ARTICLE OPEN

Check for updates

Simulating fluid flow in complex porous materials by integrating the governing equations with deep-layered machines

Serveh Kamrava[1], Muhammad Sahimi[1]✉ and Pejman Tahmasebi[2]

Fluid flow in heterogeneous porous media arises in many systems, from biological tissues to composite materials, soil, wood, and paper. With advances in instrumentations, high-resolution images of porous media can be obtained and used directly in the simulation of fluid flow. The computations are, however, highly intensive. Although machine learning (ML) algorithms have been used for predicting flow properties of porous media, they lack a rigorous, physics-based foundation and rely on correlations. We introduce an ML approach that incorporates mass conservation and the Navier–Stokes equations in its learning process. By training the algorithm to relatively limited data obtained from the solutions of the equations over a time interval, we show that the approach provides highly accurate predictions for the flow properties of porous media at all other times and spatial locations, while reducing the computation time. We also show that when the network is used for a different porous medium, it again provides very accurate predictions.

npj Computational Materials (2021)7:127 ; https://doi.org/10.1038/s41524-021-00598-2

INTRODUCTION

Fluid flow and transport in heterogeneous porous media are of fundamental importance to the working of a wide variety of systems of scientific interest, as well as applications[1,2]. Examples of such porous media include catalysts, membranes, filters, adsorbents, print paper, wood, nanostructured materials, and biological tissues, as well as soil and pavement, and oil, gas, and geothermal reservoirs. Such porous media are typically heterogeneous, with the heterogeneity manifesting itself in the shape, size, connectivity, and surface structure of the pores at small scales, and in the spatial variations of the porosity, permeability, and the elastic moduli at large length scales. Thus, any attempt to model flow and transport in porous media entails having the ability to handle big data that is used[3] in computing the spatial distribution of the pressure, fluid velocity, etc., throughout the pore space. The heterogeneity and the associated big data, contained in high-resolution two- or three-dimensional (2D or 3D) images of porous media used in their modeling imply that the calculations are highly intensive. Thus, developing efficient predictive algorithms has always been an active area of research.

Significant advancements have been made over the last decade in the development of deep-learning approaches that have made considerable contributions to progress in image analysis, which is also important to studying fluid flow and transport in porous media[4–6], and other fields. The traditional neural networks, developed to handle large data, have great potential as approximators. Feed-forward neural networks (FFNNs), representing supervised learning methods, try to identify the relationship between the input and output iteratively by minimizing a cost function. To alleviate the computational burden associated with the minimization, advanced optimization methods have been developed[7]. There is, however, no systematic approach for increasing the accuracy of fully connected FFNNs, as they rely on correlations between data and the properties to be predicted, hence requiring a large amount of training data for acceptable accuracy. Thus, it is most desirable to develop alternatives.

Progress has been made very recently to develop such alternatives, commonly referred to as physics-informed machine learning (ML) in which the network is trained partly based on the fundamental equations that govern the physics of the process under study. Thus, by incorporating the equations in the cost function, one speeds up the convergence and produces accurate predictions, whereas the network requires training with much less data. The idea of using the equations that govern the physics of fluid flow and transport in porous media in the cost function and optimization was first proposed by Sahimi and colleagues[8,9]. Development of such techniques based on the ML methods was first reported for solving differential[10] and partial differential equations[11–16], and has been recently proposed for analyzing hydrodynamic systems[17].

In this study, we aim to fill the gap between the ML methods and direct numerical simulation of fluid flow in a porous medium by combining the advantages of both approaches to obtain accurate solutions of the pressure and fluid velocity fields efficiently. The current ML methods are accurate estimator techniques when they have been sufficiently trained with a large number of datasets of enough variety. Multiple studies have used physics-informed FFNNs for a variety of purposes. However, as our goal is making predictions for fluid flow in images of porous media, it is not practical to use fully connected neural networks, due to the complexity of the images and low efficiency of such networks in discovering complex patterns.

To address the issue, we propose an approach based on an ML method that incorporates in its training (in the cost function) the governing equations for fluid flow in porous media, i.e., the mass conservation (MC) and the Navier–Stokes equations (NS). The result is a highly efficient method for predicting the flow properties. Our multiphysics approach integrates the MC and NS equations, and digital images of heterogeneous pore space with

[1]Mork Family Department of Chemical Engineering and Materials Science, University of Southern California, Los Angeles, CA, USA. [2]College of Engineering and Applied Science, University of Wyoming, Laramie, WY, USA. ✉email: moe@usc.edu

