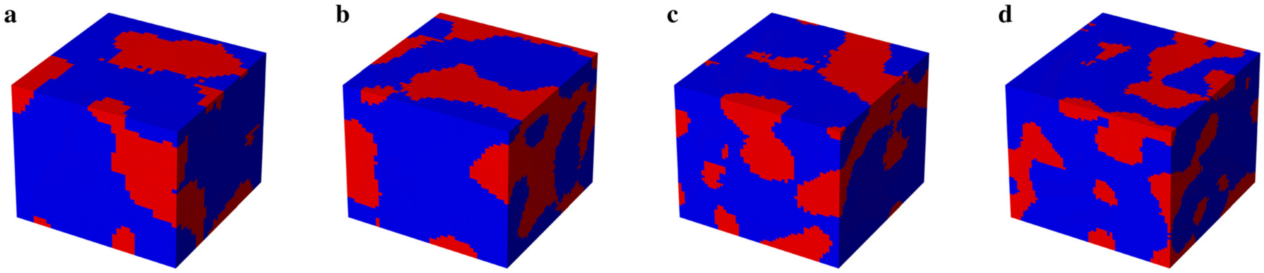


FIG. 9. A few examples of synthetically generated three-dimensional digital porous media for examining the generalizability of the proposed neural network; (a) an image of size 36^3, (b) an image of size 44^3, (c) an image of size 52^3, and (d) an image of size 60^3. Blue represents grain space, while red indicates pore space.

