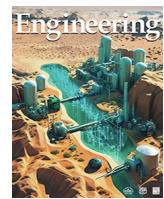




ELSEVIER

Contents lists available at ScienceDirect

# Engineering

journal homepage: [www.elsevier.com/locate/eng](http://www.elsevier.com/locate/eng)

## Research AI in Chemical Engineering—Article

# Porous-DeepONet: Learning the Solution Operators of Parametric Reactive Transport Equations in Porous Media

Pan Huang <sup>a,c</sup>, Yifei Leng <sup>a</sup>, Cheng Lian <sup>a,b,\*</sup>, Honglai Liu <sup>a,b</sup><sup>a</sup>State Key Laboratory of Chemical Engineering & Shanghai Engineering Research Center of Hierarchical Structure Nanomaterials, School of Chemical Engineering, East China University of Science and Technology, Shanghai 200237, China<sup>b</sup>School of Chemistry and Molecular Engineering, East China University of Science and Technology, Shanghai 200237, China<sup>c</sup>Product Planning and New Auto Technologies Research Institute, BYD Auto Industry Company Limited, Shenzhen 518118, China

## ARTICLE INFO

### Article history:

Received 4 May 2023

Revised 4 June 2024

Accepted 5 July 2024

Available online 19 July 2024

### Keywords:

Porous media  
Reactive transport  
Solution operator  
DeepONet  
Neural network

## ABSTRACT

Reactive transport equations in porous media are critical in various scientific and engineering disciplines, but solving these equations can be computationally expensive when exploring different scenarios, such as varying porous structures and initial or boundary conditions. The deep operator network (DeepONet) has emerged as a popular deep learning framework for solving parametric partial differential equations. However, applying the DeepONet to porous media presents significant challenges due to its limited capability to extract representative features from intricate structures. To address this issue, we propose the Porous-DeepONet, a simple yet highly effective extension of the DeepONet framework that leverages convolutional neural networks (CNNs) to learn the solution operators of parametric reactive transport equations in porous media. By incorporating CNNs, we can effectively capture the intricate features of porous media, enabling accurate and efficient learning of the solution operators. We demonstrate the effectiveness of the Porous-DeepONet in accurately and rapidly learning the solution operators of parametric reactive transport equations with various boundary conditions, multiple phases, and multi-physical fields through five examples. This approach offers significant computational savings, potentially reducing the computation time by 50–1000 times compared with the finite-element method. Our work may provide a robust alternative for solving parametric reactive transport equations in porous media, paving the way for exploring complex phenomena in porous media.

© 2024 THE AUTHORS. Published by Elsevier LTD on behalf of Chinese Academy of Engineering and Higher Education Press Limited Company. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

## 1. Introduction

Porous media play a pivotal role in a wide array of industrial sectors [1,2]—encompassing new energy [3], petrochemicals [4], photocatalysis [5], and biomedicine [6]—due to their intricate pore network and substantial specific surface area. The transport and reaction phenomena occurring within porous media are critical determinants influencing essential parameters such as energy storage efficiency, catalytic performance, and adsorption rates of industrial products [7–9]. To ensure the rational design and optimization of industrial products, it is imperative to model these complex transport and reaction processes using partial differential equations (PDEs) [10]. In the realm of industrial research, a significant challenge faced by researchers is the

rapid and accurate resolution of parametric PDEs [11–13]. These specialized PDEs involve parameters such as coefficients, domain geometries, initial conditions (ICs), and boundary conditions (BCs) that vary within defined ranges [14]. However, traditional methodologies (e.g., the finite-element method, FEM) demand extensive computational resources, as independent simulations must be conducted for each parameter variation. The intricacies of grid partitioning and nonlinear numerical computations further intensify these challenges, particularly when addressing multi-physical field interactions within intricate porous structures. Consequently, there is a critical need for innovative approaches to expedite the resolution of parametric PDEs in porous media. By overcoming the limitations of traditional methods, such novel strategies have the potential to significantly enhance the iteration rate of industrial product development, thereby fostering advancements in industrial research and applications.

\* Corresponding author.

E-mail address: [liancheng@ecust.edu.cn](mailto:liancheng@ecust.edu.cn) (C. Lian).

Using deep learning techniques based on neural networks (NNs) presents a promising avenue for addressing forward and inverse problems in numerical simulation technology, potentially revolutionizing the field [15,16]. As the universal approximation theorem indicates, NNs can solve equations by approximating nonlinear functions that satisfy given constraints. Various network architectures, including recurrent NNs (RNNs) [17], convolutional NNs (CNNs) [18], residual networks (ResNets) [19], physics-informed NNs (PINNs) [11], and derivative architectures [20,21], have been applied to solve PDEs. However, existing NNs are limited in their ability to adapt to changes in input parameters, such as alterations in domain geometries, and require retraining to accommodate new constraints. To address this limitation, NNs must learn operators rather than functions, as per a lesser-known but powerful theorem that demonstrates that NNs can approximate continuous nonlinear operators [22]. Building upon this concept, the deep operator network (DeepONet) was introduced by Lu et al. [23], designed to learn explicit and implicit operators—such as integrals, fractional Laplacians, and deterministic and stochastic differential equations—from data. The DeepONet has demonstrated successful applications in solving parametric PDEs in simple one-dimensional (1D) domain geometries across diverse fields, including quantum chemistry [24], hydromechanics [25,26], battery technology [27], power grid analysis [28], organism modeling [29], and contagion dynamics [30]. Furthermore, researchers have extended the DeepONet framework to address more complex parametric PDEs [31,32], deal with low-accuracy data [33,34], and even train NNs without data [11,35]. For example, Cai et al. [31] developed DeepM&Mnet to simulate multi-physics problems in the bulk phase by coupling multiple pretrained DeepONets, while Wang et al. [11] proposed a physics-informed DeepONet inspired by PINNs to alleviate data requirements. These advancements highlight the potential of the DeepONet and related frameworks in revolutionizing the efficient solution of parametric PDEs in the bulk phase across a wide spectrum of industrial and scientific domains.

Although the DeepONets mentioned above have demonstrated impressive capabilities, they encounter challenges when learning the solution operators of parametric PDEs in complex domain geometries such as porous media. These frameworks struggle to extract representative features from intricate porous structures, which limits their effectiveness in resolving parametric PDEs in this context. While some pioneering research [34,36–38] has been conducted on applying DeepONets in complex domain geometries, the studied structures, such as triangles and five-pointed stars, still deviate from those of real porous media [34]. To overcome this challenge, we take inspiration from the powerful image-processing capabilities of CNNs [39]. For example, Cawte and Bazylak [40] successfully utilized a three-dimensional (3D) CNN to accurately predict gas diffusion layer material permeability from the image data of porous fibrous materials. Their success highlights the potential of CNNs for extracting useful features from intricate porous structures, offering promising avenues for solving parametric PDEs in complex domains such as porous media.

In this work, we investigate an innovative extension of the DeepONet framework called the Porous-DeepONet, which is specifically tailored to solve parametric PDEs in porous media by incorporating CNNs into the DeepONet architecture, with a focus on reactive transport equations. First, the efficacy of the Porous-DeepONet is demonstrated through its application in learning solution operators within complex and random porous structures, utilizing the Fick diffusion equation as a representative example. Then, we further illustrate the power of the Porous-DeepONet by changing the type of BCs on the surface of porous media, such as whether there is a reaction, whether there is a boundary layer, and whether it is permeable. The studied equations include the Fick diffusion and surface reaction equation, the advection equa-

tion, and the heat conduction equation. Furthermore, to enhance the versatility of the Porous-DeepONet in addressing coupled equations from multi-physical fields, a combination of the Porous-DeepONet and the DeepM&Mnet is employed to predict the charging process in porous electrodes by solving the Poisson-Nernst-Planck (PNP) equations. To assess the generalizability of the Porous-DeepONet in identifying solutions for non-real porous structures, a simplified porous model is utilized to generate data for validation purposes. In addition, a comparison is made between the solution speed of the Porous-DeepONet and that of the FEM, shedding light on the computational efficiency of the proposed approach. Lastly, the feasibility of employing a physics-informed Porous-DeepONet is discussed, indicating the potential for leveraging domain knowledge to further enhance the model's performance.

## 2. Methodologies

In the subsequent sections, we first introduce the DeepONet, which is based on the generalized universal approximation theorem for the operator. Subsequently, we discuss an extension of the DeepONet called the Porous-DeepONet, which is specifically designed to learn parametric PDEs' solution operators in porous media. Moreover, we outline the approaches for coupling the Porous-DeepONet with other DeepONet frameworks. Finally, we describe the data-generation process, including the generation of real porous structures and simplified porous models, as well as the numerical details of the FEM.