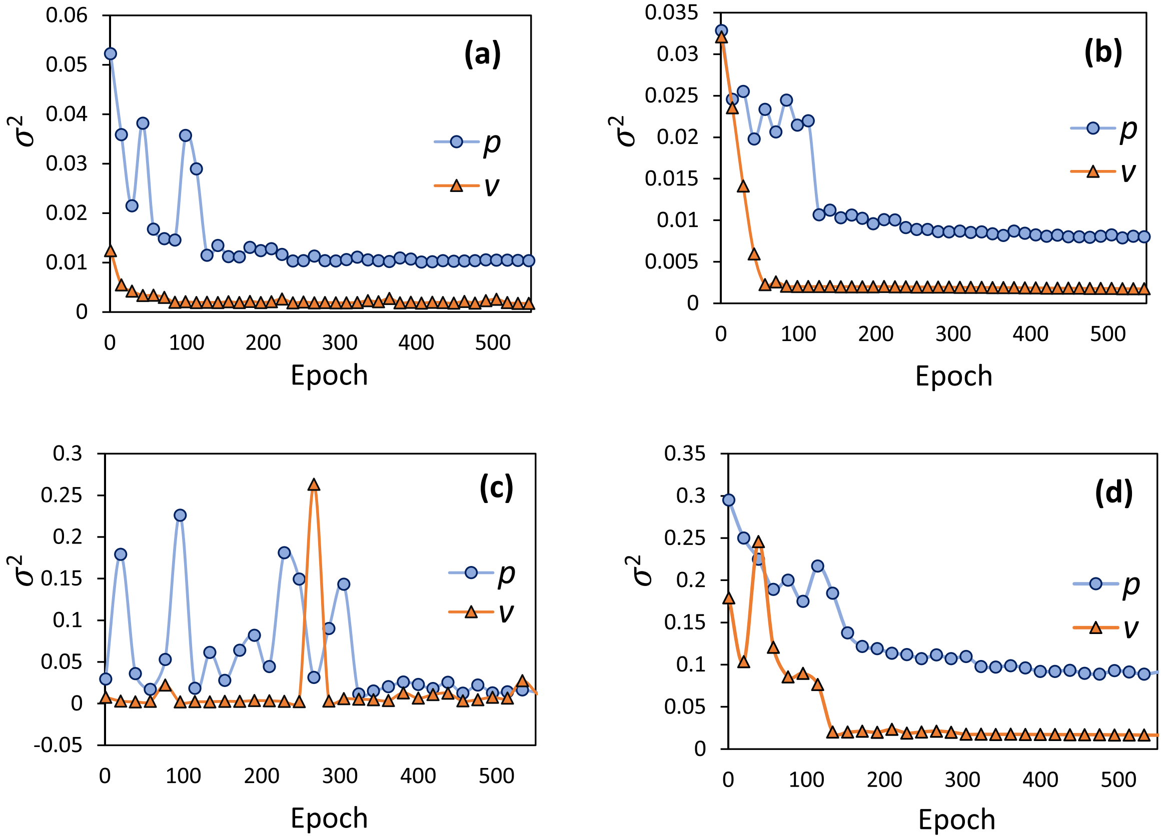

Fig. 1 Computational efficiency of the PIRED. Comparison of cost function σ^2 for the training and testing of the PIRED (**a**, **b**) and the DDML networks (**c**, **d**).

the training of the ML algorithm. As the input data, such as the images of pore space, are complex and large, one needs a network that decreases the computations time. Despite increasing applications of the ML methods, there has been only a limited number of attempts for addressing problems associated with porous materials[4–6,19–28].

RESULTS AND DISCUSSION

We divide our results into two parts. In one part, we present and discuss calculations for the polymeric membrane. In the next part, we demonstrate that the same physics-informed recurrent encoder–decoder (PIRED) network can provide accurate predictions for a completely different porous medium.

Effect of incorporating the governing equations in the learning of the algorithm

Using the proposed PIRED network, we reconstructed the velocity and pressure field in new images using only a small number of images without specifying any boundary conditions. Figure 1a, b present, respectively, the change in the cost function σ^2 for the training and testing datasets of the network. σ^2 decreases for both variables during both the training and test data, indicating convergence toward the true solutions for both the pressure and fluid velocity fields.

To demonstrate the accuracy and efficiency of the PIRED, we also computed σ^2 using a data-driven ML (DDML) algorithm, one in which the governing equations were not incorporated in the cost function. The results for σ^2 are presented in Fig. 1c, d for both the training and testing. As can be seen, compared with Fig. 1a, b, DDML has a much weaker performance.

To better illustrate the improvement produced by incorporating the governing equations in the cost function, the PIRED and DDML are compared in terms of the number of data points used for training the network based on the correlation coefficient R^2, as well as σ^2 in Fig. 2a, b. Clearly, there exists a significant difference between the two methods and, in particular, when the number of the samples is small. The discrepancy is, however, smaller when a larger number of datasets are used.

PIRED-predicted pressure and velocity fields

Figure 3 compares the predicted spatial distribution of the pressure $\hat{P}$ at four (dimensionless) times $t_1 = 15 < t_2 < t_3 < t_4 = 185$ with the actual P in one of the randomly selected 2D images not used in the training, with the results for all other slices being just as accurate (see below). Figure 4 compares the corresponding results for the magnitude $|\mathbf{v}|$ of the fluid velocity. As the spatial distributions of the differences $P - \hat{P}$ and $|\mathbf{v}| - |\hat{\mathbf{v}}|$ indicate, the predictions agree very closely with the actual data. Therefore, not only are the distributions of P and $\mathbf{v}$ reproduced accurately, the correlations between their values with increasing time are also honored.

Another quantitative comparison is based on selecting at random a vertical line (a plane in 3D) in one of the 300 testing images and comparing the PIRED-predicted P and $\mathbf{v}$ along that line with their actual values. One example is shown in Fig. 5a, b for pressure and velocity, which indicates a very good agreement between the predictions and the actual data. The same accuracy was obtained for all other slices.

Next, we compared the calculated ensemble-averaged maps of the PIRED-predicted $|\mathbf{v}|$ and $|\mathbf{P}|$ over the 300 testing images with the actual averages. The results are presented in Fig. 6, where Fig. 6a, c are the averages of actual (numerical) results and Fig. 6b,

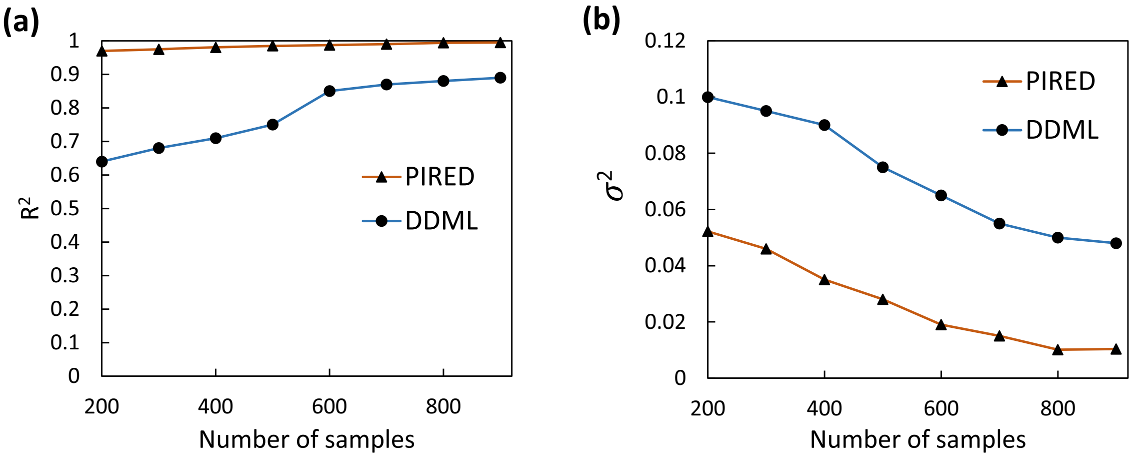

Fig. 2 Data size reduction of training of PIRED. Effect of data size on (**a**) the R^2 score and (**b**) σ^2 for PIRED and DDML networks.