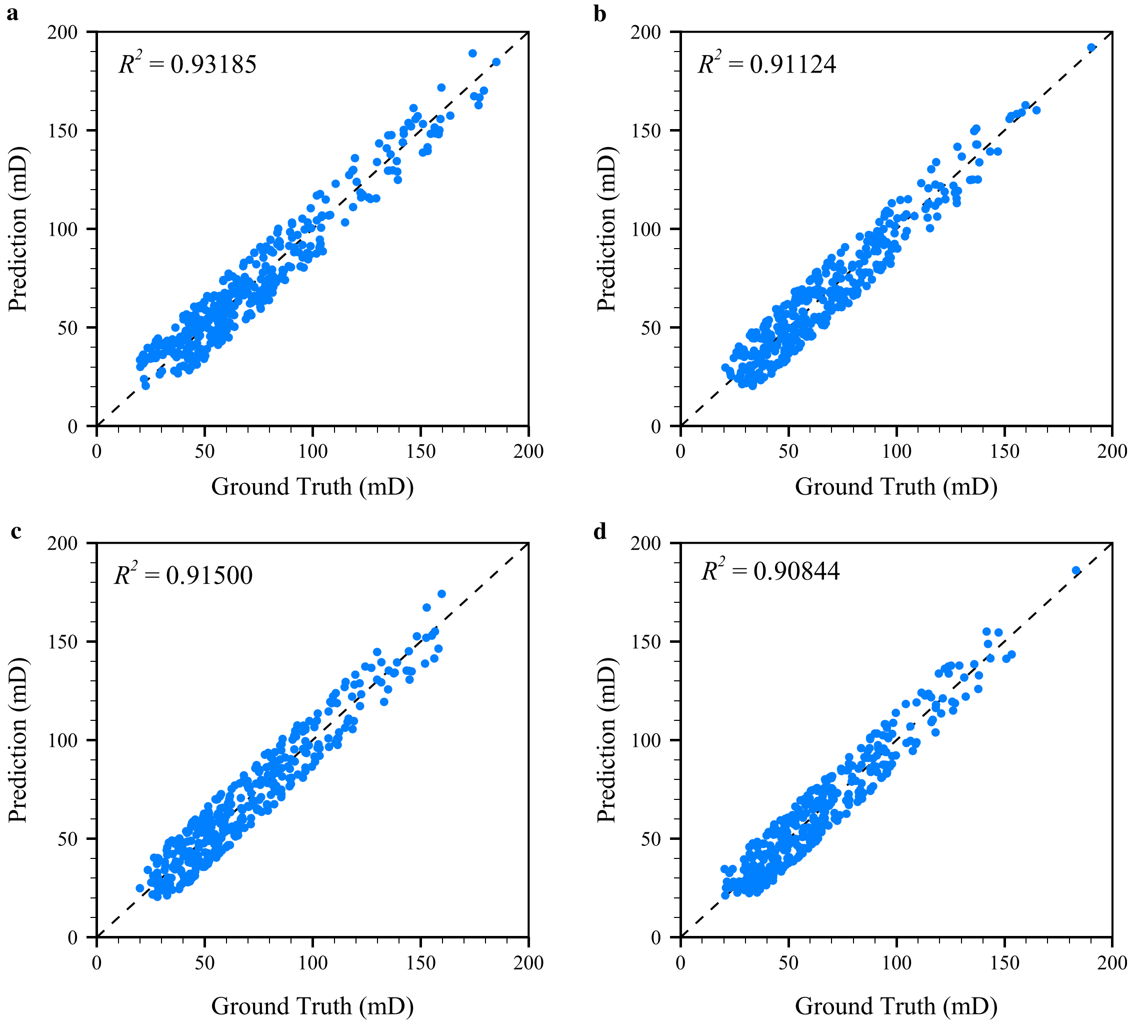


FIG. 10. R^2 plots demonstrating the generalizability of the proposed approach in classifying multi-sized images. The network, trained on images of sizes 40^3, 48^3, and 56^3, is used to predict images of sizes (a) 36^3 (375 data), (b) 44^3 (375 data), (c) 52^3 (375 data), and (d) 60^3 (375 data).

16 May 2024 02:23:56

and size, are the same in both methods and are set as those listed in Sec. IV (i.e., the training section).

Our results demonstrate that both methods perform proficiently, with the R^2 score of 0.993 48 and 0.973 60 over the test set for the intuitive approach (see Fig. 2) and the proposed approach (see Fig. 1), respectively. The evolution of the loss function for the training and validation sets indicates a convergence after approximately 3000 epochs. This deep learning experiment confirms an approximately equivalent computational cost between the two approaches. Hence, when the image size of training data are fixed, both strategies are effective for the defined image classification task and there is no significant advantage for one method over the other, according to our analysis. As a last point in this subsection, we note that one may also use static max pooling in the architecture of the traditional approach since the size of porous media is fixed in this experiment. Based on our results, the performance does not change.

2. Classification of multi-sized images

In this subsection, we compare the performance of the proposed approach (see Fig. 1) with the intuitive approach (see Fig. 2) in predicting the permeability of porous media with varying sizes. For a fair comparison between the intuitive approach and the proposed approach, we use the same training, validation, and test set described in Sec. III. The evolution of both training and validation losses is depicted in Fig. 11. Figure 11 indicates a divergence between the training and validation losses for the network used in the intuitive approach, which suffers from overfitting, whereas this is not the case for the proposed approach. The superiority of the proposed approach is also evident by the R^2 score obtained for the test set. Accordingly, the R^2 scores of the proposed approach and the intuitive approach are, respectively, 0.968 09 and −0.426 32. The negative value of the R^2 score for the intuitive approach demonstrates that its model makes worse predictions than a model that simply predicts all outputs as the mean

