



Research Article

Analyzing the efficiency and robustness of deep convolutional neural networks for modeling natural convection in heterogeneous porous media



Mohammad Mahdi Rajabi^{a,*}, Mohammad Reza Hajizadeh Javaran^{a,b}, Amadou-oury Bah^b, Gabriel Frey^c, Florence Le Ber^d, François Lehmann^b, Marwan Fahs^b

^a Civil and Environmental Engineering Faculty, Tarbiat Modares University, PO Box 14115-397, Tehran, Iran

^b Université de Strasbourg, CNRS, ENGEES, ITES UMR 7063, F-67000 Strasbourg, France

^c Université de Strasbourg, CNRS, ICube UMR 7357, F-67000 Strasbourg, France

^d Université de Strasbourg, CNRS, ENGEES, ICube UMR 7357, F-67000 Strasbourg, France

ARTICLE INFO

Article history:

Received 2 July 2021

Revised 19 September 2021

Accepted 17 October 2021

Keywords:

Natural convection

Porous media

Deep learning

Convolutional neural networks

Image-to-image regression

ABSTRACT

Natural convection in porous media (NCPM) is governed by highly nonlinear dynamics due to the dependence of fluid density and viscosity to temperature. Given the high computational cost associated with numerical modeling of NCPM, data-driven metamodels are commonly used to reduce the computational time in applications that require repetitive model runs. However, to various degrees, all metamodels previously used in this context struggle at scaling to high dimensional input and output problems. This work aims at investigating the performance of encoder-decoder convolutional neural networks (ED-CNNs), as a specialized deep learning architecture, in assisting the procedure of numerical modeling of NCPM. Our interest is directed to image-to-image regression tasks in which both the inputs and outputs of the model are high-dimensional, often high resolution, spatial distributions of the features. Large datasets of images (for e.g. heat map) can be generated by numerical modeling of NCPM, and also through high-resolution imaging and non-destructive scanning techniques. Hence, we apply ED-CNNs to develop a methodology for image-to-image regression. The goals are twofold: (1) to assess the robustness of ED-CNNs in metamodeling and uncertainty propagation analysis, and (2) to evaluate the performance of ED-CNNs as optimizer in input parameter estimation. To do so, we apply the ED-CNNs to the common benchmark of natural convection in a porous cavity. Numerical experiments highlight the robustness and efficiency of the ED-CNNs in handling heterogeneous domains.

© 2021 Elsevier Ltd. All rights reserved.

1. Introduction

Natural convection caused by density variations in saturated porous media, is an important phenomenon in many natural and industrial systems, ranging from the microscale in fuel cells to the macroscale in geoscience and environmental applications [1]. Physics-based numerical models are key to understanding, predicting and designing of processes encompassing natural convection in porous media (NCPM). As a result, these models have been studied and applied extensively in the literature [2]. However, NCPM models are computationally expensive, especially for large scale studies, as in oil and geothermal reservoir simulations, and it's not unusual to encounter models with a runtime of several hours or days. This

is notably problematic in applications that require repetitive model runs in order to obtain convergent statistics. Such applications include uncertainty propagation, sensitivity analysis, data assimilation, scenario analysis and stochastic optimization [3].

In the literature on NCPM, data-driven metamodels are commonly used to deal with such computational challenges. These metamodels are trained with a limited number of numerical model input-output pairs, and then replace the actual numerical model in repetitive tasks. Several techniques of metamodeling have been applied to NCPM, such as polynomial chaos expansions (e.g. [4,5]), Gaussian processes emulators (e.g. [6]), support vector machines [7], and neural networks [8]. Previous studies have demonstrated the accuracy and computational efficiency of applying such metamodels in relatively low-dimensional problems. However, to various degrees, all these metamodels struggle at scaling to high dimensional input and output problems [9,10]. This is due to the

* Corresponding author.

E-mail address: mmrajabi@modares.ac.ir (M.M. Rajabi).

fact that the computational cost of metamodel construction increases exponentially with increase in problem dimensionality, and this problem is exacerbated in highly non-linear problems such as NCPM.

Parameter estimation as the inverse problem, is also an important task in NCPM. It proceeds by optimizing of an objective function expressing the discrepancy between observations and numerical results. Parameter estimation is a computationally heavy task as it requires repetitive simulations [11]. Several techniques have been used in the literature such as the Levenberg–Marquardt method and genetic algorithms [12]. Despite extensive works on parameter estimation, this topic is still challenging as it requires not only an efficient forward model, but also an efficient optimizer that reduces the number of model evaluations.

Deep neural networks (DNNs) provide a possibility to handle the aforementioned challenges. Although there is no clear dividing line between DNNs and classic neural networks, the main distinction is the hierarchical feature extraction architecture of DNNs, which generally leads to a higher number of hidden layers. The use of DNNs gained momentum around 2012 as a result of breakthrough achievements in image classification [13]. With a lag of few years, DNNs became a novel force in several engineering applications, as they showed a remarkable leap in predictive capability. Well-designed DNNs have been shown to provide improved accuracy and better generalization abilities compared to other metamodels in different engineering simulation problems (see e.g. [14]). DNNs perform particularly well in situations where either or both the input and output are high-dimensional [15]. This is important in modeling NCPM, as porous media is often inherently heterogeneous and the output state variables are space and time dependent [16]. This feature of DNNs has been attributed to hierarchical projections of the input space into low-dimensional latent spaces [17], and allows for the avoidance of the usual dimensionality reduction step [18]. Recently there is an increasing interest in applying DNNs as optimizers [19].

NCPM processes are governed by complex physical laws. The outputs (temperature distribution, velocity field and heat transfer flux) have usually complex spatial distributions. Thus, this work aims at investigating the performance of DNNs in modeling of NCPM. In the next subsections, we will first provide a brief literature review on the applications of DNNs in modeling flow and/or transport processes in porous media before refining and specifying the objectives of this study in more detail.

1.1. Literature review

DNNs have been used to address a number of important problems in porous media, including flow [18,15], solute transport [20,21], multiphase flow [22], density-driven solute transport [23], and unsaturated flow [24]. A review of selected literature on the use of DNNs in modeling flow and transport in porous media is provided in Table 1. As demonstrated in this table, most previous application of DNN in modeling flow and transport in porous media deal with steady-state conditions (e.g. in [21,18]). However, time-dependent outputs in a transient system can also be modeled by considering time as an additional input to the DNN (see [22]), or treating the output at the previous time step as input to the DNN for predicting the output at the current time step (see [21]).

In most previous studies DNN modeling does not replace numerical modeling but is complementary to it [23], as DNNs are commonly trained using data obtained from numerical modeling. However, a number of studies have trained DNNs using experimental data, for example with the purpose of estimating permeability from images of porous media (see [25,26]).

After the training process, DNNs have been shown to provide a level of accuracy that is comparable to numerical models of flow

and transport in porous media, albeit with a highly reduced computational time. This allows DNNs to efficiently replace numerical models in tasks that require repetitive simulations such as uncertainty propagation analysis (as in [18]). The predictive and generalization capabilities of DNNs have been demonstrated using both hypothetical problems (in e.g. [21,18], b) and laboratory experiments (e.g. Hele-Shaw cell experiment in [23]). In the context of geoscience, a few studies (e.g. [18,15]) have tried to provide predictive uncertainty of DNNs by treating the network parameters as random variables, and performing Bayesian inference on them conditioned on limited observations.

DNNs can be developed using a purely data-driven approach (as in [18]), or they can be physics-informed (or physics-constrained) (e.g. [25]). The latter is particularly valuable when training data is insufficient or even none-existent (as in unsupervised learning, see [15]). Physics-informed DNNs can be constructed by minimizing the differential equation residuals at select points in the simulation domain (see e.g. [27,20,24]). When training data is minimal, physics-informed DNNs can be used to increase the generalization and accuracy of DNNs [27].

Current interest is mostly directed to *image-to-image* regression tasks (as in [23,21,18]), in which both the inputs and outputs of the DNN are high-dimensional, often high resolution, spatial distributions of the features. Encoder-decoder Convolutional Neural Networks (ED-CNNs) are a specialized DNN architecture well-suited for image-to-image regression. As demonstrated in Table 1, ED-CNNs are the most commonly used type of DNN in previous studies, but fully connected deep neural networks (FCDNN) (in e.g. [20,24]) and generative adversarial neural networks (GANs) (in e.g. [27,28]) have also been employed in some studies. The back-propagation algorithm is currently the dominant approach for training DNNs [19].

DNNs have allowed the concept of metamodeling to be extended to problems of input dimensions not considered before. For example, the input dimensions of 4225 or 2500 employed using DNN in Zhu and Zabaras [18] and Mo et al. [22], respectively, are out of reach using other conventional metamodeling approaches such as PCEs. DNNs have also facilitated metamodeling for non-Gaussian conductivity fields (e.g. [29]) and channelized permeability fields (e.g. [18,15]).

In the context of parameter estimation, DNNs have previously been used for: (a) the estimation of space-dependent flow and transport properties of porous media, such as permeability [25], diffusion coefficients [24,26], and thermal conductivity [14], (b) joint estimation of time-varying contaminant source characteristics and permeability (e.g. [21]), (c) joint estimation of states (e.g. head or capillary pressure) and parameters (e.g. hydraulic conductivity) [24], and (d) prediction of one state variable from another such as the estimation of velocity field from the concentration field in Kreyenbergh et al. [23]. There are basically two approaches for using DNNs in parameter estimation. The first is to use DNNs as a surrogate of the numerical models in common inverse modeling algorithms (as in [21]). The second approach is to train DNNs to directly map the state variables to parameters (as in [20,24]). In the latter case, DNN serve as an *optimizer* for parameter estimation, encompassing both the modeling and optimization tasks found in common inverse modeling algorithms, in a single model.

1.2. Study objectives

Recent progress in DNNs have benefitted various problems involving flow and transport processes in porous media, and it is becoming increasingly clear that DNNs present an opportunity for improving forward and inverse modeling. NCPM is a particular case where coupled flow and heat transfer processes lead to a highly nonlinear problem and create a complex velocity field. Despite

Table 1

Review of selected literature on the use of DNNs in modeling flow and transport in porous media.