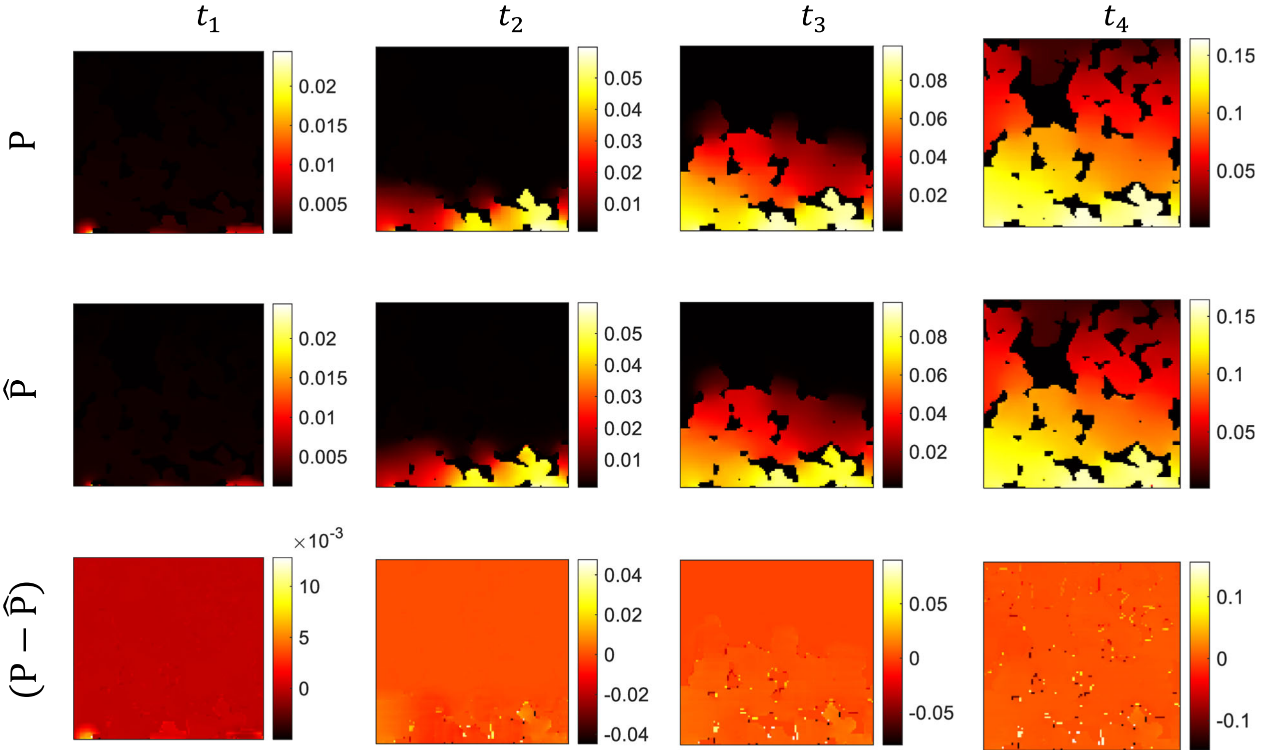

Fig. 3 Prediction of the pressure field by the trained PIRED. Comparison of the predicted pressure $\hat{P}$ with the numerically calculated P at four (dimensionless) times.

d are the averages of PIRED prediction results for velocity and pressure. The results in Fig. 6 indicate an excellent agreement.

PIRED predictions for another porous material

We define an effective permeability K by, $K = \mu Lq/(S\Delta P)$, where q, S, and ΔP are, respectively, the steady-state volume flow rate and the surface area perpendicular to the macroscopic pressure drop ΔP. K was computed for all the 300 testing slices and was predicted by the PIRED network as well. The comparison is shown in Fig. 7a.

However, a most stringent test of the PIRED network is if we predict the properties of a completely different porous medium, without using any data associated with it. Thus, we used the image of a Fontainebleau sandstone[18] with a porosity of 0.14. As the sandstone's morphology is completely different from the membrane's, we used a slightly larger number of 2D slices from the membrane (not the sandstone) to better train the PIRED network. Figure 7b compares the effective permeabilities of 100 2D slices of the sandstone with the predictions of the PIRED network. The images of the two types of porous media are also shown in Fig. 7c, d.

Figure 8 compares the predicted spatial distribution of the pressure $\hat{P}$ at four (dimensionless) times $t_1 < t_2 < t_3 < t_4$ with the actual P field in one of the randomly selected 2D images of the sandstone, with the results for all other slices being just as accurate. Figure 9 presents the corresponding results for the velocity field in the sandstone. In both cases, the agreement between the predictions and the actual fields is excellent. As the spatial distributions of the differences $P - \hat{P}$ and $|\mathbf{v}| - |\hat{\mathbf{v}}|$ in

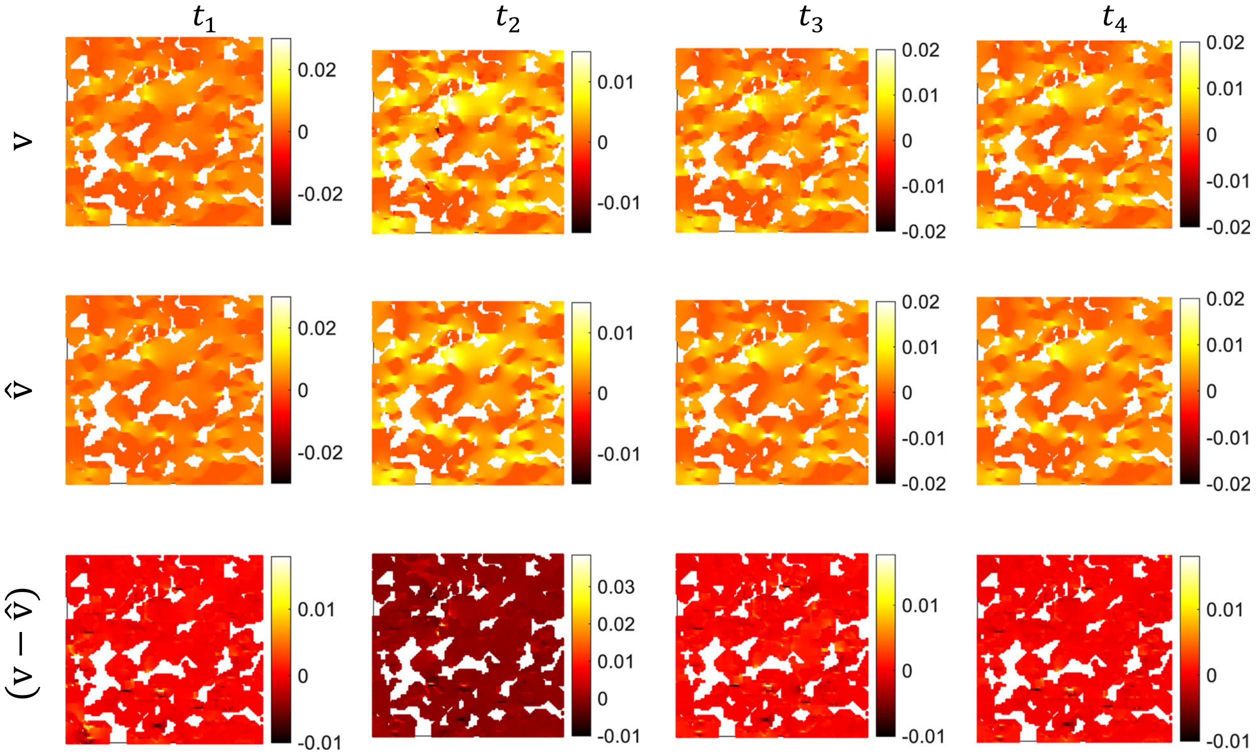

Fig. 4 Prediction of the fluid velocity field by the trained PIRED. Comparison of the predicted fluid velocity $|\hat{\mathbf{v}}|$ with the numerically calculated values at four (dimensionless) times.