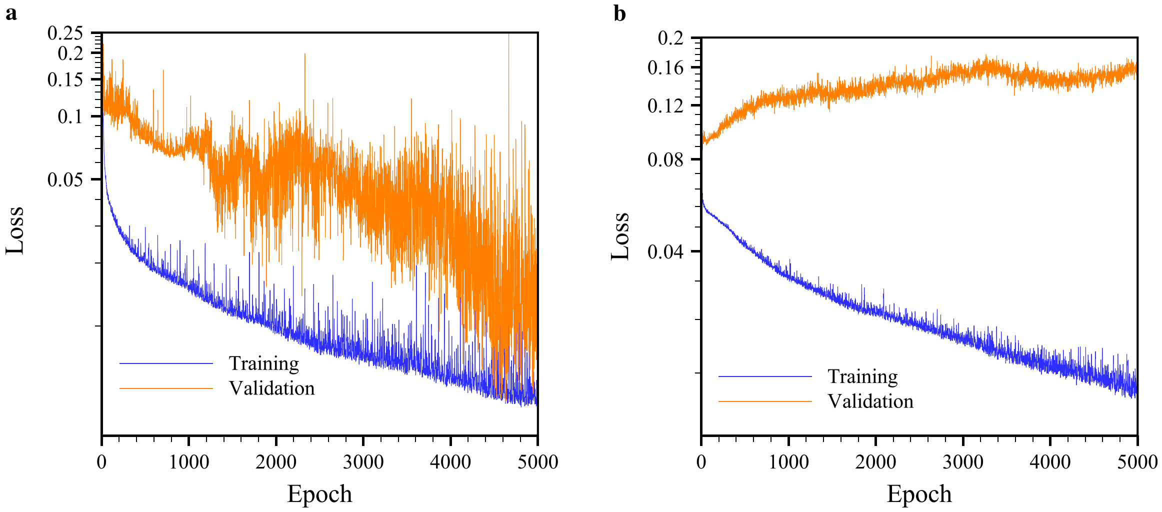


FIG. 11. Evolution of the loss function for the validation and training sets using (a) the proposed approach (see Fig. 1) and (b) the intuitive approach (see Fig. 2).

value of the dataset. Note that changing hyper-parameters, such as the number of modes, channel width, and number of FNO layers, does not improve the model of the intuitive approach.

This flaw stems from two reasons. First, using the intuitive approach, the network captures the global feature after lifting cubes into the original space, while the trainable parameters of the network are mainly defined in the Fourier space. Second, the adaptive max pooling's size is altered depending on the size of the input cubic porous medium. These two together lead to a misrepresentation of the global feature of cubes with different sizes, when the network tends to predict the permeability of the validation and test sets. Note that in Sec. V H 1, we showed that the intuitive approach worked well when it was trained over porous media with fixed sizes. However, the result of our machine learning experiments illustrates that the global features of cubes with different sizes are amalgamated. In contrast, our approach uses static max pooling consistent with the channel width of Fourier neural operators before transitioning back to the original space. This approach enables the capture of global features prior to changing spaces.

VI. SUMMARY AND FUTURE OUTLOOKS

In this research study, we introduced a novel deep learning framework based on Fourier neural operators for classifying images with different sizes (see Fig. 1). Because Fourier neural operators are resolution invariant, they have the potential to be used for the task of multi-sized image classification. To reach this goal, Fourier neural operators must be connected to a classifier, ideally using a pooling operator. To this end, we proposed the novel idea of implementing a static max pooling operator, which functions in a high dimensional space with the size of Fourier channel width. We showed the efficiency and robustness of this framework by predicting the permeability of three-dimensional digital porous media with three different sizes of 40^3, 48^3, and 56^3. We explored the effect of key parameters such as the number of Fourier modes in each dimension, the channel width of the discrete Fourier space, activation functions in different layers, and the number of Fourier units. Additionally, we showed that while the network was only trained on the porous media with the sizes of 40^3, 48^3, and 56^3, it could successfully predict the permeability of the porous media with the sizes of 36^3, 44^3, 52^3, and 60^3, indicating its generalizability. Moreover, we demonstrated that the idea of implementing an adaptive max pooling (see Fig. 2), as an intuitive approach for connecting the FNO layers to the classifier, showed a lack of performance when predicting the permeability of porous media of varying sizes. Note that the adaptive max pooling operated in spatial spaces and that pooling had to be adaptive to handle input images with varying sizes.