Refs.	Nature of the problem	Case	Type of DNN	Input	Output	Application
Zhu and Zabaras [18]	Steady-state flow (Ht)	2D square domain	CNN	K field	V, P fields	Metamodel in UPA
Kreyenberg et al. [23]	Density-driven solute transport	Hele-Shaw cell experiment	CNN	C field	V field	Estimating one state variable from another
Mo et al. [21]	Steady-state flow and transient solute transport (Ht)	2D rectangular domain	CNN	L_t, S_t, K field	H, C field	Metamodel in CSC (S/O)
Mo et al. [22]	Transient multiphase flow (Ht)	Geological carbon storage	CNN	K field	P, S_{CO_2} fields	Metamodel in UPA
Zhu et al. [15]	Steady-state flow (Ht)	2D square domain	CNN	K field	P field	Both Physics-informed and data-driven metamodel in UPA,
He et al. [20]	Steady-state flow and transport (Ht)	2D synthetic tracer experiment	FCDNN	x, y	k, H, C fields	Joint state-parameter estimation
Tartakovsky et al., [24]	Unsaturated flow (Richards equation-type) (Hm)	2D square domain	FCDNN	x, y	k, H, P fields	Joint state-parameter estimation
Jiang et al. [33]	Transient multiphase flow (Ht)	2D channelized geological system	CNN	K field	P, S field	Surrogate model for Uncertainty analysis
Xiao et al. [42]	Two phase flow dynamics (Ht)	2D synthetic reservoir non Gaussian model	FCDNN	K field	S field	Surrogate model for Uncertainty analysis
Tang et al. [43]	Two phase flow dynamics	3D synthetic reservoir	CNN	K field	P, S field	Surrogate model for Data assimilation

Ht: heterogeneous media, Hm: homogeneous media, FCDNN: Fully connected deep neural network, CNN: convolution neural network, K : Permeability, k : hydraulic conductivity V : velocity, P : pressure, H : hydraulic head x, y : spatial coordinates, C : concentration, S_{CO_2} : CO_2 saturation, L_t, S_t : time-varying contaminant source location and release strength, S : saturation, CSC: contaminant source characterization, S/O: simulation-optimization, UPA: uncertainty propagation analysis.

large interest in DNNs in porous media applications, to the best of our knowledge, these techniques have been never applied to NCPM. In this work, we will be leveraging some of the most recent advances in DNNs, to develop a robust methodology for application to modeling NCPM. Our objective is to evaluate the performance of DNNs in assisting numerical models of NCPM. Large datasets of images (for e.g. heat map) can be generated by numerical modeling of NCPM, and also through high-resolution imaging and non-destructive scanning techniques. Hence, we apply ED-CNNs (as a specialized DNN architecture) to develop a methodology for image-to-image regression. In particularly, the goals are twofold: (1) to assess the robustness of ED-CNNs in metamodeling and uncertainty propagation analysis (UPA) and (2) to evaluate the performance of ED-CNNs as optimizer in a calibration procedure.

The remaining of the paper is organized as follows: In [Section 2](#), we describe our two NCPM test cases, and explain the governing equations of the numerical model used in their simulations. We then provide a brief description of the theoretical background of ED-CNNs and explain how they are employed for forward and inverse modeling of NCPM. In [Section 3](#), we present the results and evaluate the performance of ED-CNNs in different circumstances. We finally conclude and discuss the various unexplored research directions in [Section 4](#).

2. Theoretical framework and methodology

2.1. Problem statement and assumptions

The performance of ED-CNN is studied using the common benchmark problem of natural convection in a porous cavity [Fajraoui et al. \[4\]](#). This test case has been commonly used in the literature for several purposes such as in understanding physical processes and assessing newly developed numerical schemes. Two test cases are considered, as depicted in [Fig. 1](#). In both test cases, we consider a square domain of length H [L] filled with a saturated porous medium. It is assumed that the third dimension of the enclosures is sufficiently large, so that the fluid flow and heat transports are two-dimensional. We assume Dirichlet boundary conditions for temperature at the vertical walls, and thus have constant values of temperature T_L and T_R for the left and right

walls, respectively, where $T_L > T_R$. The top and bottom boundaries of the problem domain are thermally adiabatic and impermeable and are represented by Neumann boundary conditions ($\frac{\partial T}{\partial y} = 0$). Impervious flow boundaries are imposed on all walls. Gravity acts in the negative y -direction which is orthogonal to the direction of the temperature gradient. The porous media is assumed to possess isotropic thermo-physical properties, and it is in local thermal and compositional equilibrium with the saturating fluid. Viscous dissipation and porous medium inertia are not considered, and the Soret and Dufour effects are neglected. The fluid is assumed to be Newtonian and incompressible, and its flow in the cavity is steady-state laminar and complies with Darcy's law. The distinct features of the two test case problems are as follows:

a) **Test case (1): Heterogeneous porous enclosure.** As demonstrated in [Fig. 1a](#), the square porous cavity described above is assumed to be heterogeneous with nine isotropic blocks of equal size. Each block has particular thermal and hydraulic properties leading to a specific local Rayleigh number (Ra). Randomly and independently Ra values are generated from a uniform probability distribution defined on the interval [10, 1000]. In the forward modeling problem, the objective is to estimate the temperature distribution ($T_{x,y}$) resulting from varying sets of Ra values in the nine blocks. In the inverse problem, we aim to employ temperature distribution maps to estimate the relevant Ra values in the nine blocks.

b) **Test case (2): Porous enclosure with inclined barrier layer.** In this test case, the square porous cavity is assumed to be homogeneous and isotropic with a known Ra value ($Ra = 2000$), but we consider an inclined impermeable barrier (with negligible thickness) at the center of the problem domain, as shown in [Fig. 1b](#). Barries are frequent in applications related to natural porous media [\[30\]](#). The barrier is a straight line defined by the coordinates of two points: $(x_b - d, 0.99)$ and $(x_b + d, 0.01)$ where $x_b = 0.5$ and d is randomly selected from a uniform probability distribution defined on the interval [0, 0.45]. In the forward problem, we tend to estimate the temperature distribution ($T_{x,y}$) resulting from specific values of barrier inclination as represented by d . In the inverse problem the goal is to es-

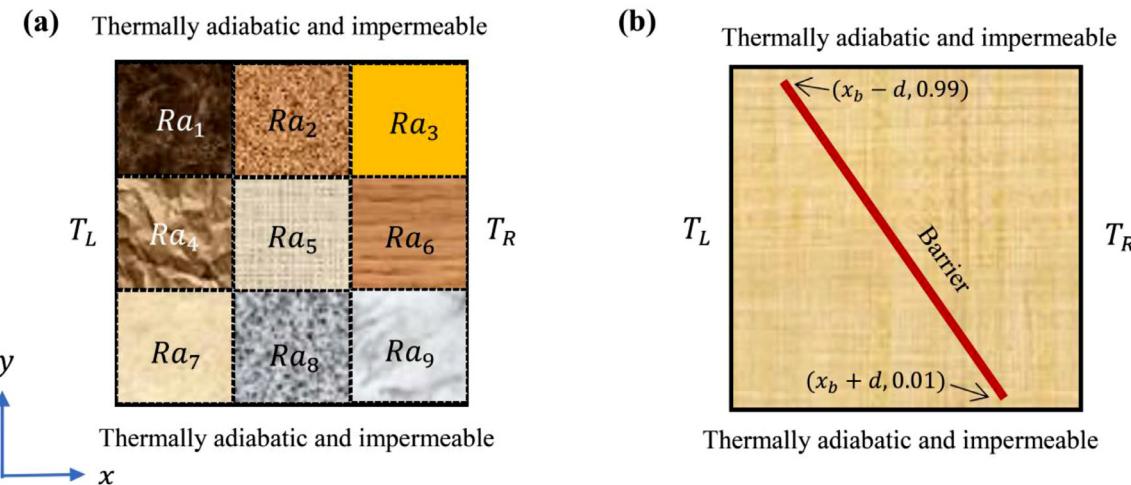


Fig. 1. Schematic diagram of the test case problems: (a) heterogeneous porous enclosure, and (b) porous enclosure with inclined barrier layer.

timate the value of d pertaining to a temperature distribution map.

Within the context of image-to-image regression, all model inputs and outputs are represented as images. So, for example, instead of a single value of d , we employ an image showing the location of the inclined barrier in the problem domain. While this may seem unnecessary for the simple examples above, it sets the stage for assessing the performance of ED-CNN in image reconstruction and high-dimensional problems.

2.2. Governing equations and the numerical model

NCPM is governed by the Darcy's law expressing linear momentum conservation, the continuity equation expressing the mass balance, and the heat transfer equation describing the energy balance. All these equations are coupled together with a variable fluid density which is a function of temperature. Under the Boussinesq approximation and with the assumptions cited in the previous section, the non-dimensional equations are giving as follows [4]:

$$\frac{\partial u^*}{\partial x^*} + \frac{\partial v^*}{\partial y^*} = 0 \quad (1)$$

$$u^* = -\frac{\partial p^*}{\partial x^*} \quad (2)$$

$$v^* = -\frac{\partial p^*}{\partial y^*} + Ra \cdot T^* \quad (3)$$

where u^* and v^* are the vertical and horizontal components of the dimensionless velocity, x^* and y^* are the dimensionless space coordinates, p^* and T^* are the dimensionless fluid pressure and temperature, and Ra is the local Rayleigh number given by:

$$Ra = \frac{k \cdot \rho_c \cdot \beta \cdot g \cdot \Delta T \cdot H}{\mu \cdot \alpha} \quad (4)$$

where k (m^2) is the hydraulic conductivity, ρ_c (kg/m^3) is the fluid density at reference temperature (T_R), β ($1/\text{K}$) is the fluid thermal expansion, $g(\text{m}/\text{s}^2)$ is the gravity, ΔT (K) is the temperature gradient between the hot and cold walls (i.e. $T_L - T_R$), $H(\text{m})$ is the square size, $\mu(\text{kg}/\text{m.s})$ is the fluid viscosity and $\alpha(\text{m}^2/\text{s})$ is the medium equivalent thermal diffusivity. The scales used in the non-dimensional analysis are H for the length, H/α for the velocity, $(T - T_R)/\Delta T$ for the temperature and $k/\mu\alpha$ for the pressure.

A finite element model is developed in COMSOL to solve these equations and generate image data. This has been done by coupling the 'Darcy's law' and 'Heat transfer in Porous Media' COMSOL modules and by considering thermally-induced density gradient. The barrier has been included using the options 'thin barrier' in the 'Darcy's law' module. The developed COMSOL model has been verified by comparison to a Fourier series solution and to an in-house developed code.

2.3. Convolutional neural networks (CNNs)

The name of CNN comes from an algebra operation called *convolution* that replaces the general matrix multiplication found in feedforward neural networks in at least one of the CNN layers [31]. The use of convolution is based on the assumption that the data has a grid-like structure, and that the local patterns carry important information [21,32]. The convolutional layers apply kernels, which are matrices typically square in dimensionality and of a smaller size than the input, to overlapping parts of the input image. The convolution operation is the dot product of the kernel weights and the values of the matching input areas. Each convolution produces a single number, and by skimming the entire input image a feature map is produced [33,22]. CNN architectures typically use a sequence of convolutional layers to detect a hierarchy of features, and then employ one or more fully connected layers to map the convolutional activations to the outcome classes (in classification problems) or values (in regression). Adaptation to high-dimensional problems is more prominent in CNNs compared to fully connected DNNs because the convolutional layers use kernels with fewer sparsely-connected weights compared to fully-connected layers. Hence, fewer parameters should be estimated during the training phase [17,34], making the training process easier and decreasing the potential for overfitting [25]. Moreover, CNNs provide enhanced capabilities at digesting large training datasets [16,13]. Details about the underlying theory of CNN can be found in Gu [35] and Jiang et al. [33].