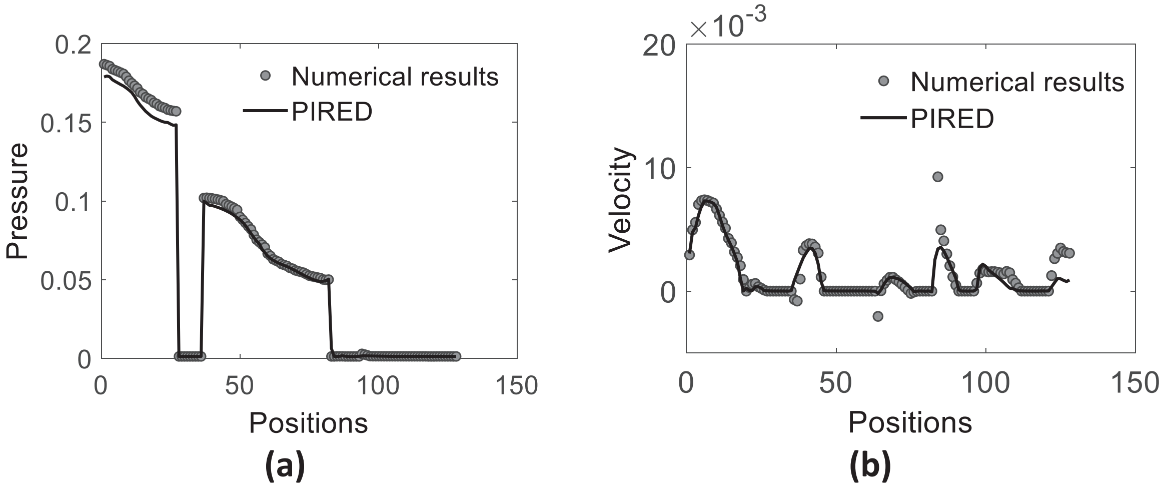

Fig. 5 Quantitative comparison of the PIRED predictions with the direct numerical results. Comparison of predicted (**a**) pressures and (**b**) fluid velocities with the numerical simulations in a randomly selected 2D image along a line perpendicular to the macroscopic direction of flow.

Figs. 8 and 9 indicate, both the pressure and velocity fields agree very closely with the actual distributions.

Summarizing, we presented a physics-informed encoder–decoder algorithm, the PIRED network, by incorporating the MC and the NS equations in its learning process, in order to predict fluid flow in a complex porous medium. The network provides highly accurate predictions for the fluid velocity and pressure fields at every point of the medium that was not used in the training, as well as its effective permeability. Not only does the PIRED network require a significantly smaller amount of data to make accurate predictions and, therefore, much less computations, it also provides accurate predictions for other types of porous media without using their data.

METHODS

We divide this section into four parts and explain the structure of PIRED and the details of the computations.

Computational details

If the input and output are both in the form of images, as is the case in this study, an autoencoder network produces more accurate predictions. However, when the input or output is represented by spatial images or time series, which is the case when one solves the MC and NS equations, recurrent neural networks (RNNs) connect the data series. The RNNs output is used as the input in the decoder section to generate output data. Therefore, we couple an RNN to an encoder–decoder network, resulting in