As a future research direction, we aim to adapt the current architecture and extend its capabilities to image classification. In contrast to the problem of permeability prediction, this approach reduces the problem's dimensionality to two. Additionally, given that the standard dataset for image classification is usually large, we anticipate improved generalizability of the proposed framework. As another research direction, we would like to examine our deep learning framework using real data rather than synthetic data. We aim to expand our work to a variety of porous media, including biological tissues and fuel cells.

ACKNOWLEDGMENTS

Financial support by the Shell-Stanford collaborative project on digital rock physics was acknowledged. Additionally, the first author would like to thank Professor Gege Wen at Imperial College London for her helpful guidance and discussion on the software engineering aspects of this study. Furthermore, we are grateful to the reviewers for their valuable feedback.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Ali Kashefi: Conceptualization (lead); Formal analysis (lead); Methodology (lead); Software (lead); Visualization (lead); Writing – original draft (lead); Writing – review & editing (lead). **Tapan**

Mukerji: Formal analysis (supporting); Funding acquisition (lead); Investigation (supporting); Methodology (supporting); Project administration (lead); Resources (lead); Supervision (lead); Writing – review & editing (supporting).

DATA AVAILABILITY

The Python code for the three-dimensional problems is available on the following GitHub repository at https://github.com/Ali-Stanford/FNOMultiSizedImages.

REFERENCES

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X., "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, see https://www.tensorflow.org/ for software available from tensorflow.org.

Andra, H., Combaret, N., Dvorkin, J., Glatt, E., Han, J., Kabel, M., Keehm, Y., Krzikalla, F., Lee, M., Madonna, C., Marsh, M., Mukerji, T., Saenger, E. H., Sain, R., Saxena, N., Ricker, S., Wiegmann, A., and Zhan, X., "Digital rock physics benchmarks–Part I: Imaging and segmentation," Comput. Geosci. **50**, 25–32 (2013a).

Andra, H., Combaret, N., Dvorkin, J., Glatt, E., Han, J., Kabel, M., Keehm, Y., Krzikalla, F., Lee, M., Madonna, C., Marsh, M., Mukerji, T., Saenger, E. H., Sain, R., Saxena, N., Ricker, S., Wiegmann, A., and Zhan, X., "Digital rock physics benchmarks–Part II: Computing effective properties," Comput. Geosci. **50**, 33–43 (2013b).

Azzizadenesheli, K., Kovachki, N., Li, Z., Liu-Schiaffini, M., Kossaifi, J., and Anandkumar, A., "Neural operators for accelerating scientific simulations and design," arXiv:2309.15325 (2023).

Blunt, M. J., Bijeljic, B., Dong, H., Gharbi, O., Iglauer, S., Mostaghimi, P., Paluszny, A., and Pentland, C., "Pore-scale imaging and modelling," Adv. Water Resour. **51**, 197–216 (2013).

Bonev, B., Kurth, T., Hundt, C., Pathak, J., Baust, M., Kashinath, K., and Anandkumar, A., "Spherical Fourier neural operators: Learning stable dynamics on the sphere," arXiv:2306.03838 (2023).

Brandstetter, J., van den Berg, R., Welling, M., and Gupta, J. K., "Clifford neural layers for PDE modeling," arXiv:2209.04934 (2023).

Chen, G., Liu, X., Li, Y., Meng, Q., and Chen, L., "Laplace neural operator for complex geometries," arXiv:2302.08166 (2023).

Choubineh, A., Chen, J., Wood, D. A., Coenen, F., and Ma, F., "Fourier neural operator for fluid flow in small-shape 2d simulated porous media dataset," Algorithms **16**, 24 (2023).

Darcy, H., *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.

Das, M. K., Mukherjee, P. P., and Muralidhar, K., *Porous Media Applications: Biological Systems* (Springer International Publishing, 2018).