2.4. Encoder-decoder CNNs

ED-CNN models have further improved performance on pixel-wise image-to-image learning tasks compared to classic CNNs [36]. ED-CNNs consist of an encoder subnetwork used for feature extraction, followed by a decoder subnetwork employed for spatial resolution restoration [37]. In the encoder CNN, the input image goes through a contracting process for down-sampling, where features

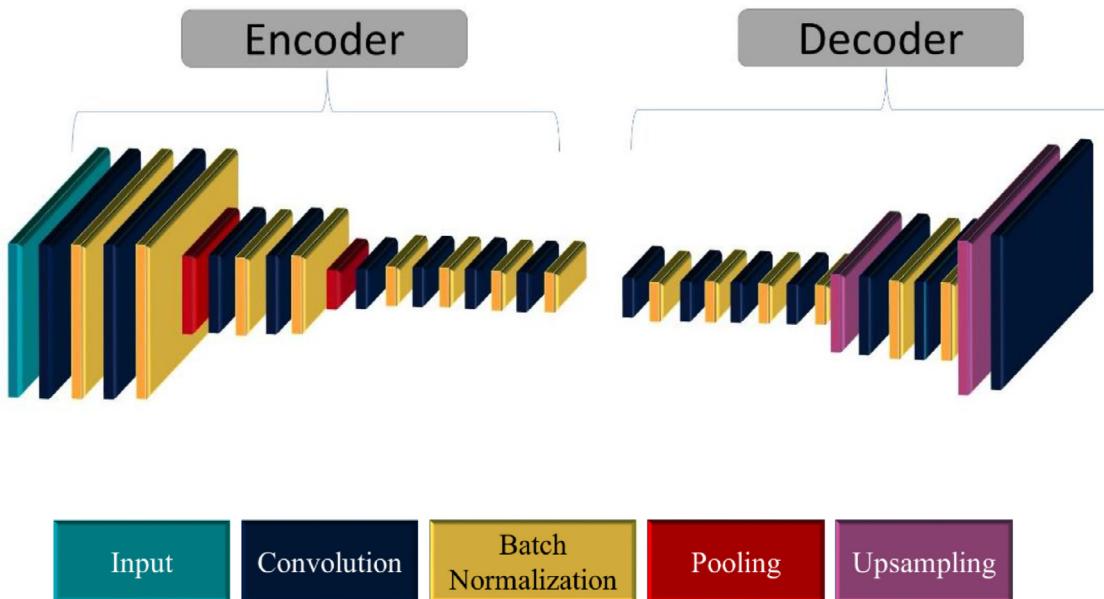


Fig. 2. Schematic of proposed ED-CNN.

of increasing semantic depth and decreasing spatial resolution are learned from the input image. At the end, the encoder subnetwork generates a number of feature maps which can be realized as coarse representation of the input image. The decoder CNN, which usually has the same network architecture as the encoder CNN but in opposite orientation, recovers the spatial resolution lost at the encoder through deconvolution and up-sampling and employs the feature maps from the encoder to constructed the output map [18]. For further details about ED-CNNs the interested reader is referred to Zhu and Zabaras [18] and Tahmasebi et al. [38].

2.5. ED-CNN training and validation

The input and output data for ED-CNN training and validation are transformed into and stored as images in greyscale PNG format using the python packages Numpy and Matplotlib (note that throughout the paper, these figures are presented in full color for visual appeal). For the heterogeneous domain case, the dataset consists of pairs of Rayleigh (as numerical model inputs) and temperature distribution (as numerical model output) images, and for the inclined barrier case, it consists of pairs of barrier location (as numerical model inputs) and temperature distribution (as numerical model output) images. All images are resized to the resolution of 32×32 to reduce computation complexity. For each test case, the data has been divided into training, validation, test, and uncertainty propagation sets. All four sets are sampled using the Latin hypercube sampling technique. The networks are trained with a maximum of 1000 input-output image pairs, and the validation and test sets contain 300 and 200 realizations, respectively. The neural network model is implemented in Python using Tensorflow and Keras. The performance evaluation metrics for the trained ED-CNNs are the root mean squared error (RMSE) and coefficient of determination (R^2 -score), calculated as follows [39,15]:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2} \quad (5)$$

$$R^2 = 1 - \frac{SS_{Reg}}{SS_T} = 1 - \frac{\sum_{i=1}^N (y_i - \hat{y}_i)^2}{\sum_{i=1}^N (y_i - m_i)^2} \quad (6)$$

Where SS_{Reg} and SS_T denote sum of the squared regression error and sum of the squared total error, respectively, y_i is numerical solution of the governing equation (i.e. Eqs. (1)–(3)) \hat{y}_i is the ED-CNN prediction, and m_i represents the mean value of numerical model for the N test data samples.

3. Results and discussion

3.1. The proposed ED-CNN architecture

Several candidate network architectures were initially trained, and the network configuration with the highest R^2 -score was selected. An illustration of the chosen ED-CNN is presented in Fig. 2, and the architecture is described in the following, with more details provided in Table 2. For both test cases, metamodeling is preformed using an ED-CNN consisting of 15 convolutional layers, all layers having a kernel size of 3×3 . Each convolutional layer is followed by a batch normalization (BN) layer which regularizes the network while reducing error and accelerating the network's performance (Zhu and Zabaras [18]). In the encoder subnetwork, the input image is down-sampled two times using pooling layers with a kernel size of 2×2 and zero padding. In the decoder subnetwork, two up-sampling layers with the same kernel size are used to rebuild the input image to its initial size of 32×32 .

A Rectified linear unit ('ReLU') activation function is utilized in all layers, and the mean squared error is employed as the loss function. The Root Mean Square Propagation ('RMSprop') algorithm [40] is used for ED-CNN model training. The batch size is chosen to be 24, the maximum number of epochs is 300, and the learning rate is 0.0001. The above described network architecture is also used to develop an optimizer for both test cases, but with two modification: (a) the sigmoid activation function is used for the last three layers of the decoder subnetwork, and (b) the Adam optimizer, a variant of stochastic gradient descent, is employed for model training. Details on the hyperparameter settings are presented in Table 3. ED-CNN training is implemented on Intel Core i7-9700(3 GHz) CPU which requires about 420 s for training 100 epochs, when the training data size varies from 100 to 1000.

Table 2
Summary of proposed ED-CNNs architectures.

Layer	Metamodel		Optimizer	
	N_{filter} , kernel size, pad, activation	Resolution	N_{filter} , kernel size, pad, activation	Resolution
Encoder				
Input	1, 3 × 3, same, ReLU	32 × 32	1, 3 × 3, same, ReLU	32 × 32
Conv1	32, 3 × 3, same, ReLU	32 × 32	32, 3 × 3, same, ReLU	32 × 32
BN1	32, 3 × 3, -, -	32 × 32	32, 3 × 3, -, -	32 × 32
Conv1	32, 3 × 3, same, ReLU	32 × 32	32, 3 × 3, same, ReLU	32 × 32
BN1	32, 3 × 3, -, -	32 × 32	32, 3 × 3, -, -	32 × 32
Pool1	32, 2 × 2, -, -	16 × 16	32, 2 × 2, -, -	16 × 16
Conv2	64, 3 × 3, same, ReLU	16 × 16	64, 3 × 3, same, ReLU	16 × 16
BN2	64, 3 × 3, -, -	16 × 16	64, 3 × 3, -, -	16 × 16
Conv2	64, 3 × 3, same, ReLU	16 × 16	64, 3 × 3, same, ReLU	16 × 16
BN2	64, 3 × 3, -, -	16 × 16	64, 3 × 3, -, -	16 × 16
Pool2	64, 2 × 2, -, -	8 × 8	64, 2 × 2, -, -	8 × 8
Conv3	128, 3 × 3, same, ReLU	8 × 8	128, 3 × 3, same, ReLU	8 × 8
BN3	128, 3 × 3, -, -	8 × 8	128, 3 × 3, -, -	8 × 8
Conv3	128, 3 × 3, same, ReLU	8 × 8	128, 3 × 3, same, ReLU	8 × 8
BN3	128, 3 × 3, -, -	8 × 8	128, 3 × 3, -, -	8 × 8
Conv4	256, 3 × 3, same, ReLU	8 × 8	256, 3 × 3, same, ReLU	8 × 8
BN4	256, 3 × 3, -, -	8 × 8	256, 3 × 3, -, -	8 × 8
Conv4	256, 3 × 3, same, ReLU	8 × 8	256, 3 × 3, same, ReLU	8 × 8
BN4	256, 3 × 3, -, -	8 × 8	256, 3 × 3, -, -	8 × 8
Decoder				
Conv5	128, 3 × 3, same, ReLU	8 × 8	128, 3 × 3, same, ReLU	8 × 8
BN5	128, 3 × 3, -, -	8 × 8	128, 3 × 3, -, -	8 × 8
Conv5	128, 3 × 3, same, ReLU	8 × 8	128, 3 × 3, same, ReLU	8 × 8
BN5	128, 3 × 3, -, -	8 × 8	128, 3 × 3, -, -	8 × 8
Conv6	64, 3 × 3, same, ReLU	8 × 8	64, 3 × 3, same, ReLU	8 × 8
BN6	64, 3 × 3, -, -	8 × 8	64, 3 × 3, -, -	8 × 8
Conv6	64, 3 × 3, same, ReLU	8 × 8	64, 3 × 3, same, ReLU	8 × 8
BN6	64, 3 × 3, -, -	8 × 8	64, 3 × 3, -, -	8 × 8
Up1	64, 2 × 2, -, -	16 × 16	64, 2 × 2, -, -	16 × 16
Conv7	32, 3 × 3, same, ReLU	16 × 16	32, 3 × 3, same, sigmoid	16 × 16
BN7	32, 3 × 3, -, -	16 × 16	32, 3 × 3, -, -	16 × 16
Conv7	32, 3 × 3, same, ReLU	16 × 16	32, 3 × 3, same, sigmoid	16 × 16
BN7	32, 3 × 3, -, -	16 × 16	32, 3 × 3, -, -	16 × 16
Up1	32, 2 × 2, -, -	32 × 32	32, 2 × 2, -, -	32 × 32
Conv (output)	3, 3 × 3, same, ReLU	32 × 32	3, 3 × 3, same, sigmoid	32 × 32

Table 3
Hyperparameters values for metamodel and optimizer networks.

Hyperparameter	Metamodel	Optimizer
Optimizer	RMSprop	Adam
Loss function	Mean squared error	Mean squared error
Samples	1000	1000
Test samples	200	200
Validation split	0.33	0.33
Batch size	24	12
epochs	100	300
Learning rate	0.0001	0.0001

3.2. ED-CNN as metamodel

In this subsection, we assess the performance of the constructed ED-CNN metamodels through the use of both statistical diagnostics and graphical methods. Fig. 3a,c shows RMSE decay with the number of epochs during the training process. It can be inferred from the RMSE plots that 60 epochs could be sufficient to reach a stable solution in both cases. To illustrate the convergence of the approximation error with respect to the training sample size, we generate four sample sets with $N = 100, 300, 700, 1000$ model evaluations for each of the two test cases. Fig. 3b illustrates the decay of the R^2 -score for the first test case based on 100 epochs. It is apparent from the R^2 -score plots that increasing the number of samples from 100 to 300 significantly affects the accuracy of the model by increasing the R^2 -score from 0.722

to 0.915, while using additional training samples only slightly improves the R^2 -score. The R^2 score plot for the inclined barrier case (Fig. 3d) shows a similar pattern. This shows that the ED-CNN can be trained accurately with a relatively small number of training data. The maximum R^2 -score obtained from using a training sample size of 1000 (with 100 epochs) is 0.941 and 0.980 for the two test cases, respectively.