### 2.1. DeepONets applied to porous media: The Porous-DeepONet

#### 2.1.1. Generalized universal approximation theorem for the operator

Suppose that  $(\mathcal{U}, \mathcal{V}, \mathcal{S})$  is a triplet of Banach spaces and  $\mathcal{N} : \mathcal{U} \times \mathcal{S} \rightarrow \mathcal{V}$  is a linear or nonlinear differential operator. Assume that the input functions are expressed as  $\mathbf{u} \in \mathcal{U}$  and the solutions of the parametric PDE as  $\mathbf{s} \in \mathcal{S}$ , respectively. Therefore, the general form of parametric PDEs can be denoted as  $\mathcal{N}(\mathbf{u}, \mathbf{s}) = 0$ . For particular ICs and BCs, we consider that there exists a unique solution  $\mathbf{s} = \mathbf{s}(\mathbf{u}) \in \mathcal{S}$  to  $\mathcal{N}(\mathbf{u}, \mathbf{s}) = 0$ . Then, the solution operator  $G : \mathcal{U} \rightarrow \mathcal{S}$  can be defined as  $G(\mathbf{u}) = \mathbf{s}(\mathbf{u})$ . According to the generalized universal approximation theorem [23], a nonlinear continuous operator  $G$  satisfies the following inequality for any positive number  $\varepsilon$ :

$$|G(\mathbf{u})(\mathbf{y}) - \langle g(\mathbf{u}(\mathbf{x}_1), \mathbf{u}(\mathbf{x}_2), \dots, \mathbf{u}(\mathbf{x}_m)), f(\mathbf{y}) \rangle| < \varepsilon \quad (1)$$

where  $|\cdot|$  is the absolute value;  $\langle \cdot, \cdot \rangle$  is the dot product in  $\mathbb{R}^p$ ;  $g : \mathbb{R}^m \rightarrow \mathbb{R}^p$  and  $f : \mathbb{R}^d \rightarrow \mathbb{R}^p$  are continuous vector functions;  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m$  are  $d$  dimensional vectors, and  $\mathbf{y}$  refers to the points in the domain of  $G(\mathbf{u})$ . However, this theorem only informs us whether the operator  $G$  can be learned and does not indicate how to learn  $G$  efficiently.

#### 2.1.2. The DeepONet

The DeepONet provides a flexible and powerful deep learning framework that encapsulates the generalized universal approximation theorem for operators above [11,23]. A DeepONet comprises two sub-networks: a trunk net and a branch net. The trunk net is responsible for extracting the latent representation of space-time locations  $\mathbf{y} = [t, \mathbf{x}, \mathbf{y}]$ , where  $t$  is time, and the  $\mathbf{x}$  and  $\mathbf{y}$  are space location, while the branch net takes as input the functions  $\mathbf{u} = [\mathbf{u}(\mathbf{x}_1), \mathbf{u}(\mathbf{x}_2), \dots, \mathbf{u}(\mathbf{x}_m)]$  in the form of discretized points. The outputs of the trunk net  $\mathbf{t} \in \mathbb{R}^q$  and branch net  $\mathbf{b} \in \mathbb{R}^q$  are  $\mathbf{t} = [t_1, t_2, \dots, t_q]^T$  and  $\mathbf{b} = [b_1, b_2, \dots, b_q]^T$ , respectively. More specifically, the solution operators  $G$  of a function  $\mathbf{u}$  evaluated at  $\mathbf{y}$  can be expressed by

$$G(\mathbf{u})(\mathbf{y}) \approx \sum_{i=1}^q \underbrace{b_i(\mathbf{u}(\mathbf{x}_1), \mathbf{u}(\mathbf{x}_2), \dots, \mathbf{u}(\mathbf{x}_m))}_{\text{Branch}} \cdot \underbrace{t_i(\mathbf{y})}_{\text{Trunk}} \quad (2)$$

To reduce the generalization error of the DeepONet, a bias  $b_0 \in \mathbb{R}$  is added in the last stage:  $G(\mathbf{u})(\mathbf{y}) \approx \sum_{i=1}^q b_i i + b_0$ . It is worth noting that there are no constraints on the locations of  $\mathbf{x}$  and  $\mathbf{y}$ , which can be chosen randomly or at equal distances. For random sampling, adopting a fully connected NN (FNN) structure is appropriate for the trunk net and branch net [23].

### 2.1.3. The Porous-DeepONet

To learn the solution operators of parametric PDEs in porous media, the various structures of porous media should be inputted into the branch network as the function  $\mathbf{u}$ . These structures can be considered to be image-like data with equal-spaced grids. However, due to the complexity and randomness of these structures, the FNNs commonly used in DeepONets [25,27,30] may not be able to encode the local information of porous structures well. Therefore, other NN architectures may be considered. CNNs have powerful image-processing capabilities and have been successfully applied to predict the transport property of porous media and reconstruct porous structures [41]. Inspired by the DeepONet and CNNs, we propose a framework for solving parametric PDEs in porous media called the Porous-DeepONet, depicted in Fig. 1(a). By inputting porous structure  $\mathbf{u}(x, y)$  and space-time coordinates  $\mathbf{y} = [t, x, y]$ , we obtain the output  $\mathbf{s} = G(\mathbf{u})(\mathbf{y})$  in the domain of the corresponding solution function  $G(\mathbf{u})$ . For example, if we aim to solve the Fick diffusion equation in different porous structures, the inputs and outputs of the Porous-DeepONet are the images of the porous structures, space-time coordinates, and the corresponding concentration. In addition, other parameters  $\theta$  (e.g., the diffusion coefficient) can be inputted into the Porous-DeepONet by adding one more branch

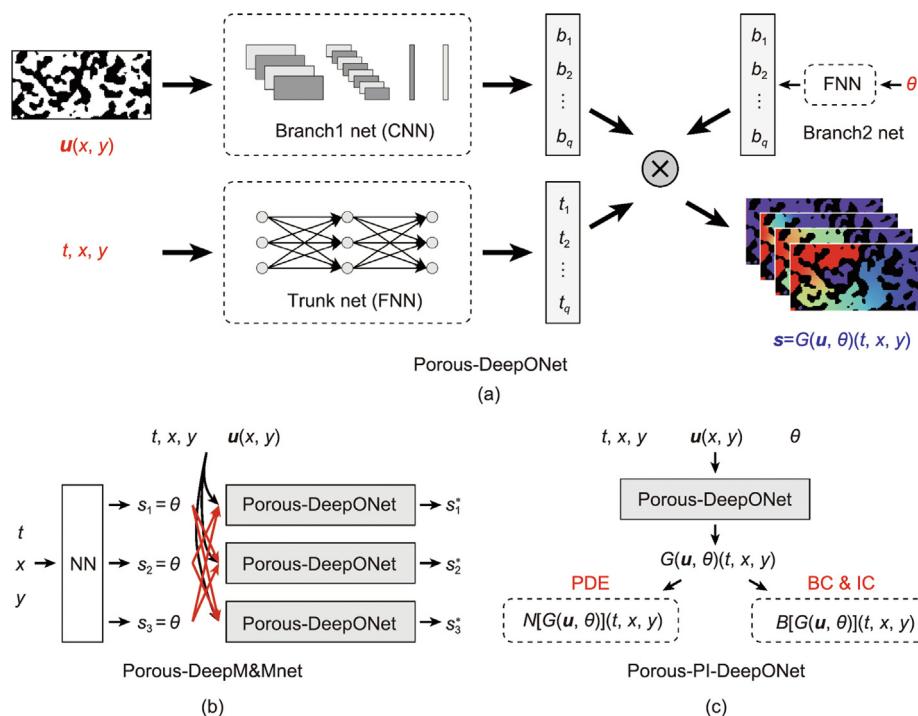
net [37]. Although a few works [29,42] have introduced CNNs into DeepONets, this DeepONet based on CNNs has not been applied to solve parametric PDEs in porous media. For example, Kontolati et al. [42] solved the 2D time-dependent Brusselator reaction-diffusion system with uncertainty sources in the bulk phase using a DeepONet based on CNNs to model an autocatalytic chemical reaction between two species. The function of the CNN in that case was to extract representative features of spatially dependent uncertainty sources, not porous media.