Fanaskov, V. and Oseledets, I. V., "Spectral neural operators," Dokl. Math. **108**, S226–S232 (2023).

Goodfellow, I., Bengio, Y., and Courville, A., *Deep Learning* (MIT Press, 2016), see http://www.deeplearningbook.org.

Gupta, J. K. and Brandstetter, J., "Towards multi-spatiotemporal-scale generalized PDE modeling," arXiv:2209.15616 (2022).

Hao, Z., Wang, Z., Su, H., Ying, C., Dong, Y., Liu, S., Cheng, Z., Song, J., and Zhu, J., "GNOT: A general neural operator transformer for operator learning," in *International Conference on Machine Learning* (PMLR, 2023), pp. 12556–12569.

Hong, J. and Liu, J., "Rapid estimation of permeability from digital rock using 3D convolutional neural network," Comput. Geosci. **24**, 1523–1539 (2020).

Hua, N. and Lu, W., "Basis operator network: A neural network-based model for learning nonlinear operators via neural basis," Neural Networks **164**, 21–37 (2023).

Huang, X., Shi, W., Meng, Q., Wang, Y., Gao, X., Zhang, J., and Liu, T. Y., "Neuralstagger: Accelerating physics-constrained neural PDE solver with spatial-temporal decomposition," arXiv:2302.10255 (2023).

Jiang, P., Yang, Z., Wang, J., Huang, C., Xue, P., Chakraborty, T. C., Chen, X., and Qian, Y., "Efficient super-resolution of near-surface climate modeling using the Fourier neural operator," J. Adv. Model. Earth Syst. **15**, e2023MS003800 (2023a).

Jiang, Z., Zhu, M., Li, D., Li, Q., Yuan, Y. O., and Lu, L., "Fourier-Mionet: Fourier-enhanced multiple-input neural operators for multiphase modeling of geological carbon sequestration," arXiv:2303.04778 (2023b).

Johnny, W., Brigido, H., Ladeira, M., and Souza, J. C. F., "Fourier neural operator for image classification," in *17th Iberian Conference on Information Systems and Technologies (CISTI)* (IEEE, 2022), pp. 1–6.

Kabri, S., Roith, T., Tenbrinck, D., and Burger, M., "Resolution-invariant image classification based on Fourier neural operators," in *Scale Space and Variational Methods in Computer Vision*, edited by Calatroni, L., Donatelli, M., Morigi, S., Prato, M., and Santacesaria, M. (Springer International Publishing, Cham, 2023).

Kashefi, A., Guibas, L. J., and Mukerji, T., "Physics-informed PointNet: On how many irregular geometries can it solve an inverse problem simultaneously? application to linear elasticity," J. Mach. Learn. Model. Comput. **4**, 1–25 (2023).

Kashefi, A. and Mukerji, T., "Point-cloud deep learning of porous media for permeability prediction," Phys. Fluids **33**, 097109 (2021).

Kashefi, A. and Mukerji, T., "Physics-informed PointNet: A deep learning solver for steady-state incompressible flows and thermal fields on multiple sets of irregular geometries," J. Comput. Phys. **468**, 111510 (2022).

Kashefi, A. and Mukerji, T., "Prediction of fluid flow in porous media by sparse observations and physics-informed PointNet," Neural Networks **167**, 80–91 (2023).

Kashefi, A., Rempe, D., and Guibas, L. J., "A point-cloud deep learning framework for prediction of fluid flow fields on irregular geometries," Phys. Fluids **33**, 027104 (2021).

Keehm, Y., Mukerji, T., and Nur, A., "Permeability prediction from thin sections: 3D reconstruction and lattice-Boltzmann flow simulation," Geophys. Res. Lett. **31**, L04606, https://doi.org/10.1029/2003GL018761 (2004).

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

Kontolati, K., Goswami, S., Shields, M. D., and Karniadakis, G. E., "On the influence of over-parameterization in manifold based surrogates and deep neural operators," J. Comput. Phys. **479**, 112008 (2023).