An exemplary ED-CNN-based estimation of the temperature distribution is shown in Fig. 4c for the heterogeneous porous enclosure test case. This prediction is based on the Rayleigh realization shown in Fig. 4a (belonging to the test dataset). For comparison, the reference temperature distribution is also shown in Fig. 4b. As demonstrated, the ED-CNN results reflect the general behavior of heat transfer obtained from numerical simulation. As expected, the isotherms are circular around the center point as they are affected by the circulation of the fluid saturating the porous media. Fig. 4d demonstrates the absolute error of the ED-CNN temperature estimates. The error is generally low, and has its maximum in the center of the domain. For the second test case, similar images are provided in Fig. 4e-h. As demonstrated, the ED-CNN metamodel provides a good approximation of the temperature distribution resulting from the presence of the inclined barrier, as the errors are generally low. In general, Fig. 4 shows that the ED-CNN-based metamodels are able to learn the characteristics of the physical flow phenomenon of NCPM.

In order to provide an overview of the ED-CNN prediction error across the entire test dataset, we calculated the RMSE for each of the 200 test sample images. The resulting error histograms are pre-

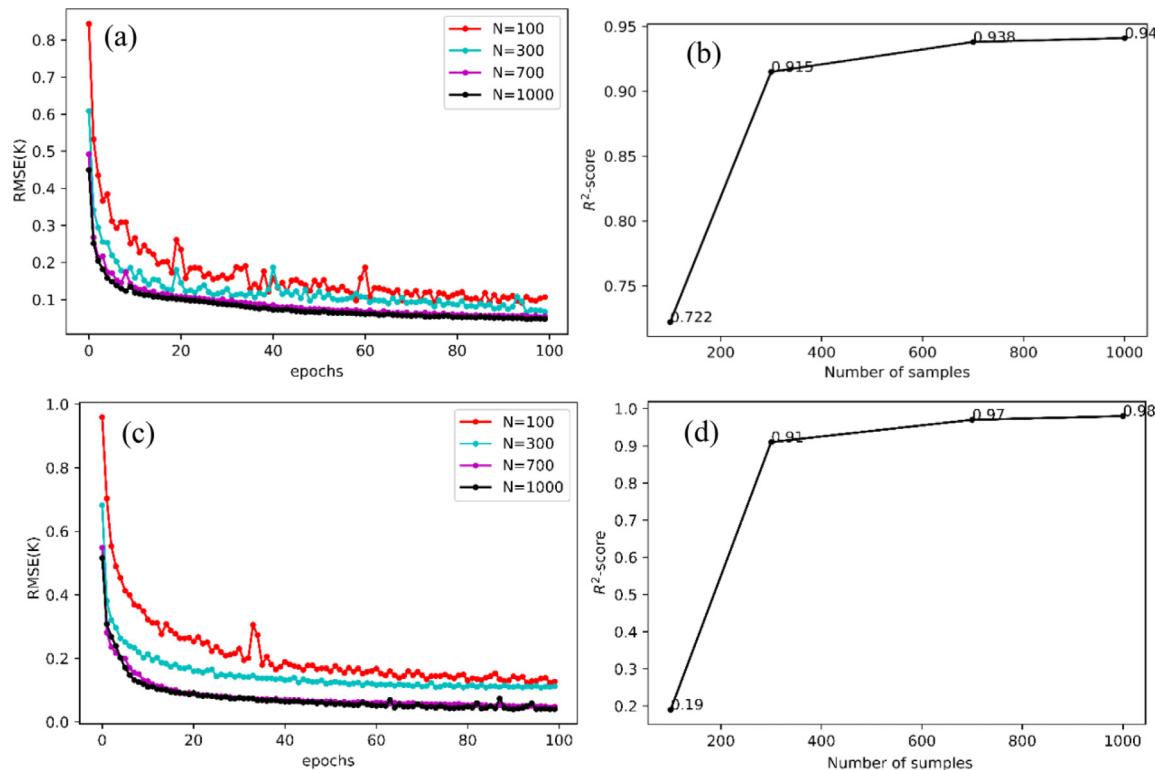


Fig. 3. R^2 -score and RMSE of the proposed ED-CNN for metamodeling in: (a,b) heterogeneous porous enclosure test case, and (c,d) porous enclosure with inclined barrier layer.

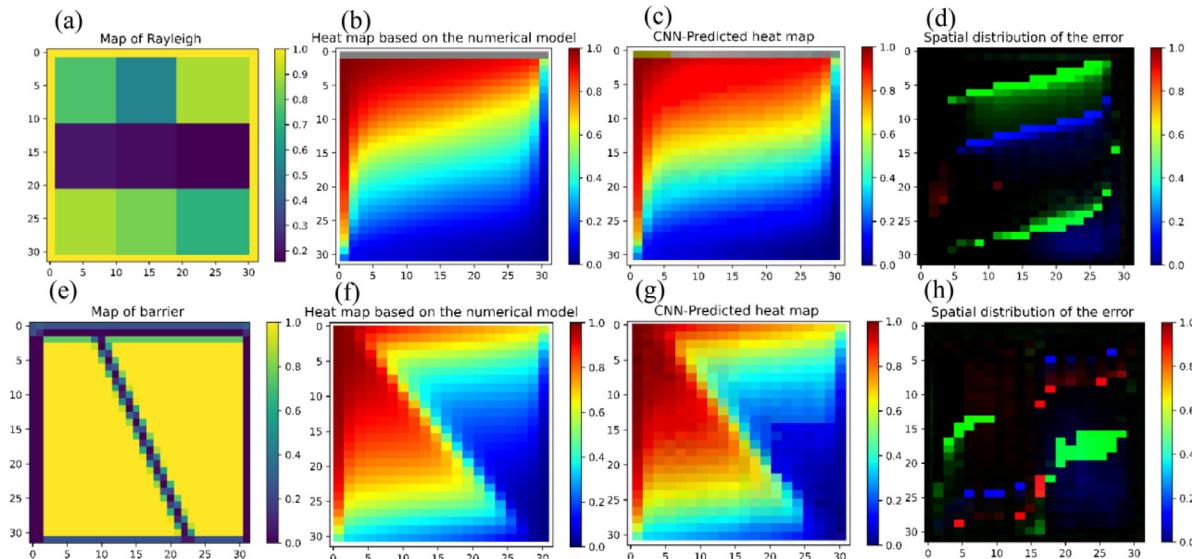


Fig. 4. Comparison of numerical model and ED-CNN metamodel temperature outputs for the metamodel: (a) heterogeneous porous enclosure test case, and (b) porous enclosure with inclined barrier layer.

sented in **Fig. 5a** and **b** for our two test cases. In the first test case, most of the predictions are covered by the 0.155 error threshold, a significant part of which is below 0.095. For the inclined barrier more than 99% of the RMSEs are less than 0.18. The mean RMSE of the test images is 0.0752 and 0.0475 for the two test cases, respectively. We also calculate the total RMSE for each pixel of the domain based on the test datasets. The resulting error distribution maps are depicted in **Fig. 5c** and **d** for the two test cases. For the heterogeneous enclosure the maximum RMSE occurs at the center of the domain where the temperature gradient is at its highest.

This is similar to the pattern observed for the absolute error in the exemplary **Fig. 4d**. For the inclined barrier case, the maximum RMSE is observed around the boundary of the barrier layer. Since the inclined barrier location varies for the various realizations according to **Fig. 1b**, this results in a higher error region consisting of two triangles joined at their apexes.

The *confusion matrix* is subsequently employed to visualize the classification performance of the ED-CNN-based metamodels. In order to build the confusion matrix, continuous temperature values have been converted to 10 classes at 0.1 width. The resulting con-

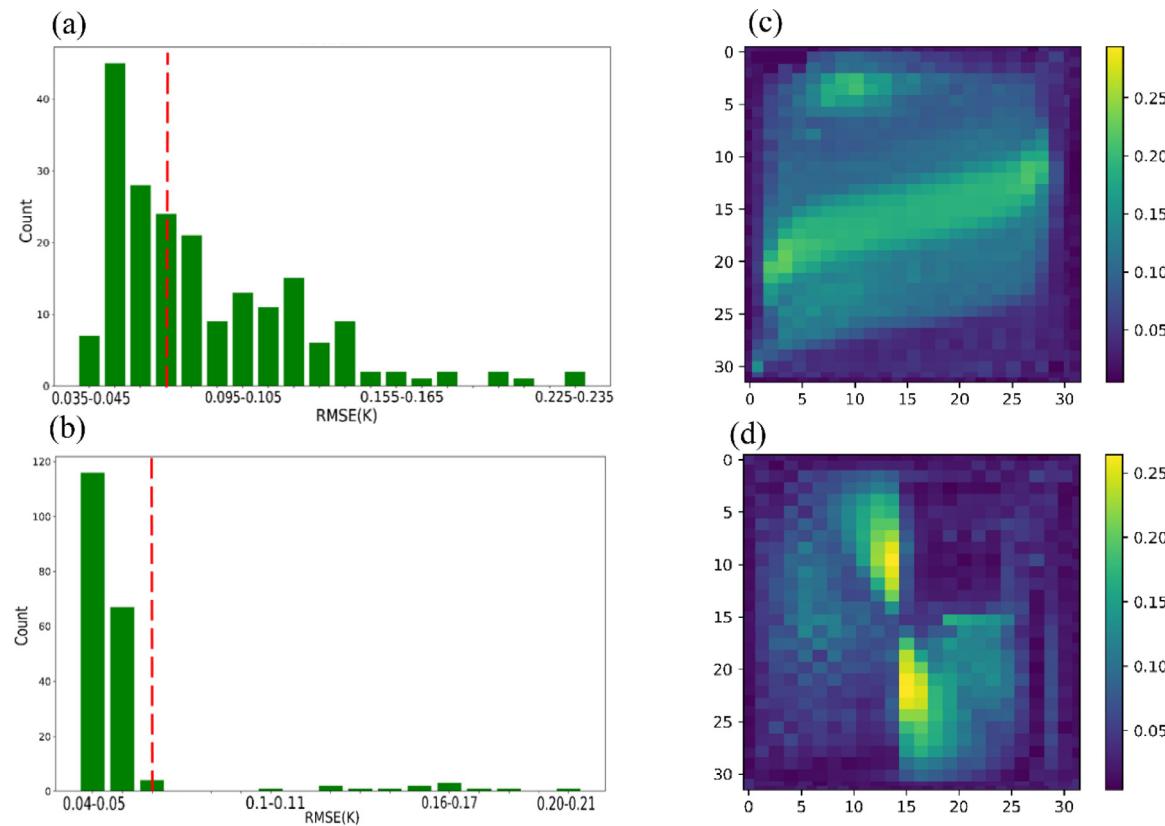


Fig. 5. (a,b) RMSE histograms for the ED-CNN metamodels, with errors pertaining to the exemplary estimations of Fig. 4 indicated by the red dashed lines, and (c,d) RMSE distribution maps for the two test cases (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

fusion matrices in [Fig 6](#) give a summary result of the expected against the predicted outcomes. The diagonals of these multi-class confusion matrices show the percentage of correct classifications for each class by the ED-CNN, and other entries show instances where a sample from one class was mistakenly assigned a different class label. The confusion matrices show that the test samples are accurately classified. This is confirmed with values of 58.32%, 64.89%, 58.3%, and 60% for accuracy, precision, recall, and F1-score, respectively for the first test case, and 61.35%, 75%, 61.3%, and 61.7% for the same metrics (in similar order) in the second test case.