When training, suppose that we sample  $N$  porous structures  $\mathbf{u}$  from function space  $\mathcal{U}$  and  $N$  other parameters  $\theta$ ; in addition, for each input  $\mathbf{u}^{(i)}$  and  $\theta^{(i)}$ , we collect  $P$  points (i.e.,  $\mathbf{y}_{\mathbf{u}^{(i)}}^{(j)}$ ) to evaluate the solution function. Therefore, the total number of training samples should be  $N \times P$ . Notably, the Porous-DeepONet is trained by adopting the Adamax and Huber loss ( $L_{\text{Huber}}$ ) as an optimizer and loss function, respectively:

$$L_{\text{Huber}} = \begin{cases} \frac{1}{2} \frac{1}{NP} \sum_{i=1}^N \sum_{j=1}^P \left[ G(\mathbf{u}^{(i)}, \theta^{(i)}) (\mathbf{y}_{uj}^{(i)}) - \widehat{G}(\mathbf{u}^{(i)}, \theta^{(i)}) (\mathbf{y}_{uj}^{(i)}) \right]^2 & \text{for } |s - \widehat{s}| \leq \delta \\ \delta \frac{1}{NP} \sum_{i=1}^N \sum_{j=1}^P \left| G(\mathbf{u}^{(i)}, \theta^{(i)}) (\mathbf{y}_{uj}^{(i)}) - \widehat{G}(\mathbf{u}^{(i)}, \theta^{(i)}) (\mathbf{y}_{uj}^{(i)}) \right| - \frac{1}{2} \delta^2 & \text{otherwise} \end{cases} \quad (3)$$

where  $\delta$  is a hyper-parameter,  $\widehat{G}$  is the true solution operator, and  $\widehat{\mathbf{s}} = \widehat{G}(\mathbf{u}, \theta)(\mathbf{y})$  is the true solution. Huber loss combines the excellent performance of both mean squared error (MSE) loss and mean absolute error (MAE) loss to achieve fast convergence speed and good robustness [43]. The average value of the absolute value of absolute error is selected as the performance metric for the test dataset:

$$\varepsilon_{\text{test}} = \frac{1}{N} \sum_{i=1}^N |s_i - \hat{s}_i| \quad (4)$$



**Fig. 1.** Extending the domain geometries of a DeepONet from a simple bulk phase to a complex porous structure. (a) The Porous-DeepONet architecture consists of a trunk net and one or more branch nets. The branch1 net is used for extracting latent representations of the structure of porous media  $\mathbf{u}(x, y)$  and the branch2 net for latent representations of other parameters  $\theta$  in parametric PDEs. Notably, a CNN is introduced as the branch1 net. The role of the trunk net is to extract the latent representation of the input space-time coordinates  $(t, x, y)$  at which the output functions  $G(\mathbf{u}, \theta)$  are evaluated. Continuous and differentiable output functions  $G(\mathbf{u}, \theta)(t, x, y)$  are then obtained by dot-producing the outputs of each subnetwork. (b) The DeepM&Mnet [31] architecture and (c) physics-informed (PI)-DeepONet [11] architecture are employed in porous media, where the subnetworks are the Porous-DeepONets proposed in this work rather than DeepONets [23]. The meaning of each symbol is detailed in the corresponding subsection.

where  $s_i$  is the result predicted by the Porous-DeepONet and  $\hat{s}_i$  is the real result calculated by the FEM. The Porous-DeepONet framework was implemented using the Python package PyTorch. The Porous-DeepONet training and the results prediction were performed using an NVIDIA GeForce RTX 3090 graphic processing unit (GPU) and an Intel® Xeon® Platinum 9242 central processing unit (CPU), respectively. These hardware resources accelerated the computational processes involved in the training and inference tasks.

## 2.2. Coupling the Porous-DeepONet with other frameworks

### 2.2.1. The Porous-DeepM&Mnet

While the Porous-DeepONet excels in learning solution operators for individual physical fields, the coupling between these fields can sometimes impact prediction accuracy. The integration of the Porous-DeepONet with the DeepM&Mnet, as proposed by Cai et al. [31], offers a powerful framework for accurately solving multi-physics PDEs in porous media. The DeepM&Mnet addresses this challenge by coupling multiple pretrained DeepONets to infer electroconvection multi-physical fields effectively. In the combined framework, depicted in Fig. 1(b) [31], we leverage both the Porous-DeepONet and the DeepM&Mnet to handle the interactions between different physical fields (e.g.,  $s_1$ ,  $s_2$ , and  $s_3$ ) in porous media. To account for the influence of  $s_2$  and  $s_3$  on  $s_1$ , these fields are treated as input parameters  $\theta$  in the prediction process, namely  $s_1 = \theta$ . The same approach is applied to  $s_2$  and  $s_3$ . One limitation arises when using a pretrained Porous-DeepONet for  $s_1$ , as it requires the availability of  $s_2$  and  $s_3$ , which may not always be feasible in practice. To address this issue, the Porous-DeepM&Mnet introduces an NN to approximate  $s_1$ ,  $s_2$ , and  $s_3$  simultaneously, while leveraging the pretrained Porous-DeepONets as constraints during the solution process. Finally, calculate the difference between the output of these pretrained Porous-DeepONets (e.g.,  $s_1^*$ ,  $s_2^*$ , and  $s_3^*$ ) and the true value as a loss for training. This hybrid approach ensures accurate predictions while incorporating the knowledge learned from the pretrained models. For more in-depth information on the DeepM&Mnet framework and its applications, readers may refer to the original literature by Cai et al. [31].

### 2.2.2. The Porous-physics-informed (PI)-DeepONet

When dealing with complex porous media, the number of samples required to represent the function space in a DeepONet can be substantial, leading to increased memory requirements for training. To address this issue and reduce the demand for data, Wang et al. [11] introduced a PI-DeepONet framework, as depicted in Fig. 1(c). The PI-DeepONet incorporates the principles of physics-based constraints into the training process, enabling the network to learn the underlying physical laws and reducing the reliance on a large volume of data. In the PI-DeepONet framework, the residuals of the governing equation and BCs and ICs are formulated as regularized losses or constraints using automatic differentiation techniques. This approach not only facilitates the learning of the underlying physics but also mitigates the demand for an extensive amount of data, making it particularly well-suited for scenarios involving complex porous media, where data availability may be limited. Readers interested in delving deeper into the details of the PI-DeepONet framework and its implementation are referred to the original literature by Wang et al. [11] for a comprehensive understanding of this innovative approach.

## 2.3. Data generation

### 2.3.1. Generation and simplification of porous structures

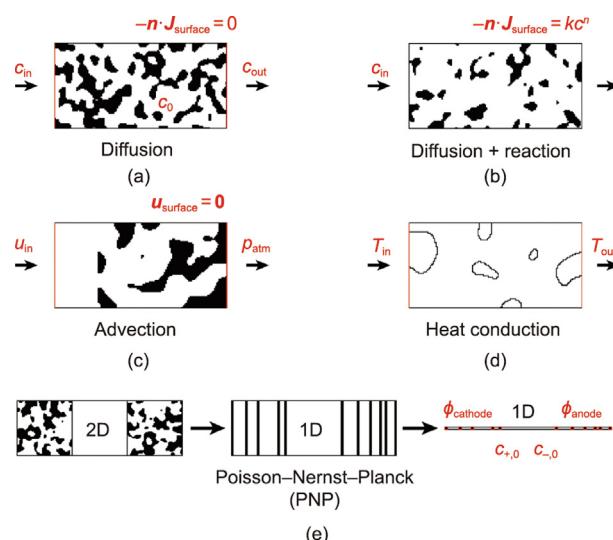
To demonstrate the effectiveness of the Porous-DeepONet in handling complex porous media structures, we selected amorphous porous structures as the model system. While experimental characterization techniques such as computed tomography (CT) and scanning electron microscopy (SEM) can provide nondestructive ways of obtaining such structures [13], the high cost of these methods made generating random structures a more feasible option. To this end, we generated 5000 disordered structures of varying porosity, morphology, and random seed using the Python package PoreSpy [44], as shown in Figs. 2(a)–(d). The setups used to generate these structures are listed in Table S1 in Appendix A. In addition to these randomly generated structures, we considered a simplified porous model called the stacked electrodes model [9,45], which represents the structure of disordered porous electrodes and can be modified by changing the number and position of the plates. Fig. 2(e) shows an example of this model. By considering both random and simplified structures, we demonstrated how the Porous-DeepONet can effectively handle a wide range of porous media structures.