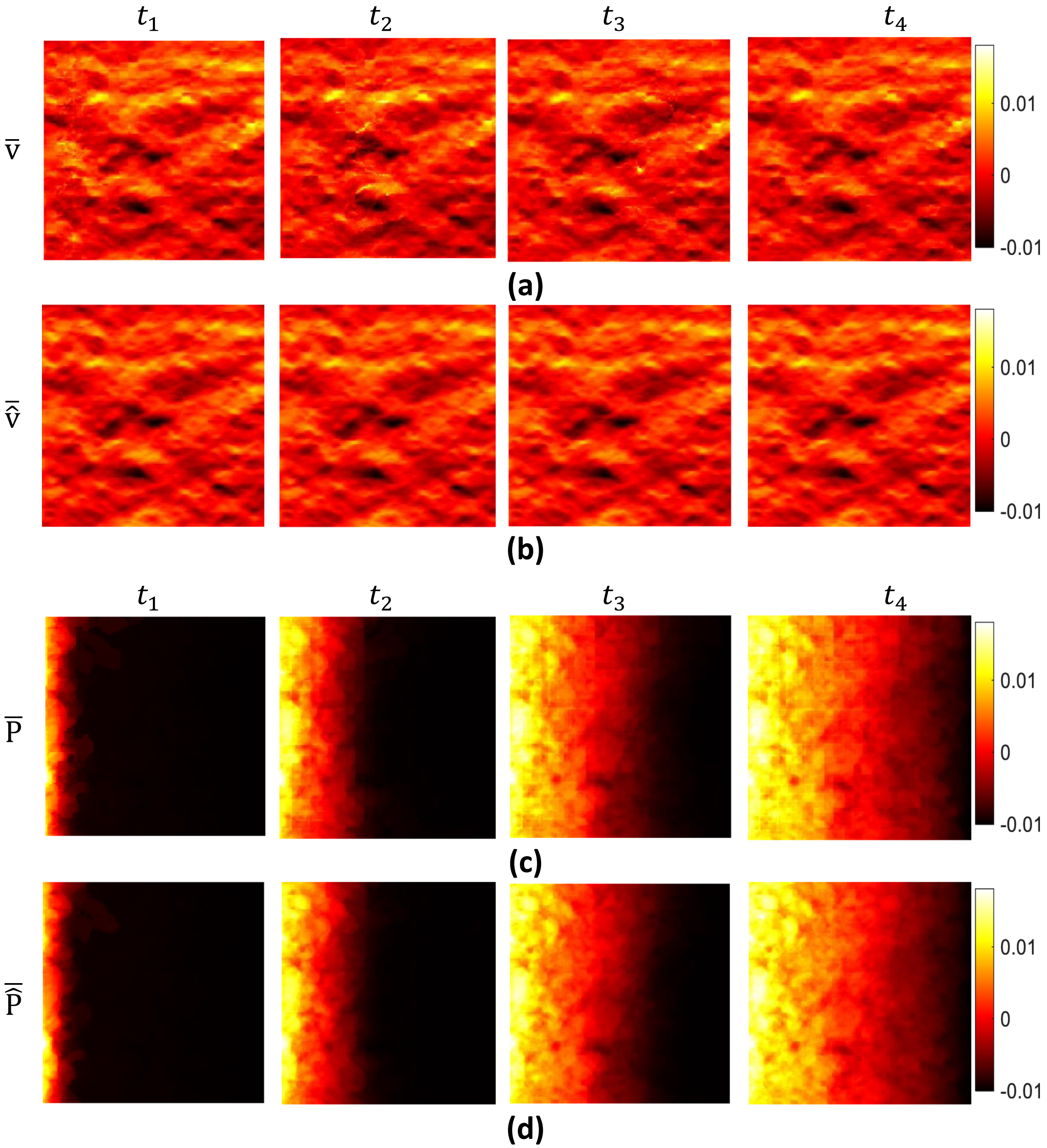

Fig. 6 Overall prediction accuracy of the PIRED. Comparison of the actual ensemble-averaged of (**a**) fluid velocities $\overline{\mathrm{v}}$ and (**c**) the fluid pressure $\overline{\mathrm{P}}$ with those predicted by the PIRED (**b**) and (**d**) for $\hat{\overline{v}}$ and $\hat{\overline{P}}$ in the polymeric network at four (dimensionless) times.

a PIRED network in which the cost function is defined partly based on the solutions of the MC and the NS equations.

We use the reverse Kullback–Leibler (KL) divergence (relative entropy)[29] for minimizing the cost function. We suppose that $p(x)$ is the true probability distribution of the input/output data, whereas $q(x)$ is an approximation to it. The reverse KL divergence from q to p is a measure of the difference between $p(x)$ and $q(x)$, and the aim is to ensure that $q(x)$ represents $p(x)$ accurately enough that it minimizes the reverse KL divergence $D_{\mathrm{KL}}(q\|p)$, given by

$$D_{\mathrm{KL}}[q(x)\|p(x)] = \sum_{x\in X} q(x)\log\left[\frac{q(x)}{p(x)}\right], \tag{1}$$

where X is the space in which $p(x)$ and $q(x)$ are defined. $D_{\mathrm{KL}} = 0$, if $q(x)$ matches $p(x)$ perfectly and, in general, it may be rewritten as

$$D_{\mathrm{KL}}[q\|p] = E_{x\sim q}[-\log p(x)] - H[q(x)], \tag{2}$$

where $H[q(x)] = E_{x\sim q}[-\log q(x)]$ is the entropy of $q(x)$, with E denoting the expected value operator and, thus, $E_{x\sim q}[-\log p(x)]$ is the cross-entropy between q and p. Optimization of D_{KL} with respect to q is defined by

$$\arg\min D_{\mathrm{KL}}[q\|p] = \arg\min E_{x\sim q}[-\log p(x)] - H[q(x)] = \arg\min E_{x\sim q}[\log p(x)] + H[q(x)]. \tag{3}$$

Thus, according to Eq. (3), one samples points from $q(x)$ and does so, such that they have the maximum probability of belonging to $p(x)$. The entropy term of Eq. (3) "encourages" $q(x)$ to be as broad as possible. Thus, the autoencoder tries to identify a distribution $q(x)$ that best approximates $p(x)$.

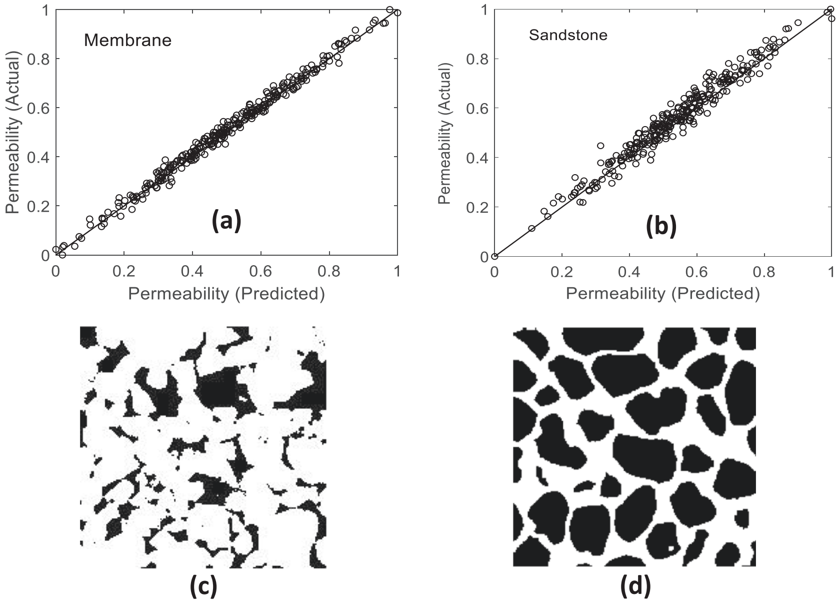

Fig. 7 PIRED efficiency in predicting permeability of another porous medium, a sandstone. Comparison of the actual and predicted permeabilities K (K is normalized according to $(K-K_{\min})/(K_{\max}-K_{\min})$) for (**a**) 300 2D images of the membrane, whose morphology is shown in (**c**), and (**b**) for 100 images of a sandstone, the morphology of which is shown in (**d**). In (**c**) and (**d**), black and white represent the solid matrix and the pores, respectively.