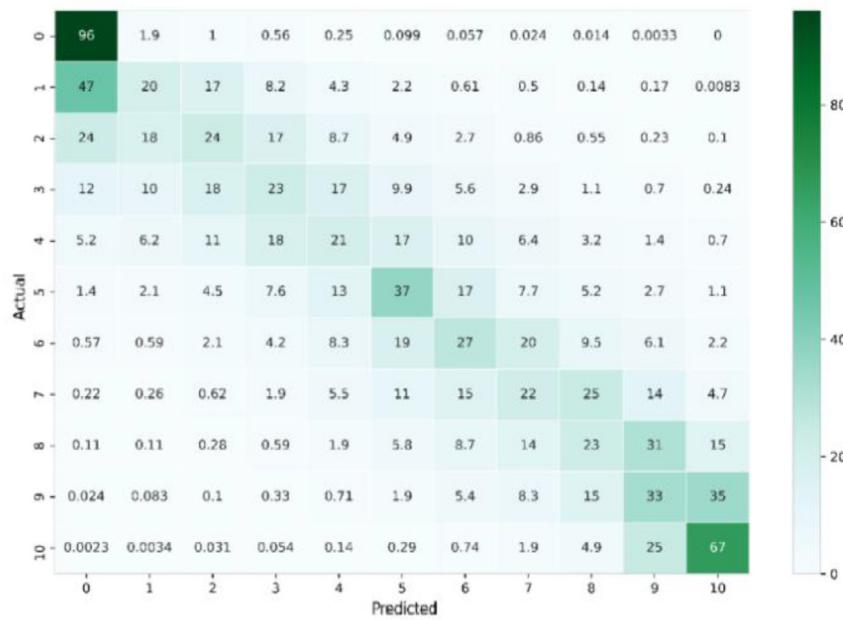
The *maximum image gradient* (IG_{max}), which is the maximum directional change in color intensity, is also calculated for Rayleigh input images of the heterogeneous test case. The objective is to understand how the intensity of variations in the Rayleigh input values affects the accuracy of ED-CNN predictions. The IG_{max} value for each Rayleigh input image in the test dataset has been plotted against the RMSE of the same image in the scatterplot of [Fig. 7](#). As demonstrated, the points are highly scattered.

To analyze the performance of the proposed ED-CNN-based metamodel in a more complex setting, we increase the number of blocks in the heterogeneous porous enclosure test case to 25 blocks with Ra values generated from the previously described probability distribution. In this new test case, acceptable results are obtained by increasing the total number of input-output image pairs used for training to 5000 and the number of training epochs to 500, and resizing the images to the resolution of 64×64 . The maximum R^2 -score obtained in this test case is 0.92. The exemplary ED-CNN-based estimation of the temperature distribution shown in [Fig. 8](#), demonstrates that the metamodel is still capable of capturing the general behavior of heat transfer with low error.

3.3. ED-CNN for uncertainty propagation analysis (UPA)

In this subsection, we test our ED-CNN-based metamodel's effectiveness and efficiency in addressing UPA tasks for NCPM problems. As the benchmark, we employ 1000 independent realizations of the inputs (i.e. Ra and d for the two test cases, respectively) in the context of numerical model-based Monte Carlo simulations (MCSs), to obtain the mean (μ_T) and standard deviation (σ_T) maps of the output temperature. The numerical models of the two test cases are subsequently replaced with the ED-CNNs metamodels, and MCSs are repeated with the same number of realizations. The resulting spatial distribution of μ_T and σ_T for the numerical models and the ED-CNNs are compared in [Fig. 9](#). The error maps obtained from pixel by pixel calculations of the mean difference between the benchmark and predicted temperature values are also shown in [Fig. 9](#). As illustrated, the ED-CNN-based μ_T and σ_T are in acceptable agreement with the benchmark maps for both test cases, though the agreement between maps of numerical model vs. ED-CNNs metamodel mean values ([Fig. 9a vs b](#), and [Figs. 9g vs 8h](#)) is considerably better compared to the standard deviation maps ([Fig. 9d vs e](#), and [Figs. 9j vs 8k](#)). The maximum error in μ_T and σ_T estimation for the first test case are observed near the high temperature gradient region around the center of the problem domain, though the error signs differ for μ_T and σ_T in upper and lower bands around the center. For the second test case, maximum errors occur around the barrier in a similar way for both μ_T and σ_T . Maximum absolute errors are equal to 0.161 and 0.125 for μ_T , and 0.051 and 0.105 for σ_T estimations of the two test cases, respectively. To assess the computational time saving resulting from the use of ED-CNN in UPA, we define the time saving

(a)



(b)

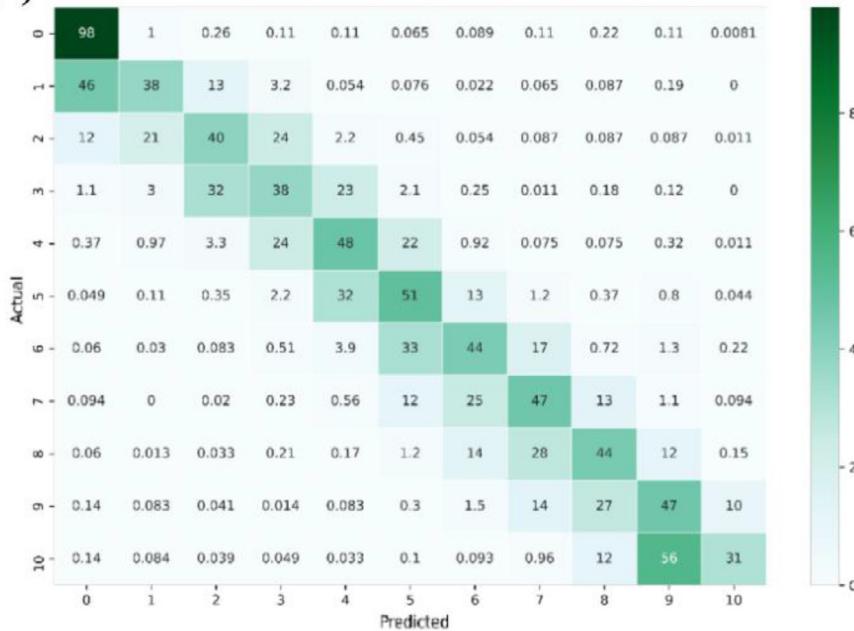


Fig. 6. Normalized confusion matrices for the metamodel of (a) heterogeneous porous enclosure test case, and (b) porous enclosure with inclined barrier layer.

ratio (TSR) as [41]:

$$TSR = \frac{t_s - t_m}{t_s} \quad (7)$$

where t_s is the time required to perform a set of calculations using the numerical simulator, and t_m is the time required to perform the same set of calculations using the metamodel. The later includes the computational time required for training and metamodeling, but excludes the computational time employed for generation of the training dataset. The TSR for ED-CNN in UPA is 0.981 and 0.985 for the two test cases, respectively, which is high for an image-to-image regression task with large numbers of inputs/outputs. This TSR is particularly

appealing in applications that require repeated UPA, such as uncertainty-based simulation-optimization and global sensitivity analysis.

3.4. ED-CNN as optimizer

In this subsection, we evaluate the performance of ED-CNNs for input parameter estimation (i.e. as full replacement of an inverse model). **Fig. 10a,c** shows RMSE decay with the number of epochs during the training process for the two test cases. **Fig. 10b,d** demonstrates improvement in the R^2 -score with increasing sizes of the training dataset. For the heterogeneous enclosure, all 1000 realizations are required to predict the Rayleigh number distribution

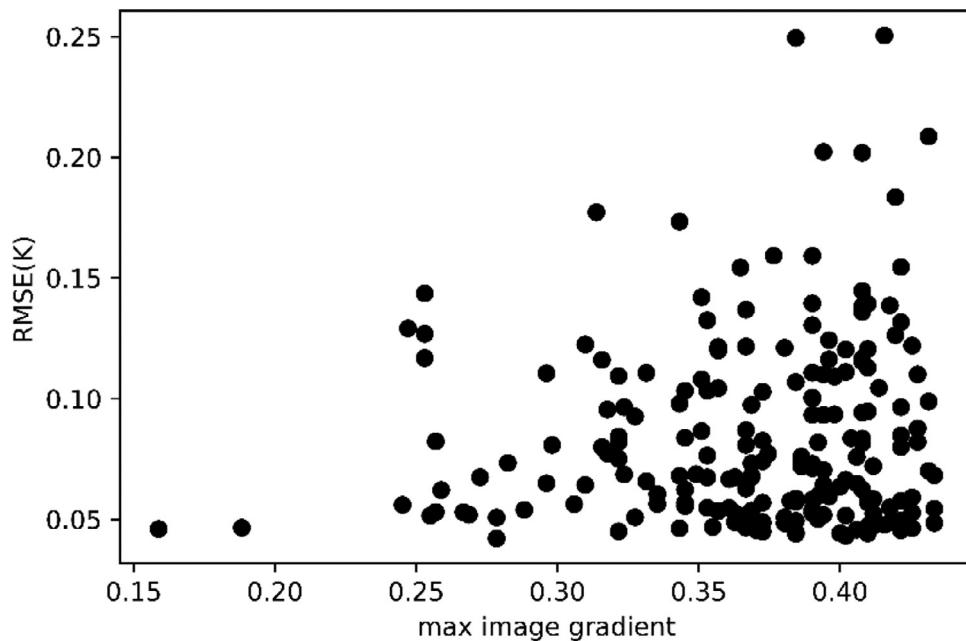


Fig. 7. Scatterplot of maximum Rayleigh image gradients vs. RMSEs for the heterogeneous porous enclosure test case.

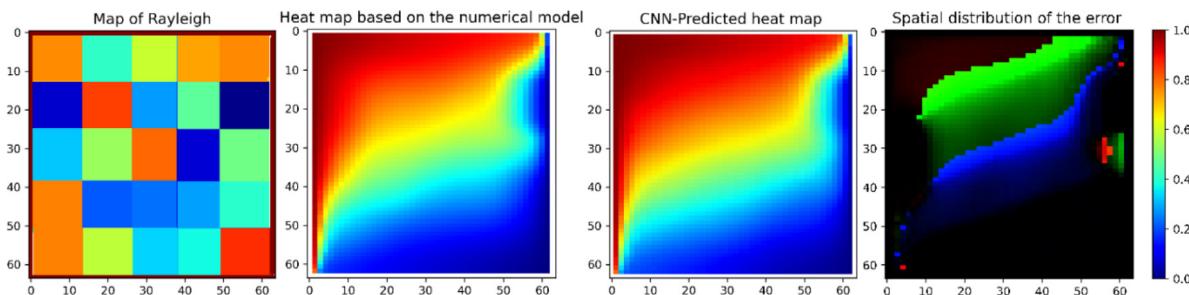


Fig. 8. Comparison of numerical model and ED-CNN metamodel temperature outputs for the heterogeneous porous enclosure test case with the number of blocks increased to 25.

from the heat map with a R^2 -score of above 0.9. For this number of training points, the network starts to stabilize at around 200 epochs. For the barrier case, the ED-CNN is able to predict the images with a R^2 -score of 0.97 using only 100 training samples. Note that for the first test case, the inverse problem involved estimation of nine unknown input parameters (Ra_1, Ra_2, \dots, Ra_9), while for the second test case, it only involves the estimation of one parameter (d).