**2.3.2. Numerical details of the FEM**

The governing equations and their associated BCs and ICs were solved using the FEM implemented in COMSOL Multiphysics 5.3. The direct solver employed for the computations was the parallel direct sparse matrix solver (PARDISO), while the fully coupled solver utilized an automatic Newton solver. A relative tolerance factor of  $10^{-3}$  was set for the solvers. In the grid generation process, conventionally sized cells were used to discretize the porous structures, as illustrated in Fig. S1 in Appendix A. The grid scheme for the stacked electrodes model followed the approach detailed in the literature [45]. The FEM computations were carried out on a server equipped with an Intel® Xeon® Platinum 9242 CPU.

## 3. Results and discussion

As described below, we conducted a series of comprehensive numerical studies to demonstrate the effectiveness of the Porous-DeepONet in solving various types of parametric PDEs related to reactive transport processes. Table 1 summarizes the different examples considered, while the associated training details can be found in Section S1 and Tables S2 and S3 in Appendix A.



**Fig. 2.** Simulation domains and corresponding BCs and ICs for different equations, including (a) the Fick diffusion equation, (b) the Fick diffusion and reaction equation, (c) the advection equation, (d) the heat conduction equation, and (e) the PNP equations. Notably, the mass flux, momentum flux, and energy flux are equal to zero in the black region, except for example (e). The meaning of each symbol is detailed in the corresponding subsection.

By evaluating their performance across these diverse examples, we demonstrated the broad applicability of Porous-DeepONets in handling complex PDEs related to porous media structures. This comprehensive approach underscores the potential of the Porous-DeepONet as a powerful tool for advancing our understanding of porous media behavior and improving the design and optimization of related technologies.

### 3.1. The Fick diffusion equation

The first example in this study focuses on showcasing the capability of the Porous-DeepONet to handle the complex structures of porous media. We considered the transient-state Fick diffusion equation within various porous structures in this context. The Fick diffusion equation is commonly used to model the diffusion of molecules, ions, and other substances in porous media such as zeolites, membranes, and cells. Fig. 2(a) illustrates a typical application scenario of the Fick diffusion equation in porous media, such as seawater desalination. In the figure, the white regions represent pores, while the black regions represent solid material. The inlet concentration  $c_{\text{in}}$  and outlet concentration  $c_{\text{out}}$  remain constant over time. In addition, the surface of the porous medium, top boundary, and bottom boundary are impermeable. As time progresses, substances within the porous medium diffuse from regions of high concentration to regions of low concentration until equilibrium is reached. The corresponding transient-state Fick diffusion equation for this scenario can be expressed as follows:

$$\frac{\partial c}{\partial t} = D \nabla^2 c \quad (5)$$

with the following ICs and BCs:

$$c_0 = 1000 \text{ mol} \cdot \text{m}^{-3} \quad (6)$$

$$c_{\text{in}} = 2000 \text{ mol} \cdot \text{m}^{-3}, c_{\text{out}} = 1000 \text{ mol} \cdot \text{m}^{-3}, \quad (7)$$

$$-\mathbf{n} \cdot \mathbf{J}_{\text{surface}/\text{top}/\text{bottom}} = 0 \text{ mol} \cdot (\text{m}^2 \cdot \text{s})^{-1}$$

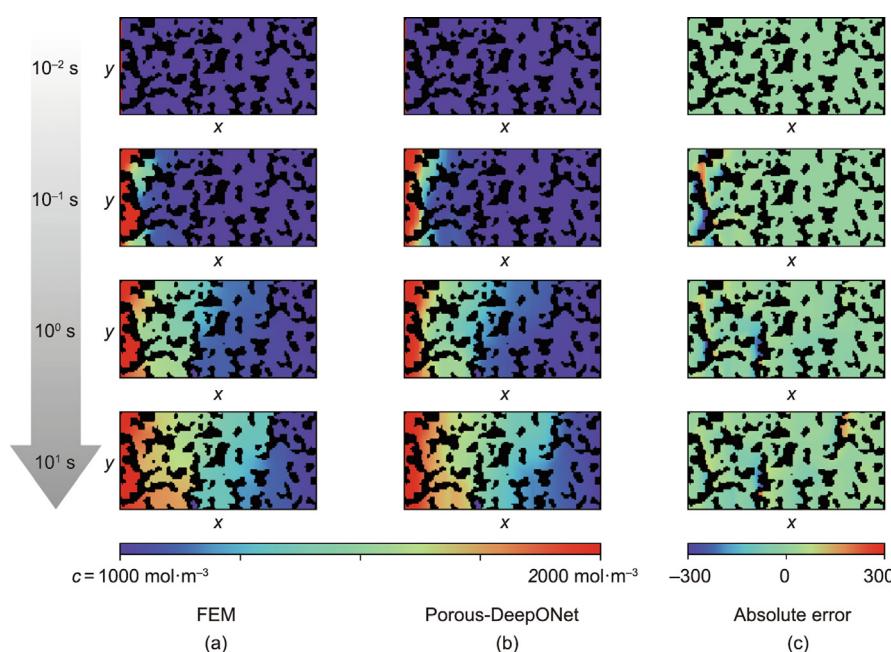
where  $c$  represents the concentration of the diffusing substance;  $D$  is the diffusion coefficient;  $\nabla$  is the Hamiltonian operator;  $c_0$  is the initial concentration;  $\mathbf{n}$  represents the unit normal vector of the surface;  $\mathbf{J}_{\text{surface}}$  denotes the surface molar flux and is calculated as  $\mathbf{J}_{\text{surface}} = -D \nabla c|_{c=c_{\text{surface}}}$ ; and  $\mathbf{J}_{\text{top}}$  and  $\mathbf{J}_{\text{bottom}}$  represent the top and bottom molar flux, respectively. For the specific values of these parameters, please refer to Table S1.

We use the Porous-DeepONet to map the solution operator from a porous structure  $\mathbf{u}(x, y)$  to the full spatiotemporal solution  $c(t, x, y)$  of the Fick diffusion equation. The comparison between the predicted solution by the Porous-DeepONet and the exact solution by the FEM for a random input porous structure in the test dataset shows excellent agreement from the initial state (i.e.,  $10^{-2}$  s) to the equilibrium state (i.e., 10 s), as shown in Fig. 3. The prediction errors are mainly concentrated where the concentration gradient is steep, which aligns with observations by Wang et al. [11]. More visualizations for different input porous structures are provided in Fig. S9 in Appendix A. In addition, the distributions

**Table 1**

Summary of examples studied in this work of Porous-DeepONets across various parametric PDEs. The meaning of each variable is detailed in the corresponding subsection.

PDE	Neural network	Trunk inputs	Branch inputs	Total number of data
Diffusion	Porous-DeepONet	$t, x, y$	5000 structures	216 268 800
Diffusion + reaction	Porous-DeepONet	$x, y$	5000 structures, $k$ and $n$	40 960 000
Advection	Porous-DeepONet	$x, y$	5000 structures and $u_{\text{in}}$	40 960 000
Heat conduction	Porous-DeepONet	$x, y$	5000 structures and $\lambda_2$	40 960 000
PNP	Porous-DeepM&Mnet	$t, x$	5000 structures, $D_+$ and $D_-$	54 774 606



**Fig. 3.** Solving the parametric Fick diffusion equation in real porous structures. (a) The exact solution by FEM; (b) the predicted solution by Porous-DeepONet; (c) the absolute error at  $10^{-2}$ ,  $10^{-1}$ ,  $10^0$ , and  $10^1$  s.

of the absolute error in the training dataset and test dataset indicate that most of the values of absolute errors are within a certain range of  $-25$  to  $25 \text{ mol}\cdot\text{m}^{-3}$ , and the resulting test error  $\varepsilon_{\text{test}}$  is  $6.6 \text{ mol}\cdot\text{m}^{-3}$ , as shown in Fig. S10 in Appendix A. Based on these observations, it can be concluded that Porous-DeepONets are an effective extension to enhance and generalize the capabilities of DeepONets in solving parametric PDEs in porous media.