Kovachki, N. B., Lanthaler, S., and Stuart, A. M., "Operator learning: Algorithms and analysis," arXiv:2402.15715 (2024).

Kumeria, T., "Advances on porous nanomaterials for biomedical application (drug delivery, sensing, and tissue engineering)," ACS Biomater. Sci. Eng. **8**, 4025–4027 (2022).

Lanthaler, S., "Computation and analysis of statistical solutions of the incompressible Euler equations," Ph.D. thesis (ETH Zurich, 2021).

Lantuejoul, C., *Geostatistical Simulation: Models and Algorithms* (Springer, 2002).

Le Ravalec-Dupin, M., Roggero, F., and Froidevaux, R., "Conditioning truncated Gaussian realizations to static and dynamic data," SPE J. **9**, 475–480 (2004).

Lee, S., "Mesh-independent operator learning for partial differential equations," in ICML 2022 2nd AI Science Workshop (2022).

Lehmann, F., Gatti, F., Bertin, M., and Clouteau, D., "3D elastic wave propagation with a factorized Fourier neural operator (F-FNO)," Comput. Methods Appl. Mech. Eng. **420**, 116718 (2024).

Li, Z., Huang, D. Z., Liu, B., and Anandkumar, A., "Fourier neural operator with learned deformations for PDEs on general geometries," arXiv:2207.05209 (2022a).

Li, Z., Kovachki, N., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A., and Anandkumar, A., "Fourier neural operator for parametric partial differential equations," arXiv:2010.08895 (2020a).

Li, Z., Kovachki, N., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A., and Anandkumar, A., "Neural operator: Graph Kernel network for partial differential equations," arXiv:2003.03485 (2020b).

Li, Z., Kovachki, N., Azizzadenesheli, K., Liu, B., Stuart, A., Bhattacharya, K., and Anandkumar, A., "Multipole graph neural operator for parametric partial differential equations," Adv. Neural Inf. Process. Syst. **33**, 6755–6766 (2020c).

Li, Z., Kovachki, N. B., Choy, C., Li, B., Kossaifi, J., Otta, S. P., Nabian, M. A., Stadler, M., Hundt, C., Azizzadenesheli, K., and Anandkumar, A., "Geometry-informed neural operator for large-scale 3D PDEs," arXiv:2309.00583 (2023).

Li, Z., Peng, W., Yuan, Z., and Wang, J., "Fourier neural operator approach to large eddy simulation of three-dimensional turbulence," Theor. Appl. Mech. Lett. **12**, 100389 (2022b).

Li, Z., Zheng, H., Kovachki, N., Jin, D., Chen, H., Liu, B., Azizzadenesheli, K., and Anandkumar, A., "Physics-informed neural operator for learning partial differential equations," arXiv:2111.03794 (2021).

Liang, Y. and Fletcher, D., "Computational fluid dynamics simulation of forward osmosis (fo) membrane systems: Methodology, state of art, challenges and opportunities," Desalination **549**, 116359 (2023).

Liu, M., Ahmad, R., Cai, W., and Mukerji, T., "Hierarchical homogenization with deep-learning-based surrogate model for rapid estimation of effective permeability from digital rocks," J. Geophys. Res. **128**, e2022JB025378, https://doi.org/10.1029/2022JB025378 (2023).

Lyu, Y., Zhao, X., Gong, Z., Kang, X., and Yao, W., "Multi-fidelity prediction of fluid flow based on transfer learning using Fourier neural operator," Phys. Fluids **35**, 077118 (2023).

Majumdar, R., Karande, S., and Vig, L., "How important are specialized transforms in neural operators?," in 1st Workshop on the Synergy of Scientific and Machine Learning Modeling @ ICML2023 (2023), see https://openreview.net/forum?id=DU3Z6ZdqhZ.