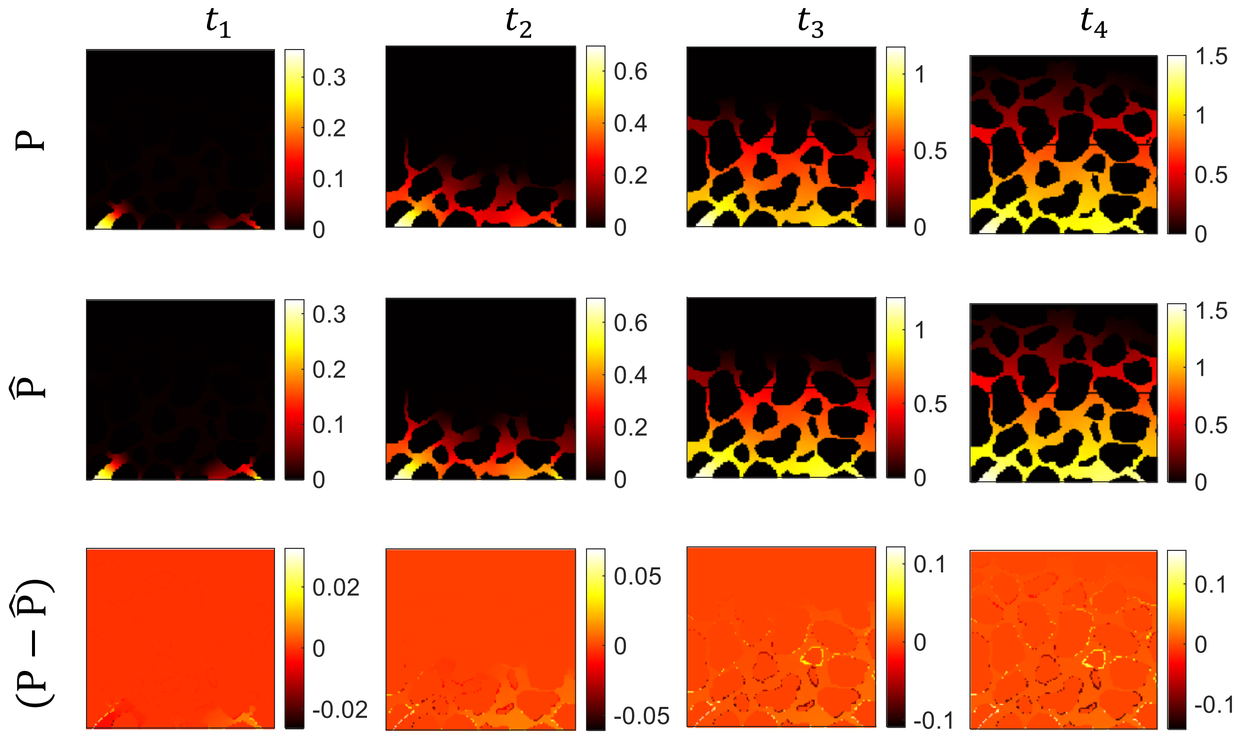

Fig. 8 PIRED predictions for the fluid pressure in the sandstone. Comparison of the predicted pressure $\hat{P}$ with the numerically calculated P at four (dimensionless) times in a randomly selected 2D cut of the sandstone.

Network architecture

As in the present problem the input and output images are distinct, the PIRED network, a supervised one, consists of encoder and decoder, known as the U-Net and residual U-Net (see Fig. 10). The encoder has four blocks with the block containing the standard convolutional (CL) and activation layers (AL), and the pooling and batch normalization layers (PL and BNL). The PL compresses the input to its most important characteristics, eliminating the unnecessary features, and stores them in the latent layer, which itself consists of the AL, CL, and BNL. The BNL not only allows the use of higher learning rates by reducing internal covariate shift but also

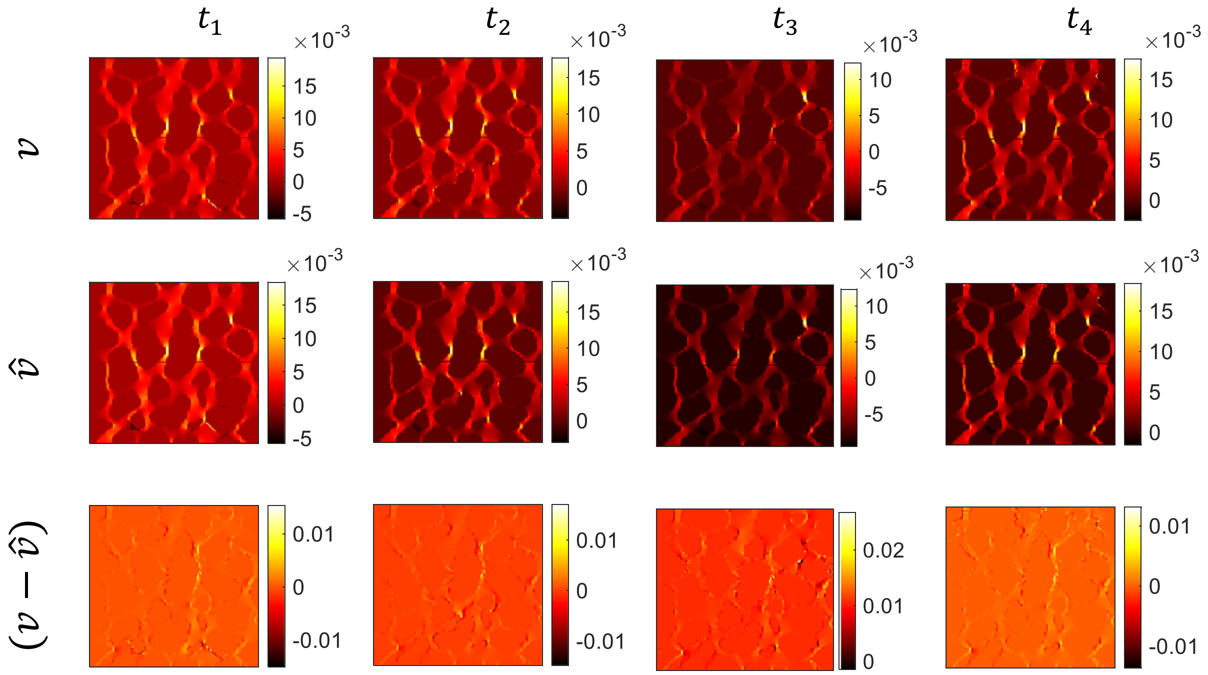

Fig. 9 PIRED predictions for the fluid velocity in the sandstone. Comparison of the predicted fluid velocities *v* with the numerically calculated values v̂ at four times.

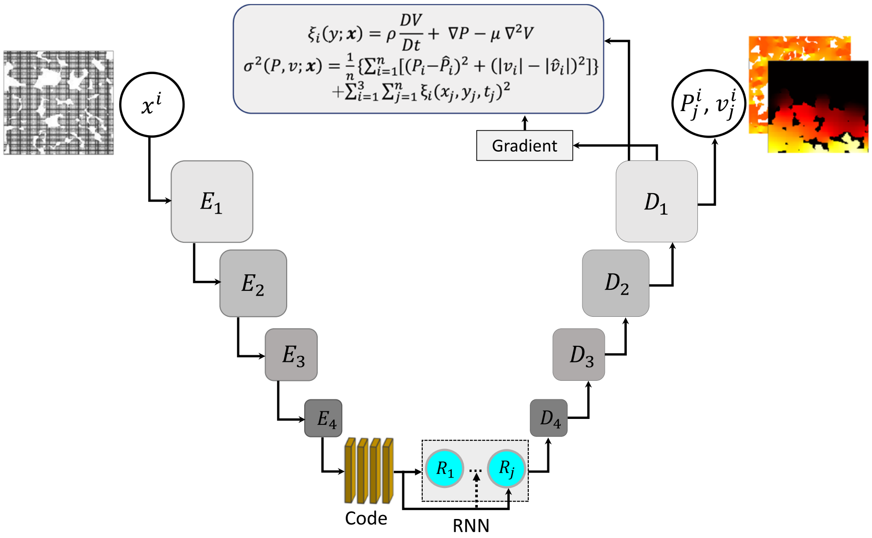
$\xi_i(y;\boldsymbol{x}) = \rho\frac{DV}{Dt} + \nabla P - \mu\nabla^2 V$
$\sigma^2(P,v;\boldsymbol{x}) = \frac{1}{n}\{\sum_{i=1}^{n}[(P_i - \hat{P}_i)^2 + (|v_i| - |\hat{v}_i|)^2]\} + \sum_{i=1}^{3}\sum_{j=1}^{n}\xi_i(x_j, y_j, t_j)^2$
x^i
P_j^i, v_j^i
Gradient
E_1
E_2
E_3
E_4
D_1
D_2
D_3
D_4
R_1
R_j
Code
RNN