### 3.2. The Fick diffusion and surface reaction equation

The next example we provided involves a steady-state Fick diffusion and surface reaction equation with various reaction rates, where the surface morphology influences the surface reaction and is more intricate than a homogeneous reaction. Fig. 2(b) illustrates the diffusion of material in a porous medium and its reaction on the surface—a crucial phenomenon in catalysts and adsorbents. In this scenario, we assume that the reaction rate follows a power exponential function form. It is important to note that, to capture the diffusion-reaction process more realistically, the outlet concentration is not fixed. The corresponding equation can be expressed as follows:

$$D\nabla^2 c = 0 \quad (8)$$

with the following BCs:

$$\begin{aligned} c_{\text{in}} &= 100 \text{ mol} \cdot \text{m}^{-3}, \quad \mathbf{n} \cdot D\nabla c_{\text{out}} = 0 \text{ mol} \cdot (\text{m}^2 \cdot \text{s})^{-1}, \\ -\mathbf{n} \cdot \mathbf{J}_{\text{surface}} &= kc^n, \quad -\mathbf{n} \cdot \mathbf{J}_{\text{top/bottom}} = 0 \text{ mol} \cdot (\text{m}^2 \cdot \text{s})^{-1} \end{aligned} \quad (9)$$

where  $k$  represents the reaction rate constant, and  $n$  is the reaction order. By adjusting the values of  $k$  and  $n$ , and the structures of the catalyst or adsorbent, Eqs. (8) and (9) can effectively model the diffusion and reaction phenomena in a wide range of catalysts or adsorbents. The specific values and permissible ranges for these parameters are detailed in Table S1.

The results obtained from applying the Porous-DeepONet to learn the solution operator mapping the porous structure  $\mathbf{u}(x,y)$  and the reaction parameters  $k$  and  $n$  to the solution  $c(x,y)$  are

presented in Fig. 4. The test dataset includes four representative input  $\mathbf{u}(x,y)$  and  $kc^n$ , showcasing a range of reaction depths from high to low, resulting in a gradual increase in the outlet concentration. The comparison between the Porous-DeepONet predictions and the ground truth demonstrates a good agreement across different scenarios, capturing the transition from large reaction rates and complex structures to small reaction rates and simpler structures. However, some inaccuracies can be observed in regions where the concentration exhibits significant gradients, similar to the example of the Fick diffusion discussed earlier.

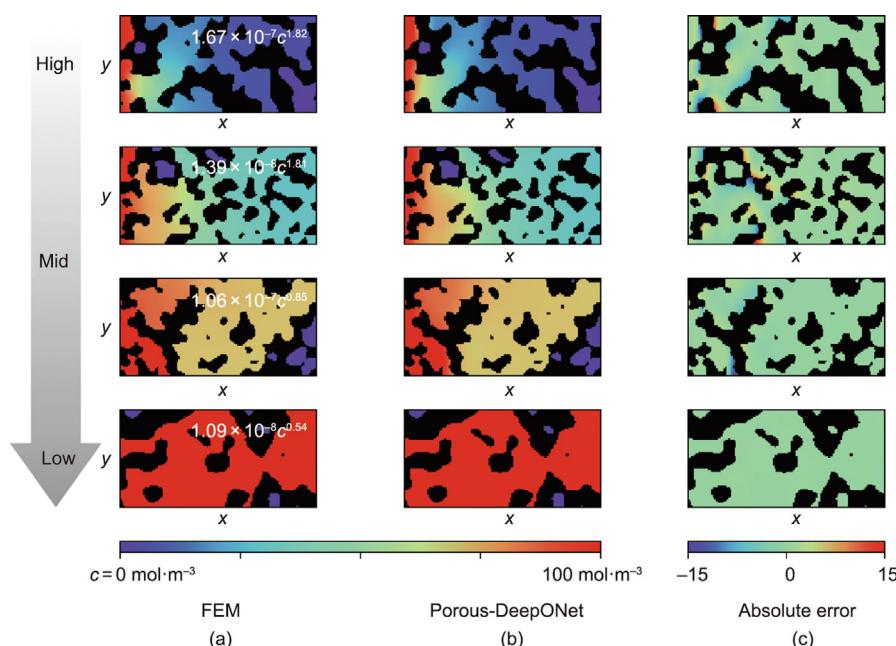
To evaluate the performance of the Porous-DeepONet model, the distributions of the absolute error in the training and testing datasets are illustrated in Fig. S11(a) in Appendix A. The test error  $\varepsilon_{\text{test}}$  is reported to be  $2.4 \text{ mol}\cdot\text{m}^{-3}$ , indicating that the Porous-DeepONet can serve as an effective surrogate for PDEs with complex and variable flux BCs. This finding highlights the capability of the Porous-DeepONet to accurately capture the behavior of the system under diverse and challenging conditions, demonstrating its potential for practical applications in modeling complex porous media with varying reaction parameters and BCs.

### 3.3. Advection equation

To further demonstrate the ability of the Porous-DeepONet to handle boundary layers, the advection equation is considered, as shown in Fig. 2(c). In this scenario, the inlet velocity differs across different structures, and the outlet pressure is set to atmospheric pressure. The surface of the porous media is subjected to a no-slip BC. Moreover, to accurately simulate the change in velocity at the entrance, it is assumed that the porous medium is connected to the bulk phase. The corresponding advection equation is as follows:

$$\rho(\mathbf{u} \cdot \nabla) \mathbf{u} = \nabla \cdot [-\rho \mathbf{I} + \mathbf{K}] \quad (10)$$

$$\rho \nabla \cdot (\mathbf{u}) = 0 \quad (11)$$



**Fig. 4.** Solving the parametric Fick diffusion and surface reaction equation in real porous structures. (a) The exact solution by FEM; (b) the predicted solution by Porous-DeepONet; (c) the absolute error from high, to middle, to low reaction rate. The formulas in the subfigures are the reaction rate  $kc^n$ , where  $k$  represents the reaction rate constant,  $c$  denotes the reactant concentration, and  $n$  is the reaction order.

$$\mathbf{K} = \mu(\nabla \mathbf{u} + (\nabla \mathbf{u})^T) \quad (12)$$

with the following BCs:

$$\mathbf{u}_{in} = -\mathbf{u}_{in}\mathbf{n}, p_{out} = p_{atm}, \mathbf{u}_{surface/top/bottom} = \mathbf{0} \quad (13)$$

where  $\mathbf{u}$  is the velocity,  $u$  is the magnitude of the velocity,  $\rho$  is the mass density,  $p$  is the pressure,  $\mathbf{I}$  is a unit matrix,  $\mathbf{K}$  is the viscous resistance, and  $\mu$  is the viscosity. Eqs. (10)–(13) likely capture the complex phenomena associated with mass transport in these systems. These types of equations are crucial for understanding and optimizing the performance of fuel cells and reaction towers. The values and ranges of the parameters involved in these equations are listed in Table S1, providing important information for characterizing the system's behavior and potentially enabling further analysis and modeling.

The Porous-DeepONet is employed to predict the magnitude of the velocity  $u(x, y)$  by inputting porous structure  $\mathbf{u}(x, y)$  and the magnitude of the inlet velocity  $u_{in}$ . As depicted in Fig. 5, the results demonstrate that the trained Porous-DeepONet exhibits an overall excellent agreement with the reference PDE solution, even though the velocity is not zero at certain surface parts. The distributions of the absolute error further indicate that the test error  $\varepsilon_{test}$  amounts to  $0.0012 \text{ m}\cdot\text{s}^{-1}$ , as illustrated in Fig. S11(b) in Appendix A. This finding underscores the capability of the Porous-DeepONet to effectively solve parametric PDEs with BCs related not only to flux but also to variable parameters such as  $u$ .

### 3.4. Heat conduction equation

This example aims to explore the potential of the Porous-DeepONet in solving multiphase parametric PDEs, as illustrated in Fig. 2(d). In contrast to the previous examples, let us consider a scenario in which heat can be transferred between two phases. More specifically, we assume that there is no thermal resistance between the two phases in porous media when the inlet temperature is higher than the outlet temperature. To clearly differentiate between the two phases, we designate phase 1 as air with a low

thermal conductivity and phase 2 as metal with a high thermal conductivity. The corresponding equation is provided below:

$$\lambda_1 \nabla^2 T = 0, \lambda_2 \nabla^2 T = 0 \quad (14)$$

with the following BCs:

$$T_{in} = 313.15 \text{ K}, T_{out} = 293.15 \text{ K}, -\mathbf{n} \cdot \mathbf{Q}_{top/bottom} = 0 \text{ W}\cdot\text{m}^{-2} \quad (15)$$