Masroor, M., Emami Niri, M., and Sharifinasab, M. H., "A multiple-input deep residual convolutional neural network for reservoir permeability prediction," Geoenergy Sci. Eng. **222**, 211420 (2023).

Maust, H., Li, Z., Wang, Y., Leibovici, D., Bruno, O., Hou, T., and Anandkumar, A., "Fourier continuation for exact derivative computation in physics-informed neural operators," arXiv:2211.15960 (2022).

Meng, Y., Jiang, J., Wu, J., and Wang, D., "Transformer-based deep learning models for predicting permeability of porous media," Adv. Water Resour. **179**, 104520 (2023).

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L. *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Proceedings of 33rd International Conference on Neural Information Processing Systems* (ACM, 2019), pp. 8024–8035.

Pathak, J., Subramanian, S., Harrington, P., Raja, S., Chattopadhyay, A., Mardani, M., Kurth, T., Hall, D., Li, Z., Azizzadenesheli, K. *et al.*, "Fourcastnet: A global data-driven high-resolution weather model using adaptive Fourier neural operators," arXiv:2202.11214 (2022).

Peng, W., Qin, S., Yang, S., Wang, J., Liu, X., and Wang, L. L., "Fourier neural operator for real-time simulation of 3D dynamic urban microclimate," Build. Environ. **248**, 111063 (2024).

Peng, W., Yuan, Z., Li, Z., and Wang, J., "Linear attention coupled Fourier neural operator for simulation of three-dimensional turbulence," Phys. Fluids **35**, 015106 (2023a).

Peng, Z., Yang, B., Liu, L., and Xu, Y., "Rapid surrogate modeling of magnetotelluric in the frequency domain using physics-driven deep neural networks," Comput. Geosci. **176**, 105360 (2023b).

Poels, Y., Derks, G., Westerhof, E., Minartz, K., Wiesen, S., and Menkovski, V., "Fast dynamic 1D simulation of divertor plasmas with neural PDE surrogates," arXiv:2305.18944 (2023).

Qi, C. R., Su, H., Mo, K., and Guibas, L. J., "Pointnet: Deep learning on point sets for 3D classification and segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2017a), pp. 652–660.

Qi, C. R., Yi, L., Su, H., and Guibas, L. J., "Pointnet++: deep hierarchical feature learning on point sets in a metric space," in *Advances in Neural Information Processing Systems*, edited by Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (Curran Associates, Inc., 2017b).

Rahman, M. A., Florez, M. A., Anandkumar, A., Ross, Z. E., and Azizzadenesheli, K., "Generative adversarial neural operators," arXiv:2205.03017 (2022a).

Rahman, M. A., Ross, Z. E., and Azizzadenesheli, K., "U-no: U-shaped neural operators," arXiv:2204.11127 (2022b).

Renn, P. I., Wang, C., Lale, S., Li, Z., Anandkumar, A., and Gharib, M., "Forecasting subcritical cylinder wakes with Fourier neural operators," arXiv:2301.08290 (2023).

Subramanian, S., Harrington, P., Keutzer, K., Bhimji, W., Morozov, D., Mahoney, M. W., and Gholami, A., "Towards foundation models for scientific machine learning: Characterizing scaling and transfer behavior," in *37th Conference on Neural Information Processing System* (MIT Press, 2024), p. 36.

Sun, H., Zhou, L., Fan, D., Zhang, L., Yang, Y., Zhang, K., and Yao, J., "Permeability prediction of considering organic matter distribution based on deep learning," Phys. Fluids **35**, 032014 (2023).

Tembely, M., AlSumaiti, A. M., and Alameri, W., "A deep learning perspective on predicting permeability in porous media from network modeling to direct simulation," Comput. Geosci. **24**, 1541–1556 (2020).

Thodi, B. T., Ambadipudi, S. V. R., and Jabari, S. E., "Fourier neural operator for learning solutions to macroscopic traffic flow models: Application to the forward and inverse problems," arXiv:2308.07051 (2023).