An exemplary ED-CNN-based estimation of the Rayleigh distribution from an input temperature map is shown in Fig. 11c, with the actual Rayleigh map shown in Fig. 11b. Fig. 11d demonstrates the absolute error of the ED-CNN-based Rayleigh estimates. Errors are mostly in between -0.5 and 0.41 for the various zones, and the ED-CNN has managed to predict the general outline of the Ra distribution. A similar set of exemplary figures are provided for the porous enclosure with inclined barrier layer in Fig. 11e,f. In this case, the ED-CNN-based estimation of the barrier layer location is highly accurate, and the model has been able to find the barrier location using the input temperature distribution.

The RMSE histogram of the ED-CNN optimizers are shown in Fig. 12a,b. For the first test case, errors are spread in a wide interval, and the highest frequency of error is observed in the interval between 0.025 and 0.080. For the second test case, error are generally lower, and in the interval between 0 and 0.05.

The mean RMSE of the test images is 0.068 and 0.0162 for the two test cases, respectively. The spatial distributions of the total RMSE (based on the test datasets) are shown in Fig. 12c,d. For the heterogeneous enclosure, we have nine color blocks pertaining to the nine estimated Rayleigh values. The RMSE is significantly higher at the center of the domain, notably because the output temperature distribution is least sensitive to the Rayleigh number in the central block. For the inclined barrier problem, errors are only meaningful on the displacement interval of the barrier. In this region, the maximum RMSE for the estimation of barrier location pertains to the case in which the barrier layer is vertical.

When the number of blocks in the heterogeneous porous enclosure test case is increased to 25, the ED-CNN-based optimizer is still capable of estimating Ra values with acceptable accuracy. This is illustrated in the exemplary maps of Fig. 13. However, to obtain acceptable results (with a R^2 -score of 0.89) the number of training sample and epochs are increased to 5000 and 500, respectively. We are not able to obtain acceptable results from the ED-CNN-based optimizer when the number of blocks are significantly increasing beyond the value of 25, notably because variations in the resulting steady-state temperature distribution are not enough to capture the input Ra distribution, and transient numerical model outputs are required for optimizer training.

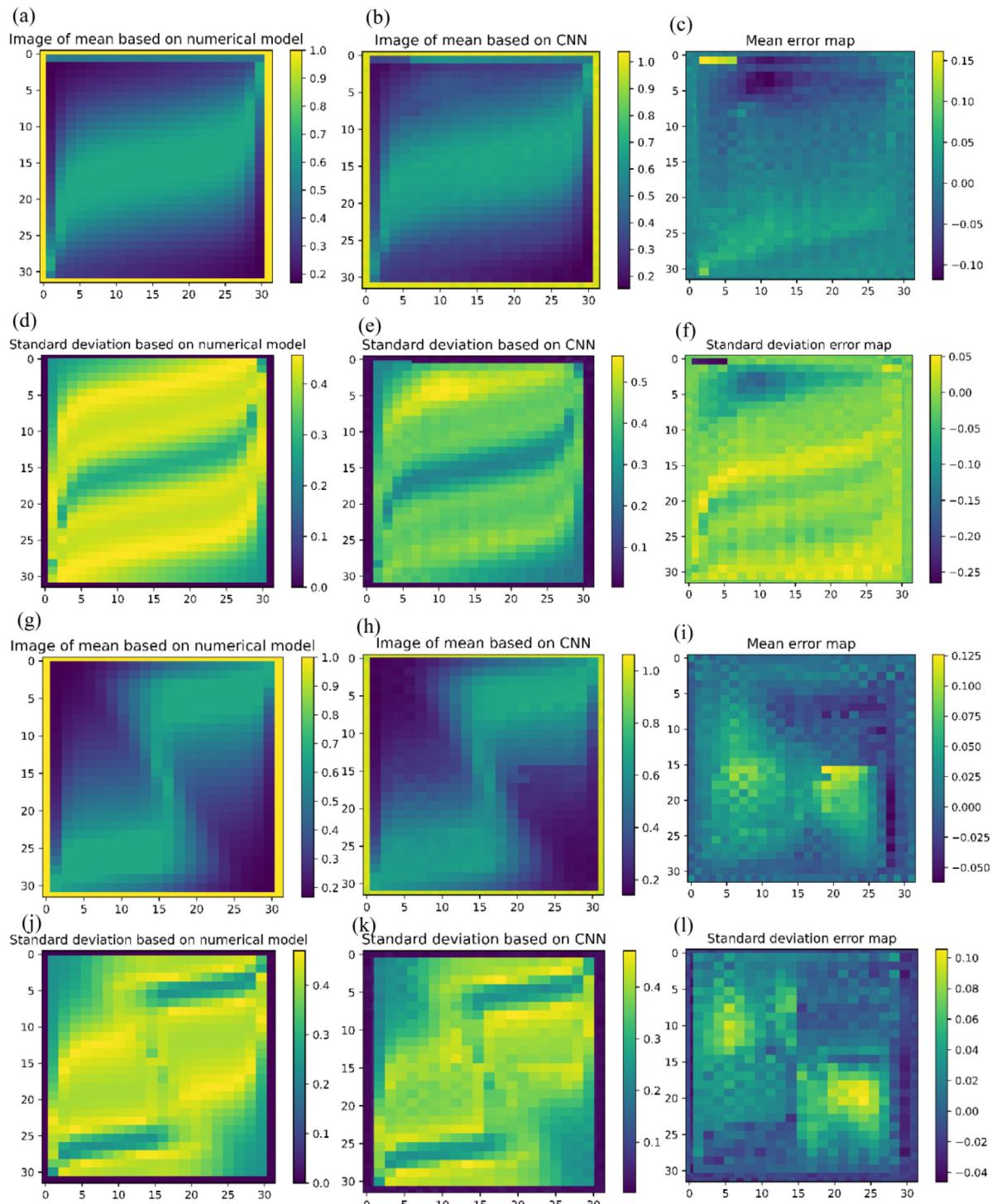


Fig. 9. Spatial distribution of the mean, standard deviation and their estimation errors for temperature in: (a–f) heterogeneous porous enclosure test case, and (g–l) porous enclosure with inclined barrier layer.

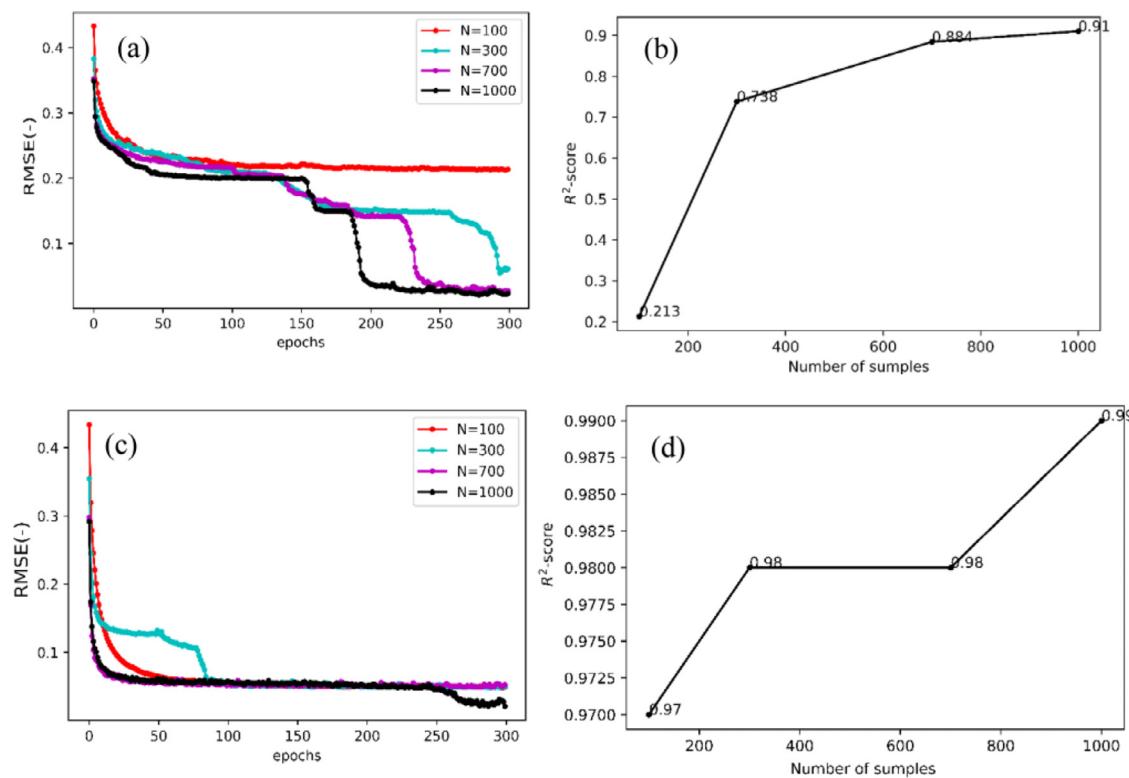


Fig. 10. R^2 -score and RMSE of the proposed ED-CNN in parameter estimation (optimizer case) in: (a,b) heterogeneous porous enclosure test case, and (c,d) porous enclosure with inclined barrier layer.

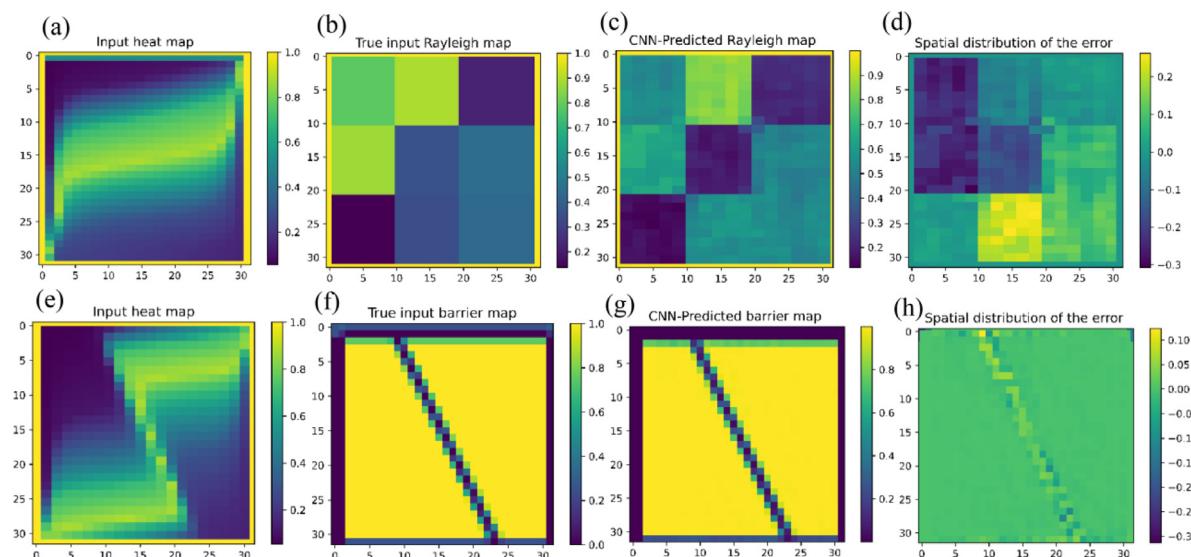


Fig. 11. Comparison of actual and ED-CNN estimates of: (a-d) Rayleigh distribution in the heterogeneous porous enclosure test case, and (e-h) barrier location in the porous enclosure with inclined barrier layer.