Fig. 10 Schematic of the proposed PIRED network. Network architecture with E_i and D_i, indicating the encoder and decoder blocks; σ^2 is the cost function, x^i is the input, and the pressure P^j and fluid velocity $|v|^j$ are the output.

acts as a regularizer for reducing overfitting[30]. The mean $\langle x\rangle$ and variance Var[*x*] of batches of data *x* are computed in the BNL and a new normalized variable *y* is defined by

$$y = \gamma\frac{x - \langle x\rangle}{\sqrt{\mathrm{Var}[x] + \epsilon}} + \beta. \tag{4}$$

Here, γ and β are learnable parameter vectors that have the same size as the input data, and ϵ is set at 10^{-5}. During training, the layer keeps running estimates of its computed mean and variance, and uses them for normalization during evaluation. The variance is calculated by the biased estimator.

The decoder also consists of four blocks, with each block containing the CL, AL, and BNL, as well as a transposed CL (TCL), which is similar to a deconvolutional layer in that if, e.g., the first encoder has a size 128 × 64 × 64 (i.e., 128 features with a size 64 × 64), then, one has a similar size in the decoder. The TCL uses the features extracted by the PL to reconstruct the output, the pressure *P* and fluid velocity **v** fields at each specified time epoch. The latent layer of the RNN that we use improves its performance and speeds up significantly the overall network's computations, because it is in the form of residual blocks, i.e., layers that, instead of having only one connection, are connected to further previous layers.

The input and output data

We used a high-resolution 3D image of a polymeric membrane of size 500 × 500 × 1000 voxels. Its porosity, thickness, permeability, and mean pore size are, respectively, 0.77, 60 μm, 10^{-12} m^2, and 8 μm, respectively. An image of a 2D slice of the membrane is shown in Fig. 7c. We selected at random 700 2D slices of the image with size 175 × 175 pixels for the fluid flow calculations and training the PIRED, and another 300 slices for testing the accuracy. The 700 images were inserted in the PIRED network's first layer, whereas the last layer contained the output, the distributions of P and $\mathbf{v}$. The fluid density and viscosity were set at 0.997 g/cm^3 and 1.89×10^{-3} g/(cm · s), with the fluid injection velocity being 1 cm/s.

We computed the P and $\mathbf{v}$ fields at four times and represented them by images. It is noteworthy that the amount of the data needed for computing P and $\mathbf{v}$ is significantly smaller than what would be needed by the standard ML methods.

Training PIRED

If L and v_0 represent the characteristic length scale and fluid velocity in the medium, we introduce dimensionless variables, $x^* = x/L$, $y^* = y/L$, $\mathbf{v}^* = \mathbf{v}/v_0$, $t^* = tv_0/L$, and $P^* = PL/(\mu v_0)$. Deleting superscript $_*$ for convenience, the MC equation, $\nabla \cdot \mathbf{v} = \partial v_x/\partial x + \partial v_y/\partial y = 0$ remains unchanged. The NS equation becomes

$$\frac{D\mathbf{v}}{Dt} = \frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} = \mathrm{Re}^{-1}\left(-\nabla P + \nabla^2 \mathbf{v}\right), \tag{5}$$

where $\mathrm{Re} = \rho v_0 L/\mu$ is the Reynolds number. We define three functions, $\xi_1 = \nabla \cdot \mathbf{v}$, $\xi_2 = Dv_x/Dt - \mathrm{Re}^{-1}\left(-\partial P/\partial x + \nabla^2 v_x\right)$, and $\xi_3 = Dv_y/Dt - \mathrm{Re}^{-1}\left(-\partial P/\partial y + \nabla^2 v_y\right)$, and incorporate them in the cost function σ^2, minimized by the PIRED network, instead of naively minimizing the squared error between the data and predicted values of $\mathbf{v}$ and P. For exact convergence to the actual (numerically calculated) values (by solving the MC and NS equations), we must have $\xi_i = 0$ with $i = 1 - 3$. Thus, the PIRED network learns that the mapping between the input and output must comply with $\xi_i = 0$, which not only enriches its training but also accelerates convergence to the actual values. σ^2 is defined by

$$\sigma^2 = \frac{1}{n}\left\{\sum_{i=1}^{n}\left[\left(P_i - \hat{P}_i\right)^2 + \left(|v_i| - |\hat{v}_i|\right)^2\right]\right\} + \sum_{i=1}^{3}\sum_{j=1}^{n}\xi_i\left(x_j, y_j, t_j\right)^2, \tag{6}$$

where n is the number of data points used in the training, and $_{P_i}$ and $|v_i|$ are the actual pressure and magnitude of the velocity at point (x_i, y_i) at time t_i, with superscript ^ denoting the predictions by the PIRED network.

The derivatives in ξ_i were estimated using the Sobel operator[31], an inexpensive and effective way for computing the gradients, used commonly in image processing. It may be thought of as a smoothed finite-difference operator consisting of two 3×3 convolution kernels for the horizontal (H) and vertical (V) directions, which convolve with the image $\mathbf{I}$ in order to estimate the H and V derivatives. The kernels are given by, $G_x = \mathbf{M} * \mathbf{I}$ and $G_y = \mathbf{M}^T * \mathbf{I}$, where T and $*$ represent the transpose and convolution operations, and

$$\mathbf{M} = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}. \tag{7}$$

To further quantify the accuracy of the results, we also computed R^2, which is a measure indicating the closeness of the predictions and the actual data; for a very accurate model, one should have, $R^2 \approx 1$.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}\|\psi_i - \hat{\psi}_i\|_2^2}{\sum_{i=1}^{n}\|\psi_i - (1/n)\sum_{j=1}^{n}\psi_j\|_2^2}, \tag{8}$$

where $\hat{\psi}_i$ is the network's prediction, ψ_i is the actual value, and n is the number of the samples.

We solved the MC and the NS equations using the open-source OpenFOAM. The fluid was injected at one side and a fixed pressure was applied to the opposite side. The other two boundaries were assumed to be impermeable.

Solving the MC and NS equations in each 2D image took about 6 central processing unit (CPU) minutes. The computations for training the PIRED network on an Nvidia Tesla V100 graphics processing unit (GPU) took about 2 GPU hours. Then, the tests took less than a second.