Tran, A., Mathews, A., Xie, L., and Ong, C. S., "Factorized Fourier neural operators. arXiv:2111.13802 (2023).

Wen, G., Li, Z., Azizzadenesheli, K., Anandkumar, A., and Benson, S. M., "U-FNO–An enhanced Fourier neural operator-based deep-learning model for multiphase flow," Adv. Water Resour. **163**, 104180 (2022).

White, C., Berner, J., Kossaifi, J., Elleithy, M., Pitt, D., Leibovici, D., Li, Z., Azizzadenesheli, K., and Anandkumar, A., "Physics-informed neural operators with exact 1differentiation on arbitrary geometries," in The Symbiosis of Deep Learning and Differential Equations III (2023a).

White, C., Tu, R., Kossaifi, J., Pekhimenko, G., Azizzadenesheli, K., and Anandkumar, A., "Speeding up Fourier neural operators via mixed precision," arXiv:2307.15034 (2023b).

Wu, J., Yin, X., and Xiao, H., "Seeing permeability from images: Fast prediction with convolutional neural networks," Sci. Bull. **63**, 1215–1222 (2018).

Xi, J., Ersoy, O. K., Cong, M., Zhao, C., Qu, W., and Wu, T., "Wide and deep Fourier neural network for hyperspectral remote sensing image classification," Remote Sens. **14**, 2931 (2022).

Xie, C., Zhu, J., Yang, H., Wang, J., Liu, L., and Song, H., "Relative permeability curve prediction from digital rocks with variable sizes using deep learning," Phys. Fluids **35**, 096605 (2023).

Xiong, W., Ma, M., Sun, P., and Tian, Y., "Koopmanlab: Machine learning for solving complex physics equations," arXiv:2301.01104 (2023).

Yang, H., Li, Z., Sastry, K., Mukhopadhyay, S., Anandkumar, A., Khailany, B., Singh, V., and Ren, H., "Large scale mask optimization via convolutional Fourier neural operator and litho-guided self–training," arXiv:2207.04056 (2022).

Yang, Y., Gao, A. F., Azizzadenesheli, K., Clayton, R. W., and Ross, Z. E., "Rapid seismic waveform modeling and inversion with neural operators," IEEE Trans. Geosci. Remote Sens. **61**, 1–12 (2023).

You, H., Yu, Y., D'Elia, M., Gao, T., and Silling, S., "Nonlocal kernel network (NKN): A stable and resolution-independent deep neural network," J. Comput. Phys. **469**, 111536 (2022).

Zhao, J., George, R. J., Zhang, Y., Li, Z., and Anandkumar, A., "Incremental Fourier neural operator," arXiv:2211.15188 (2022).

Zhao, X., Chen, X., Gong, Z., Zhou, W., Yao, W., and Zhang, Y., "RecFNO: A resolution-invariant flow and heat field reconstruction method from sparse observations via Fourier neural operator," Int. J. Therm. Sci. **195**, 108619 (2024).

Zhao, X., Sun, Y., Zhang, T., and Xu, B., "Local convolution enhanced global Fourier neural operator for multiscale dynamic spaces prediction," arXiv:2311.12902 (2023).

Zhu, M., Feng, S., Lin, Y., and Lu, L., "Fourier-DeepONet: Fourier-enhanced deep operator networks for full waveform inversion with improved accuracy, generalizability, and robustness," arXiv:2305.17289 (2023a).

Zhu, M., Zhang, H., Jiao, A., Karniadakis, G. E., and Lu, L., "Reliable extrapolation of deep neural operators informed by physics or sparse observations," Comput. Methods Appl. Mech. Eng. **412**, 116064 (2023b).

Zou, C., Azizzadenesheli, K., Ross, Z. E., and Clayton, R. W., "Deep neural Helmholtz operators for 3D elastic wave propagation and inversion," arXiv:2311.09608 (2023).