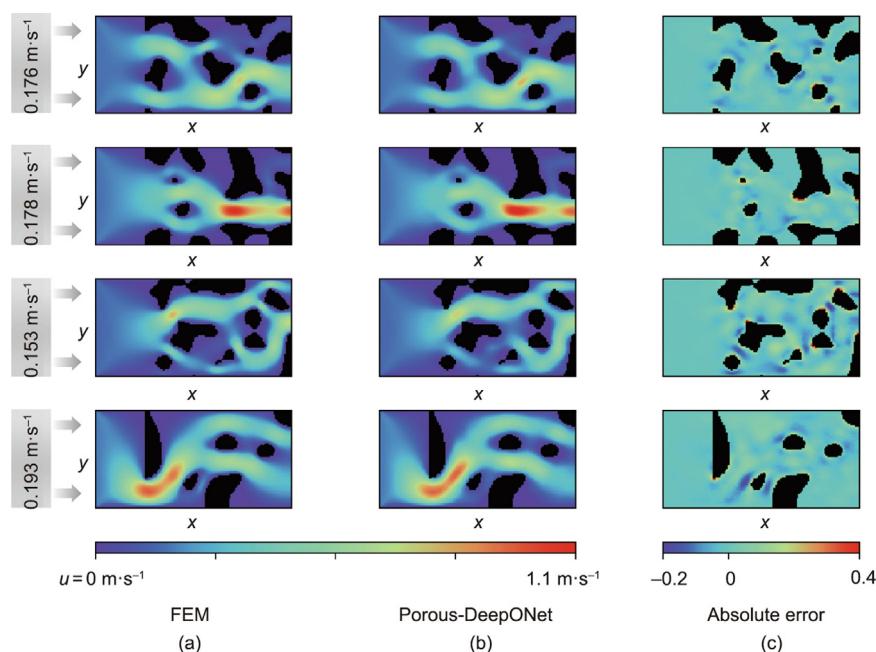
$$(-\mathbf{n} \cdot \mathbf{Q}_{surface,1}) + (-\mathbf{n} \cdot \mathbf{Q}_{surface,2}) = 0 \text{ W}\cdot\text{m}^{-2}, T_{surface,1} = T_{surface,2} \quad (16)$$

where  $\lambda$  is the heat conductivity,  $T$  is temperature, and  $\mathbf{Q}$  is the heat flux, given by  $\mathbf{Q} = -\lambda \nabla T$ . The specific values and permissible ranges of the parameters in Eqs. (14)–(16) are provided in Table S1.

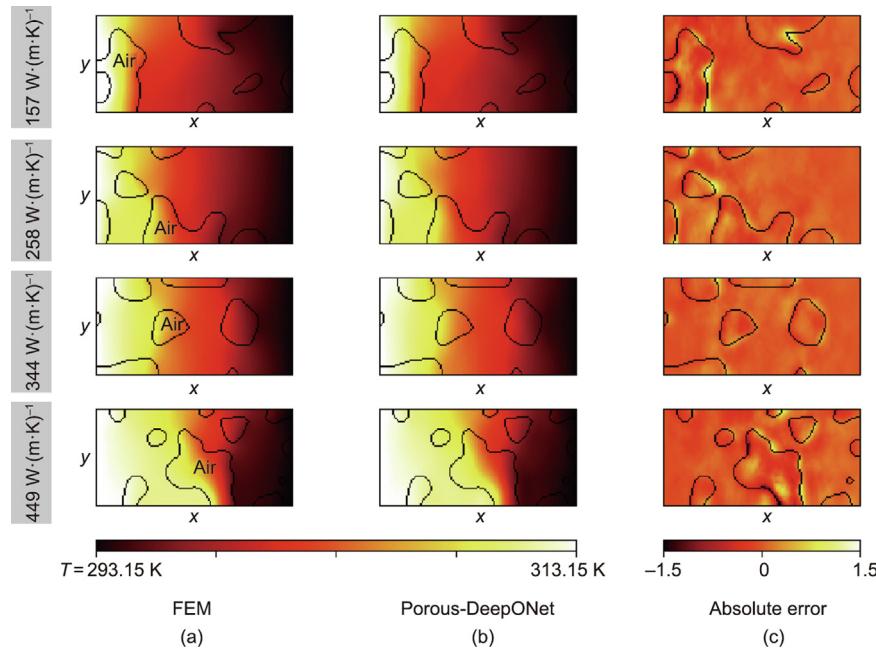
Fig. 6 demonstrates that the trained Porous-DeepONet model can predict the solution of the heat conduction equation  $T(x, y)$  for any given heat conductivity of phase 2  $\lambda_2$  and porous structure  $\mathbf{u}(x, y)$  in the test dataset. The average of the absolute value of the absolute error is found to be  $0.020 \text{ K}$ , as detailed in Fig. S11(c) in Appendix A. Interestingly, unlike the single-phase examples discussed earlier, the maximum error does not occur where the gradient is the largest but rather occurs on the surface. This phenomenon suggests that, when learning the solution of parametric PDEs in porous structures, special attention needs to be given to regions with steep gradients and the surface of porous structures. These findings highlight the importance of accurately modeling and capturing the behavior of the surface and regions with significant gradients when using Porous-DeepONets to solve PDEs in porous structures.

### 3.5. The PNP equations

The last example aims to solve multi-physics parametric PDEs in porous media using the Porous-DeepM&Mnet. In addition, the possibility of combining the Porous-DeepM&Mnet with other porous models is explored by replacing real porous structures with the porous model. Considering the representativeness of the model



**Fig. 5.** Solving the parametric advection equation in real porous structures. (a) The exact solution by FEM; (b) the predicted solution by Porous-DeepONet; (c) the absolute error at the inlet velocity of  $0.176, 0.178, 0.153$ , and  $0.193 \text{ m}\cdot\text{s}^{-1}$ .



**Fig. 6.** Solving the parametric heat conduction equation in real porous structures. (a) The exact solution by FEM; (b) the predicted solution by Porous-DeepONet; (c) the absolute error at the heat conductivity of non-air phase of 157, 258, 344, and 449 W·(m·K)<sup>-1</sup>.

and the complexity of the PDEs, we chose the stack electrode model [9,45] and the PNP equations to investigate the charging process in porous electrodes. The simplification process of real porous electrodes is shown in Fig. 2(e). The simplified porous structure can be regarded as a 1D image, which is easily inputted into the Porous-DeepONet and Porous-DeepM&Mnet. After the potential is applied to the cathode and anode, the anions and cations diffuse from the center of the two plates and the bulk phase to the vicinity of the plate. They then adsorb onto the plate to form electric double layers (EDLs). The following PDEs describe this phenomenon:

$$\frac{\partial c_{\pm}}{\partial t} + \nabla \cdot \mathbf{J}_{\pm} = 0 \quad (17)$$

$$\mathbf{J}_{\pm} = -D_{\pm} \left( \nabla c_{\pm} + \frac{z_{\pm} F c_{\pm}}{RT} \nabla \phi \right) \quad (18)$$

$$\epsilon_0 \epsilon_r \nabla \cdot (-\nabla \phi) = Fq \quad (19)$$

with the following ICs and BCs:

$$c_{+,0} = c_{-,0} = 100 \text{ mol} \cdot \text{m}^{-3} \quad (20)$$

$$\phi_{\text{cathode}} = 0.01 \text{ V}, \phi_{\text{anode}} = -0.01 \text{ V}, -\mathbf{n} \cdot \mathbf{J}_{\text{left/right}} = 0 \quad (21)$$

where  $\pm$  represents the cation and anion, respectively; while  $z$  denotes the valence of the ion. In addition,  $\phi$  represents the potential,  $q$  corresponds to the space charge concentration (given by  $q = z_{+}c_{+} + z_{-}c_{-}$ ),  $R$  is the ideal gas constant and  $F$  represents the Faraday constant.  $\epsilon_0$  and  $\epsilon_r$  denote the vacuum and relative dielectric constant, respectively. It is important to note that varying the number and position of plates generates different porous structures. Furthermore,  $D_{+}$  and  $D_{-}$  are variables used to represent various combinations of anions and cations. More detailed information is given in Table S1.