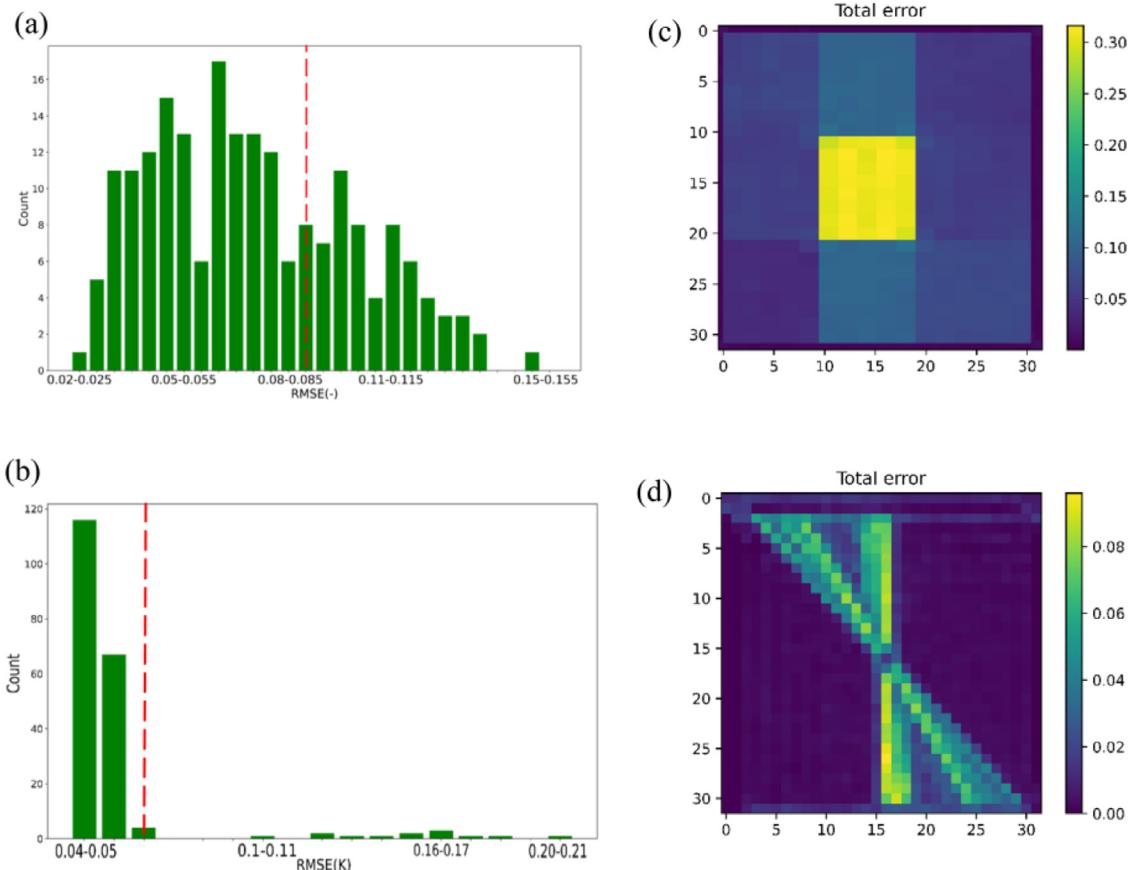


Fig. 12. (a,b) RMSE histograms for the optimizer ED-CNNs, with errors pertaining to the exemplary estimations of Fig. 10 indicated by the red dashed lines, and (c,d) RMSE distribution maps for the two test cases (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

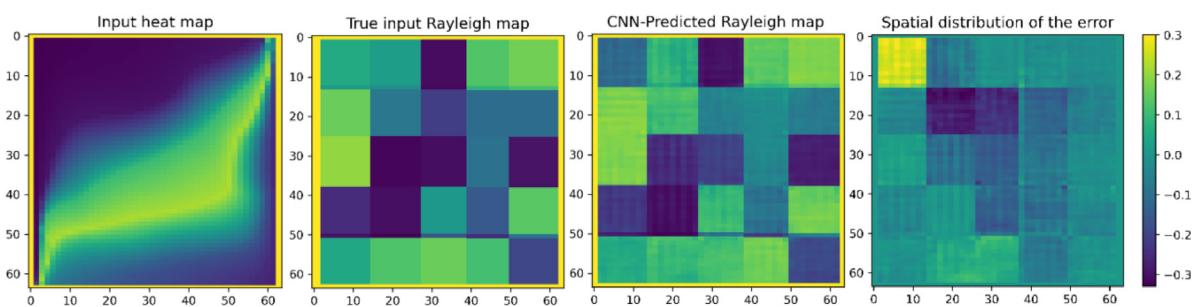


Fig. 13. Comparison of actual and ED-CNN estimates of the Rayleigh distribution in the heterogeneous porous enclosure test case with the number of blocks increased to 25.

3.5. Comparison with previous studies

Table 4 shows the number of training samples, epoch size, optimization technique, activation function type and the training computational time in previous DNN applications in the context of flow and transport in porous media, and compares them with those of the current study. The number of training samples are on the same scale in the various studies (with the exception of [29]), and generally range between 400 and 1600. The number of epochs are rather similar, and all studies rely on the 'ReLU' activation function (in some cases in conjunction with 'softplus', 'sigmoid' or 'tanh'), and the 'Adam', 'SGD', or 'RMSprop' optimization techniques. The training time for our ED-CNN is at the low end of similar studies. This can be attributed to the efficient network architecture that is built after several trials and errors.

4. Conclusion

The investigation of the relevance of deep learning to engineering problems is currently a hot topic. The transfer of these techniques into operational tools requires more applications to different engineering problems, testing it on common benchmarks, and interdisciplinary collaboration with the deep learning community [16]. This work is framed in this context. We employ an ED-CNN architecture for image-to-image regression in the context of natural convection in porous media, based on the benchmark of a porous square cavity. The developed model was able to provide relatively good approximations with limited training samples, effectively solving the curse of dimensionality problem associated with high dimensional input and output spaces. Within the context of the proposed methodology, image datasets obtained from numerical modeling, high-resolution imaging and non-destructive scanning techniques can be employed to train ED-CNNs for the estimation of temperature distribution due to natural convection (in forward modeling), or to estimate the characteristics of porous media from a specific temperature distribution (in inverse modeling). Our key results can be summarized as follows:

- 1 For two popular test cases (i.e. heterogeneous square enclosure, and porous enclosure with inclined barrier layer) approximately 300 training realizations and 60 training epochs could be sufficient to reach a stable solution.
- 2 The ED-CNN results reflect the general behavior of heat transfer obtained from numerical simulation. For the heterogeneous enclosure the maximum error occurs at the center of the domain where the temperature gradient is at its lowest, and the temperature is insensitive to Ra . For the inclined barrier case, the maximum error is observed around the boundary of the barrier layer.
- 3 For the heterogeneous enclosure, 1000 training realizations and at least 200 training epochs are required to predict the Rayleigh number distribution from the heat map with a R^2 -score of above 0.9. For the barrier case, the ED-CNN is able to predict the location of the barrier with a R^2 -score of 0.97 using only 100 training samples.
- 4 For the inverse heterogeneous enclosure problem, the Rayleigh estimation error is significantly higher at the center of the domain, notably because the output temperature distribution is less sensitive to the Rayleigh number in the central block. For the inclined barrier problem, the maximum error in the estimation of barrier location pertains to the case in which the barrier layer is vertical.
- 5 The suggested ED-CNN technique can be effectively used as meta-model in uncertainty analysis studies.

The ED-CNN model presented here can be easily extended to multiple input/output images. This work shows the capacities of

Table 4
Different architectures used for porous media problems by different researchers.

Reference	Architecture	Objective	Total number of samples	Epochs	Optimization	Activation function	Training Time cost*
Mo et al. [21]	DCED	Contaminant Source Identification	1500	200	SGD	ReLU, softplus	110 min
Mo et al. [22]	DCED	Uncertainty Quantification of Dynamic Multiphase Flow in Heterogeneous Media	1600	200	SGD	ReLU, sigmoid	40 min
Mo et al. [29]	CAAE and DRDCN	Estimation of Non-Gaussian Hydraulic Conductivities	40,000	50	Adam	ReLU, Leaky ReLU, sigmoid	1.7 h
Jiang et al. [33]	Deep Residual U-net CNN	dynamic contaminant tracing prediction	1000	200	SGD	ReLU	230 min
Xiao et al. [42]	cRRB-U-Net	estimating the static parameters of geological reservoirs	1500	100	Adam	ReLU	25 min
Tang et al. [43]	Deep Residual U-net CNN	data assimilation using different history matching	400	260	Adam	ReLU	7 h
Kamirava et al. [44]	Deep Residual U-net CNN	fluid flow and transport prediction in complex porous membranes	800	205	SGD	ReLU	4 h
Zhang et al. [45]	VAE and GAN	stochastic reconstructions on porous media	N/A	480	Adam	Leaky ReLU, Tanh, sigmoid	2 and 5 h
Current metamodel study	CAE	Heat map prediction in heterogeneous porous media	1000	100	RMSprop	ReLU	9 min
Current optimizer study	CAE	Rayleigh number prediction in heterogeneous porous media	1000	300	Adam	ReLU, sigmoid	27 min

DCED: Dense Convolutional Encoder Decoder, SGD: Stochastic Gradient Descent, CAAE: Convolutional Adversarial AutoEncoder, DRDCN: Deep Residual Dense Convolutional Network, cRRB: Residual-in-Residual Dense, VAE: Variant AutoEncoder, GAN: Generative Adversarial Network, CAE: Convolutional AutoEncoder.

* Does not include the computational time required for the generation of the training dataset.