DATA AVAILABILITY

The data supporting findings within this paper are available from the corresponding author upon reasonable request.

CODE AVAILABILITY

The code supporting findings within this paper are available from (servehka@usc.edu) upon reasonable request.

Received: 18 April 2021; Accepted: 20 July 2021;
Published online: 12 August 2021

REFERENCES

1. Blunt, M. J. *Multiphase Flow in Permeable Media: A Pore-Scale Perspective*. (Cambridge Univ. Press, 2017).
2. Sahimi, M. *Flow and Transport in Porous Media and Fractured Rock*. 2nd edn (Wiley-VCH, 2011).
3. Jablonka, K. M., Ongari, D., Moosavi, S. M. & Smit, B. Big-data science in porous materials: materials genomics and machine learning. *Chem. Rev.* **120**, 8066 (2020).
4. Kamrava, S., Tahmasebi, P. & Sahimi, M. Enhancing images of shale formations by a hybrid stochastic and deep learning algorithm. *Neural Netw.* **118**, 310 (2019).
5. Wu, H., Fang, W. Z., Kang, Q., Tao, W. Q. & Qiao, R. Predicting effective diffusivity of porous media from images by deep learning. *Sci. Rep.* **9**, 1 (2019).
6. Kamrava, S., Tahmasebi, P. & Sahimi, M. Linking morphology of porous media to their macroscopic permeability by deep learning. *Transp. Porous Media* **131**, 427 (2020).
7. Bishop, C. M. *Neural Networks for Pattern Recognition*. (Clarendon, 1995).
8. Hamzehpour, H. & Sahimi, M. Development of optimal models of porous media by combining static and dynamic data: the porosity distribution. *Phys. Rev. E* **74**, 026308 (2006).
9. Hamzehpour, H., Rasaei, M. R. & Sahimi, M. Development of optimal models of porous media by combining tatic and dynamic data: the permeability and porosity distributions. *Phys. Rev. E* **75**, 056311 (2007).
10. Raissi, M., Perdikaris, P. & Karniadakis, G. E. Machine learning of linear differential equations using Gaussian processes. *J. Comput. Phys.* **348**, 683 (2017).
11. Bar-Sinai, Y., Hoyer, S., Hickey, J. & Brenner, M. P. Learning data-driven discretizations for partial differential equations. *Proc. Natl. Acad. Sci. USA* **116**, 15344 (2019).
12. Meade, A. J. & Fernandez, A. A. The numerical solution of linear ordinary differential equations by feedforward neural networks. *Math. Comput. Model* **19**, 1 (1994).
13. He, S., Reif, K. & Unbehauen, R. Multilayer neural networks for solving a class of partial differential equations. *Neural Netw.* **13**, 385 (2000).
14. Aarts, L. P. & Van Der Veer, P. Neural network method for solving partial differential equations. *Neural Process. Lett.* **14**, 261 (2001).
15. Han, J., Jentzen, A. & Weinan, E. Solving high-dimensional partial differential equations using deep learning. *Proc. Natl. Acad. Sci. USA* **115**, 8505 (2018).
16. Raissi, M., Perdikaris, P. & Karniadakis, G. E. Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **378**, 686 (2019).
17. Raissi, M., Yazdani, A. & Karniadakis, G. E. Hidden fluid mechanics: learning velocity and pressure fields from flow visualizations. *Science* **367**, 1026 (2020).
18. Andreä, H. et al. Digital rock physics benchmarks-Part I: imaging and segmentation. *Comput. Geosci.* **50**, 25 (2013).
19. Tahmasebi, P., Kamrava, S., Bai, T. & Sahimi, M. Machine learning in geo- and environmental sciences: from small to large scale. *Adv. Water Resour.* **142**, 103619 (2020).
20. Van der Linden, J. H., Narsilio, G. A. & Tordesillas, A. Machine learning framework for analysis of transport through complex networks in porous, granular media: a focus on permeability. *Phys. Rev. E.* **94**, 022904 (2016).
21. Erofeev, A., Orlov, D., Ryzhov, A. & Koroteev, D. Prediction of porosity and permeability alteration based on machine learning algorithms. *Transp. Porous Media* **128**, 677 (2019).
22. Kamrava, S., Sahimi, M. & Tahmasebi, P. Quantifying accuracy of stochastic methods of reconstructing complex materials by deep learning. *Phys. Rev. E* **101**, 043301 (2020).
23. Mosser, L., Dubrule, O. & Blunt, M. J. Reconstruction of three-dimensional porous media using generative adversarial neural networks. *Phys. Rev. E* **96**, 043309 (2017).
24. Zhang, F., Teng, Q., Chen, H., He, X. & Dong, X. Slice-to-voxel stochastic reconstructions on porous media with hybrid deep generative model. *Comput. Mater. Sci.* **186**, 110018 (2021).
25. Wei, H., Zhao, S., Rong, Q. & Bao, H. Predicting the effective thermal conductivities of composite materials and porous media by machine learning methods. *Int. J. Heat Mass Transfer* **127**, 908 (2018).

26. Alqahtani, N., Alzubaidi, F., Armstrong, R. T., Swietojanski, P. & Mostaghimi, P. Machine learning for predicting properties of porous media from 2d X-ray images. *J. Pet. Sci. Eng.* **184**, 106514 (2019).
27. Feng, J., Teng, Q., He, X. & Wu, X. Accelerating multi-point statistics reconstruction method for porous media via deep learning. *Acta Mater.* **159**, 296 (2018).
28. Srinivasan, S., Karra, S., Hyman, J., Viswanathan, H. & Srinivasan, G. Model reduction for fractured porous media: a machine learning approach for identifying main flow pathways. *Comput. Geosci.* **23**, 617 (2019).
29. Kullback, S. & Leibler, R. A. On information and sufficiency. *Annal. Math. Statist* **22**, 79 (1951).
30. Ioffe, S. & Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. Preprint at https://arXiv:1502.03167V3 (2015).
31. Zhu, Y., Zabaras, N., Koutsourelakis, P. S. & Perdikaris, P. Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data. *J. Comput. Phys.* **394**, 56 (2019).

ACKNOWLEDGEMENTS

We are grateful to the National Science Foundation for partial support of this work through Grant CBET 2000968. Most of the computations were carried out at the High-Performance Computing Center of the University of Southern California.

AUTHOR CONTRIBUTIONS

S.K.: Methodology of PIRED network, computations, software, and contributing to the writing of the first draft of the paper. P.T.: Data curation and review. M.S.: Conceptualization, supervision of the project, writing the final draft of the paper, and funding acquisition.

COMPETING INTERESTS

The authors declare no competing interests.

ADDITIONAL INFORMATION

Correspondence and requests for materials should be addressed to M.S.

Reprints and permission information is available at http://www.nature.com/reprints

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.