Fig. 7 presents the results of  $\phi$ ,  $c_{+}$ , and  $c_{-}$  predicted by the Porous-DeepM&Mnet, which is based on the Porous-DeepONet. The displayed results are taken from a randomly selected entry in the test dataset (further visualizations can be found in Fig. S12 in Appendix A). These results demonstrate that the Porous-DeepM&Mnet effectively captures the details of the EDLs adsorbed

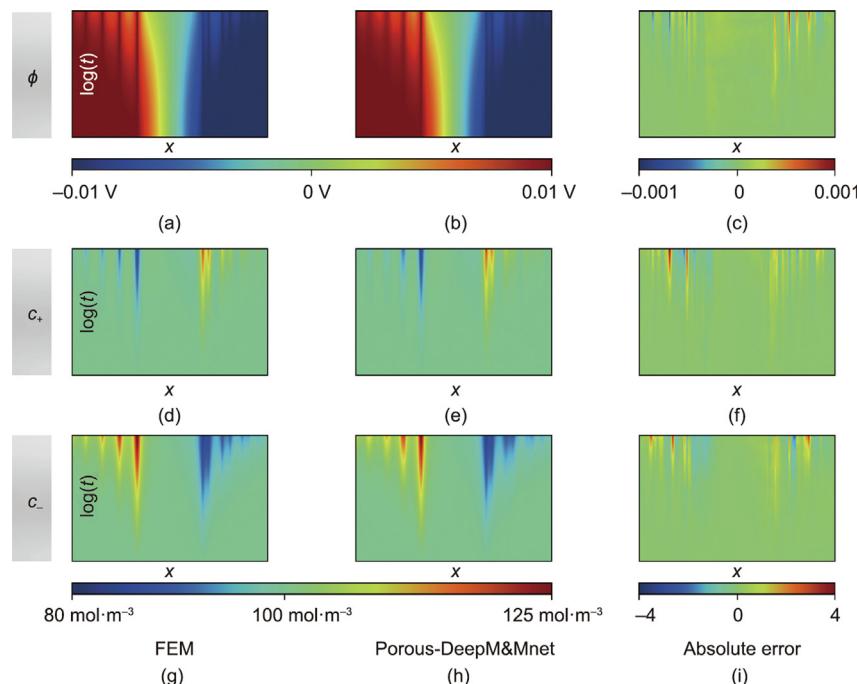
on the electrode plate. The test errors  $\varepsilon_{\text{test}}$  for  $\phi$ ,  $c_{+}$ , and  $c_{-}$  are  $8.1 \times 10^{-6} \text{ V}$ ,  $0.015 \text{ mol} \cdot \text{m}^{-3}$ , and  $0.014 \text{ mol} \cdot \text{m}^{-3}$ , respectively, as can be easily observed in Figs. S11(d)–(f) in Appendix A. Therefore, the Porous-DeepONet can seamlessly integrate with other DeepONet frameworks to solve more complex parametric PDEs.

### 3.6. Comparison of the computation time between the Porous-DeepONet and FEM

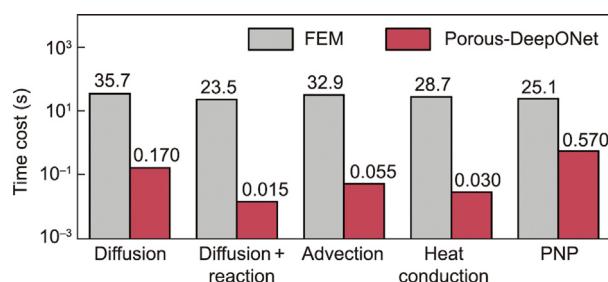
To demonstrate the high solving efficiency of the Porous-DeepONet, we measured the time required for the the Porous-DeepONet and the FEM to solve the same problem on an Intel® Xeon® Platinum 9242 CPU with a single CPU core, as shown in Fig. 8. The time cost of the trained Porous-DeepONet includes loading the trained model and inferring the results, whereas the time cost of the FEM encompasses importing the porous structure, meshing, assembling the matrix, and solving. It was found that the Porous-DeepONet was up to three orders of magnitude  $\mathcal{O}(10^3)$  faster than the conventional FEM in solving the same problem. When the Porous-DeepM&Mnet was based on the Porous-DeepONet, the solution speed improved by about 50 times when solving the PNP equations. It should be noted that the first and last examples took significantly more time than the other examples because they consider time  $t$ , which requires a greater number of inferences (Table 1). To reduce the inference time, the Porous-DeepONet can be used to infer results in parallel on a CPU or GPU.

## 4. Conclusions

This work introduces the Porous-DeepONet, a deep-learning framework designed for learning the solution operators of parametric PDEs in porous media, with a focus on reactive transport equations. We employed CNNs to extend the DeepONet, resulting in a simple yet highly effective approach capable of handling complex domain geometries. This extension enables significant improvements in solving efficiency compared with conventional FEM methods. To evaluate the accuracy and applicability of the Porous-DeepONet, we solved various reactive transport equations,



**Fig. 7.** Solving the PNP equations in a porous model. The exact solution by FEM, the predicted solution by Porous-DeepM&Mnet, and the absolute error of (a–c) the potential  $\phi$ ; (d–f) the concentration of cation  $c_+$ ; and (g–i) the concentration of anion  $c_-$ , respectively.



**Fig. 8.** Comparison between FEM and Porous-DeepONet regarding the average time cost of solving the Fick diffusion equation, the Fick diffusion and surface reaction equation, the advection equation, the heat conduction equation, and the PNP equations.

including the Fick diffusion equation, the Fick diffusion and surface reaction equation, the advection equation, and the heat conduction equation. The results demonstrated that the Porous-DeepONet can efficiently solve single-phase and multiphase parametric PDEs with complex BCs, achieving up to three orders of magnitude faster computation times than the conventional FEM. We also combined the Porous-DeepONet with the DeepM&Mnet to tackle the challenge of solving multi-physical field coupled PNP equations, leading to a substantial reduction in computation time by a factor of 50.

In addition to the examples mentioned above, the Porous-DeepONet has a wide range of applications in various scientific and engineering fields involving porous media, including quantum chemistry and social sciences. Despite the promising initial results, several technical challenges require further investigation. Firstly, to address inaccuracies in step gradient regions and near surfaces, integrating a self-attention mechanism or utilizing data-augmentation techniques in the Porous-DeepONet could be beneficial. Secondly, the requirement of numerous paired input–output observations for the Porous-DeepONet may be costly. An alternative approach could involve transitioning the Porous-DeepONet from being data-driven to being driven by physical information, as demonstrated by the PI-DeepONet [11]. However, due to the

complexity of BCs in porous media and challenges with convergence in the loss function of PINNs, exploring the Porous-PI-DeepONet remains a valuable research direction. Thirdly, the Porous-DeepONet is currently limited to solving parametric PDEs in fixed-size porous structures. To enhance its adaptability to more complex input porous structures (e.g., random-size SEM images post-segmentation from databases), advanced architectures such as fully CNNs need to be integrated into the Porous-DeepONet. Lastly, due to limited video memory size and training time constraints, the Porous-DeepONet encounters limitations when handling super-large 3D structures with an RTX 3090 graphics card. Drawing inspiration from examples like the PNP equations, simplified porous models such as the stacked electrodes model [45] and pore network model [46] can serve as alternative inputs to the Porous-DeepONet. Addressing these technical challenges not only has the potential to improve the performance of the Porous-DeepONet but also positions it as a viable alternative to the FEM for solving PDEs in diverse structures.

## Acknowledgments

This work was supported by the National Key Research and Development Program of China (2022YFA1503501), the National Natural Science Foundation of China (22378112, 22278127, and 22078088), the Fundamental Research Funds for the Central Universities (2022ZFJH004), and the Shanghai Rising-Star Program (21QA1401900).

## Compliance with ethics guidelines

Pan Huang, Yifei Leng, Cheng Lian, and Honglai Liu declare that they have no conflict of interest or financial conflicts to disclose.

## Appendix A. Supplementary data

Supplementary data to this article can be found online at <https://doi.org/10.1016/j.eng.2024.07.002>.