ED-CNN in modeling NCPM based on hypothetical cases. It provides a guidelines for the application of ED-CNNs in such context. Further work can be done by applying the developed technique on more realistic problems in different applications. Future development can also be directed towards: (a) modeling double diffusive natural convection, and (b) estimation of porous media characteristics from laboratory imaging of the heat map. The results of this paper showed that transient simulations would be useful in highly heterogeneous domains. Extending the application of the ED-CNN technique to deal with transient data could be an interesting topic.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Mohammad Mahdi Rajabi: Formal analysis, Writing – original draft. **Mohammad Reza Hajizadeh Javarani:** Investigation, Visualization, Writing – original draft. **Amadou-oury Bah:** Software, Investigation. **Gabriel Frey:** Conceptualization, Methodology. **Florence Le Ber:** Methodology, Conceptualization. **François Lehmann:** Methodology, Writing – review & editing. **Marwan Fahs:** Conceptualization, Writing – review & editing, Funding acquisition.

Acknowledgment

Mohammad Mahdi Rajabi and Marwan Fahs would acknowledge the support from the National School of Water and Environmental Engineering of Strasbourg (Grant No. 6903NNES).

References

- [1] D.A. Nield, A. Bejan, Convection in Porous Media, Springer International Publishing, Cham, 2017, doi:[10.1007/978-3-319-49562-0](https://doi.org/10.1007/978-3-319-49562-0).
- [2] Y. Su, J.H. Davidson, Modeling Approaches to Natural Convection in Porous Media, Springer International Publishing, Cham, 2015, doi:[10.1007/978-3-319-14237-1](https://doi.org/10.1007/978-3-319-14237-1).
- [3] M.M. Rajabi, B. Ataie-Ashtiani, C.T. Simmons, Polynomial chaos expansions for uncertainty propagation and moment independent sensitivity analysis of seawater intrusion simulations, *J. Hydrol. (Amst)* 520 (2015) 101–122.
- [4] N. Fajraoui, M. Fahs, A. Younes, B. Sudret, Analyzing natural convection in porous enclosure with polynomial chaos expansions: effect of thermal dispersion, anisotropic permeability and heterogeneity, *Int. J. Heat Mass Transf.* 115 (2017) 205–224.
- [5] M.M. Rajabi, M. Fahs, A. Panjehfouladgaran, B. Ataie-Ashtiani, C.T. Simmons, B. Belfort, Uncertainty quantification and global sensitivity analysis of double-diffusive natural convection in a porous enclosure, *Int. J. Heat Mass Transf.* 162 (2020) 120291.
- [6] D. Crevillen-Garcia, R.D. Wilkinson, A.A. Shah, H. Power, Gaussian process modeling for uncertainty quantification in convectively-enhanced dissolution processes in porous media, *Adv. Water Resour.* 99 (2017) 1–14.
- [7] Y. Varol, H.F. Oztop, E. Avci, Estimation of thermal and flow fields due to natural convection using support vector machines (SVM) in a porous cavity with discrete heat sources, *Int. Commun. Heat Mass Transf.* 35 (8) (2008) 928–936.
- [8] N.A. Ahamad, A. Athani, I.A. Badruddin, Heat transfer prediction in a square porous medium using artificial neural network, in: Proceedings of the AIP Conference, 1953, AIP Publishing LLC, 2018.
- [9] S. Shahane, N.R. Aluru, S.P. Vanka, Uncertainty quantification in three dimensional natural convection using polynomial chaos expansion and deep neural networks, *Int. J. Heat Mass Transf.* 139 (2019) 613–631.
- [10] H. Wu, X. Liu, W. An, S. Chen, H. Lyu, A deep learning approach for efficiently and accurately evaluating the flow field of supercritical airfoils, *Comput. Fluids* 198 (2020) 104393.
- [11] J. Berger, T. Busser, S. Reddy, G.S. Dulikravich, Evaluation of the reliability of a heat and mass transfer model in hygroscopic material, *Int. J. Heat Mass Transf.* 142 (2019) 118258.
- [12] J. Carrera, A. Alcolea, A. Medina, J. Hidalgo, L.J. Slooten, Inverse problem in hydrogeology, *Hydrogeol. J.* 13 (1) (2005) 206–222.
- [13] C. Shen, A transdisciplinary review of deep learning research and its relevance for water resources scientists, *Water Resour. Res.* 54 (11) (2018) 8558–8593.
- [14] H. Wei, S. Zhao, Q. Rong, H. Bao, Predicting the effective thermal conductivities of composite materials and porous media by machine learning methods, *Int. J. Heat Mass Transf.* 127 (2018) 908–916.
- [15] Y. Zhu, N. Zabaras, P.S. Koutsourelakis, P. Perdikaris, Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data, *J. Comput. Phys.* 394 (2019) 56–81.
- [16] J. Marçais, J.R. de Dreuzy, Prospective interest of deep learning for hydrological inference, *Groundwater* 55 (5) (2017) 688–692.
- [17] I. Goodfellow, Y. Bengio, A. Courville, *Deep Learning*, MIT Press, Cambridge, MA, USA, 2016.
- [18] Y. Zhu, N. Zabaras, Bayesian deep convolutional encoder–decoder networks for surrogate modeling and uncertainty quantification, *J. Comput. Phys.* 366 (2018) 415–447.
- [19] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *J. Comput. Phys.* 378 (2019) 686–707.
- [20] Q. He, D. Barajas-Solano, G. Tartakovsky, A.M. Tartakovsky, Physics-informed neural networks for multiphysics data assimilation with application to subsurface transport, *Adv. Water Resour.* 141 (2020) 103610.
- [21] S. Mo, N. Zabaras, X. Shi, J. Wu, Deep autoregressive neural networks for high-dimensional inverse problems in groundwater contaminant source identification, *Water Resour. Res.* 55 (5) (2019) 3856–3881.
- [22] S. Mo, Y. Zhu, N. Zabaras, X. Shi, J. Wu, Deep convolutional encoder-decoder networks for uncertainty quantification of dynamic multiphase flow in heterogeneous media, *Water Resour. Res.* 55 (1) (2019) 703–728.
- [23] P.J. Kreyenberg, H.H. Bauser, K. Roth, Velocity field estimation on density-driven solute transport with a convolutional neural network, *Water Resour. Res.* 55 (8) (2019) 7275–7293.
- [24] A.M. Tartakovsky, C.O. Marrero, P. Perdikaris, G.D. Tartakovsky, D. Barajas-Solano, Physics-informed deep neural networks for learning parameters and constitutive relationships in subsurface flow problems, *Water Resour. Res.* 56 (5) (2020) e2019WR026731.
- [25] S. Kamrava, P. Tahmasebi, M. Sahimi, Linking morphology of porous media to their macroscopic permeability by deep learning, *Transp. Porous Media* 131 (2) (2020) 427–448.
- [26] H. Wu, W.Z. Fang, Q. Kang, W.Q. Tao, R. Qiao, Predicting effective diffusivity of porous media from images by deep learning, *Sci. Rep.* 9 (1) (2019) 1–12.
- [27] Fraces, C. G., Papaioannou, A., & Tchelepi, H. (2020). Physics Informed Deep Learning for Transport in Porous Media. Buckley Leverett Problem. preprint arXiv:<http://arxiv.org/abs/2001.05172> (visited on 05/01/2021).
- [28] S. Liu, Z. Zhong, A. Takbiri-Borujeni, M. Kazemi, Q. Fu, Y. Yang, A case study on homogeneous and heterogeneous reservoir porous media reconstruction by using generative adversarial networks, *Energy Procedia* 158 (2019) 6164–6169.
- [29] S. Mo, N. Zabaras, X. Shi, J. Wu, Integration of adversarial autoencoders with residual dense convolutional networks for estimation of non-Gaussian hydraulic conductivities, *Water Resour. Res.* 56 (2) (2020) e2019WR026082.
- [30] R.A. Crooijmans, C.J.L. Willem, H.M. Nick, D.F. Bruhn, The influence of facies heterogeneity on the doublet performance in low-enthalpy geothermal sedimentary reservoirs, *Geothermics* 64 (2016) 209–219.
- [31] Z. Kong, C. Zhang, H. Lv, F. Xiong, Z. Fu, Multimodal feature extraction and fusion deep neural networks for short-term load forecasting, *IEEE Access* 8 (2020) 185373–185383.
- [32] S. Selvin, R. Vinayakumar, E.A. Gopalakrishnan, V.K. Menon, K.P. Soman, Stock price prediction using LSTM, RNN and CNN-sliding window model, in: Proceedings of the International Conference on Advances in Computing, Communications and Informatics (ICACCI), IEEE, 2017, pp. 1643–1647.
- [33] Z. Jiang, P. Tahmasebi, Z. Mao, Deep residual U-net convolution neural networks with autoregressive strategy for fluid flow predictions in large-scale geosystems, *Adv. Water Resour.* 150 (2021) 103878.
- [34] C. Hui, S. Kang, Entropy-based pruning method for convolutional neural networks, *J. Supercomput.* 75 (6) (2019) 2950–2963.
- [35] J. Gu, T. Liu, X. Wang, G. Wang, J. Cai, T. Chen, Recent advances in convolutional neural networks, *Pattern Recognit.* 77 (2018) 354–377.
- [36] X. Ji, Q. Yan, D. Huang, B. Wu, X. Xu, A. Zhang, M. Wu, Filtered selective search and evenly distributed convolutional neural networks for casting defects recognition, *J. Mater. Process. Technol.* 292 (2021) 117064.
- [37] O. Ronneberger, P. Fischer, T. Brox, U-net: convolutional networks for biomedical image segmentation, in: Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, Cham, 2015, pp. 234–241.
- [38] P. Tahmasebi, S. Kamrava, T. Bai, M. Sahimi, Machine learning in geo-and environmental sciences: from small to large scale, *Adv. Water Resour.* (2020) 103619.
- [39] D. Kumar, T. Roshni, A. Singh, M.K. Jha, P. Samui, Predicting groundwater depth fluctuations using deep learning, extreme learning machine and Gaussian process: a comparative study, *Earth Sci. Inform.* 13 (4) (2020) 1237–1250.
- [40] A. Rabbani, M. Babaei, R. Shams, Y. Da Wang, T. Chung, DeepPore: a deep learning workflow for rapid and comprehensive characterization of porous materials, *Adv. Water Resour.* 146 (2020) 103787.
- [41] M.M. Rajabi, H. Ketabchi, Uncertainty-based simulation-optimization using Gaussian process emulation: application to coastal groundwater management, *J. Hydrol. (Amst)* 555 (2017) 518–534.
- [42] C. Xiao, O. Leeuwenburgh, H.X. Lin, A. Heemink, Conditioning of deep-learning surrogate models to image data with application to reservoir characterization, *Knowl. Based Syst.* 220 (2021) 106956.

- [43] M. Tang, Y. Liu, L.J. Durlofsky, Deep-learning-based surrogate flow modeling and geological parameterization for data assimilation in 3D subsurface flow, *Comput. Methods Appl. Mech. Eng.* 376 (2021) 113636.
- [44] S. Kamrava, P. Tahmasebi, M. Sahimi, Physics-and image-based prediction of fluid flow and transport in complex porous membranes and materials by deep learning, *J. Membr. Sci.* 622 (2021) 119050.
- [45] F. Zhang, Q. Teng, H. Chen, X. He, X. Dong, Slice-to-voxel stochastic reconstructions on porous media with hybrid deep generative model, *Comput. Mater. Sci.* 186 (2021) 110018.