## References

- [1] Cai G, Yan P, Zhang L, Zhou HC, Jiang HL. Metal-organic framework-based hierarchically porous materials: synthesis and applications. *Chem Rev* 2021;121(20):12278–326.
- [2] Sun MH, Huang SZ, Chen LH, Li Y, Yang XY, Yuan ZY, et al. Applications of hierarchically structured porous materials from energy storage and conversion, catalysis, photocatalysis, adsorption, separation, and sensing to biomedicine. *Chem Soc Rev* 2016;45(12):3479–563.
- [3] Zhou J, Wang B. Emerging crystalline porous materials as a multifunctional platform for electrochemical energy storage. *Chem Soc Rev* 2017;46(22):6927–45.
- [4] Perego C, Millini R. Porous materials in catalysis: challenges for mesoporous materials. *Chem Soc Rev* 2013;42(9):3956–76.
- [5] Wang X, Chen L, Chong SY, Little MA, Wu Y, Zhu WH, et al. Sulfone-containing covalent organic frameworks for photocatalytic hydrogen evolution from water. *Nat Chem* 2018;10(12):1180–9.
- [6] Yuk H, Zhang T, Lin S, Parada GA, Zhao X. Tough bonding of hydrogels to diverse non-porous surfaces. *Nat Mater* 2016;15(2):190–6.
- [7] Pan H, Shao Y, Yan P, Cheng Y, Han KS, Nie Z, et al. Reversible aqueous zinc/manganese oxide energy storage from conversion reactions. *Nat Energy* 2016;1(5):16039.
- [8] Tian S, Wang B, Gong W, He Z, Xu Q, Chen W, et al. Dual-atom Pt heterogeneous catalyst with excellent catalytic performances for the selective hydrogenation and epoxidation. *Nat Commun* 2021;12(1):3181.
- [9] Lian C, Janssen M, Liu H, van Roij R. Blessing and curse: how a supercapacitor's large capacitance causes its slow charging. *Phys Rev Lett* 2020;124(7):076001.
- [10] Tao H, Lian C, Liu H. Multiscale modeling of electrolytes in porous electrode: from equilibrium structure to non-equilibrium transport. *Green Energy Environ* 2020;5(3):303–21.
- [11] Wang S, Wang H, Perdikaris P. Learning the solution operator of parametric partial differential equations with physics-informed DeepONets. *Sci Adv* 2021;7(40):eabi8605.
- [12] Lu L, Meng X, Mao Z, Karniadakis GE. DeepXDE: a deep learning library for solving differential equations. *SIAM Rev* 2021;63(1):208–28.
- [13] Bostanabad R, Zhang Y, Li X, Kearney T, Brinson LC, Apley DW, et al. Computational microstructure characterization and reconstruction: review of the state-of-the-art techniques. *Prog Mater Sci* 2018;95:1–41.
- [14] Quarteroni A, Rozza G, Manzoni A. Certified reduced basis approximation for parametrized partial differential equations and applications. *J Math Ind* 2011;1(1):3.
- [15] Yunus RB, Abdul Karim SA, Shafie A, Izzatullah M, Kherd A, Hasan MK, et al. An overview on deep learning techniques in solving partial differential equations. In: Abdul Karim SA, editor. Intelligent systems modeling and simulation II: machine learning, neural networks, efficient numerical algorithm and statistical methods. Cham: Springer International Publishing; 2022. p. 37–47.
- [16] Karniadakis GE, Kevrekidis IG, Lu L, Perdikaris P, Wang S, Yang L. Physics-informed machine learning. *Nat Rev Phys* 2021;3(6):422–40.
- [17] Ren P, Rao C, Liu Y, Wang JX, Sun H. PhyCRNet: physics-informed convolutional-recurrent network for solving spatiotemporal PDEs. *Comput Methods Appl Mech Eng* 2022;389:114399.
- [18] Winovich N, Ramani K, Lin G. ConvPDE-UQ: convolutional neural networks with quantified uncertainty for heterogeneous elliptic partial differential equations on varied domains. *J Comput Phys* 2019;394:263–79.
- [19] Ruthotto L, Haber E. Deep neural networks motivated by partial differential equations. *J Math Imaging Vis* 2020;62(3):352–64.
- [20] Gao H, Sun L, Wang JX. PhyGeoNet: physics-informed geometry-adaptive convolutional neural networks for solving parameterized steady-state PDEs on irregular domain. *J Comput Phys* 2021;428:110079.
- [21] Beck C, Hutzenthaler M, Jentzen A, Kuckuck B. An overview on deep learning-based approximation methods for partial differential equations. *Discrete Contin Dyn Syst B* 2023;28(6):3697–746.
- [22] Chen T, Chen H. Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. *IEEE Trans Neural Netw* 1995;6(4):911–7.
- [23] Lu L, Jin P, Pang G, Zhang Z, Karniadakis GE. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nat Mach Intell* 2021;3(3):218–29.
- [24] Sharma Priyadarshini M, Venturi S, Zanardi I, Panesi M. Efficient quasi-classical trajectory calculations by means of neural operator architectures. *Phys Chem Chem Phys* 2023;25(20):13902–12.
- [25] Lin C, Maxey M, Li Z, Karniadakis GE. A seamless multiscale operator neural network for inferring bubble dynamics. *J Fluid Mech* 2021;929:A18.
- [26] Lin C, Li Z, Lu L, Cai S, Maxey M, Karniadakis GE. Operator learning for predicting multiscale bubble growth dynamics. *J Chem Phys* 2021;154(10):104118.
- [27] Zheng Q, Yin X, Zhang D. Inferring electrochemical performance and parameters of Li-ion batteries based on deep operator networks. *J Energy Storage* 2023;65:107176.
- [28] Moya C, Zhang S, Lin G, Yue M. DeepONet-grid-UQ: a trustworthy deep operator framework for predicting the power grid's post-fault trajectories. *Neurocomputing* 2023;535:166–82.
- [29] Yin M, Ban E, Rego BV, Zhang E, Cavinato C, Humphrey JD, et al. Simulating progressive intramural damage leading to aortic dissection using DeepONet: an operator-regression neural network. *J R Soc Interface* 2022;19(187):20210670.
- [30] Pickering E, Guth S, Karniadakis GE, Sapsis TP. Discovering and forecasting extreme events via active learning in neural operators. *Nat Comput Sci* 2022;2(12):823–33.
- [31] Cai S, Wang Z, Lu L, Zaki TA, Karniadakis GE. DeepM&Mnet: inferring the electroconvection multiphysics fields based on operator approximation by neural networks. *J Comput Phys* 2021;436:110296.
- [32] Yin M, Zhang E, Yu Y, Karniadakis GE. Interfacing finite elements with deep neural operators for fast multiscale modeling of mechanics problems. *Comput Methods Appl Mech Eng* 2022;402:115027.
- [33] Lu L, Pestourie R, Johnson SG, Romano G. Multifidelity deep neural operators for efficient learning of partial differential equations with application to fast inverse design of nanoscale heat transport. *Phys Rev Res* 2022;4(2):023210.
- [34] Lu L, Meng X, Cai S, Mao Z, Goswami S, Zhang Z, et al. A comprehensive and fair comparison of two neural operators (with practical extensions) based on FAIR data. *Comput Methods Appl Mech Eng* 2022;393:114778.
- [35] Li W, Bazant MZ, Zhu J. Phase-field DeepONet: physics-informed deep operator neural network for fast simulations of pattern formation governed by gradient flows of free-energy functionals. *Comput Methods Appl Mech Eng* 2023;416:116299.
- [36] Goswami S, Kontolati K, Shields MD, Karniadakis GE. Deep transfer operator learning for partial differential equations under conditional shift. *Nat Mach Intell* 2022;4(12):1155–64.
- [37] Goswami S, Bora A, Yu Y, Karniadakis GE. Physics-informed deep neural operator networks. In: Rabczuk T, Bathe KJ, editors. Machine learning in modeling and simulation: methods and applications. Cham: Springer; 2023. p. 219–54.
- [38] Sun J, Li J, Hao Y, Qi C, Ma C, Sun H, et al. Boundary-to-solution mapping for groundwater flows in a Toth basin. *Adv Water Resour* 2023;176:104448.
- [39] Jablonka KM, Ongari D, Moosavi SM, Smit B. Big-data science in porous materials: materials genomics and machine learning. *Chem Rev* 2020;120(16):8066–129.
- [40] Cawte T, Bazylak A. A 3D convolutional neural network accurately predicts the permeability of gas diffusion layer materials directly from image data. *Curr Opin Electrochem* 2022;35:101101.
- [41] Wang Y, Arns CH, Rahman SS, Arns JY. Porous structure reconstruction using convolutional neural networks. *Math Geosci* 2018;50(7):781–99.
- [42] Kontolati K, Goswami S, Shields MD, Karniadakis GE. On the influence of over-parameterization in manifold based surrogates and deep neural operators. *J Comput Phys* 2023;479:112008.
- [43] Sun Q, Zhou WX, Fan J. Adaptive huber regression. *J Am Stat Assoc* 2020;115(529):254–65.
- [44] Gostick JT, Khan ZA, Tranter TG, Kok MDR, Agnaou M, Sadeghi M, et al. PoreSpy: a Python toolkit for quantitative analysis of porous media images. *J Open Source Softw* 2019;4(37):1296.
- [45] Huang P, Tao H, Yang J, Lian C, Liu H. Four stages of thermal effect coupled with ion-charge transports during the charging process of porous electrodes. *AIChE J* 2022;68(10):e17790.
- [46] Huang P, Tao H, Liu H, Lian C. Accelerating charging dynamics using self-driven optimizing porous structures. *AIChE J* 2024;70(4):e18313.