ORIGINAL PAPER

Estimating permeability of 3D micro-CT images by physics-informed CNNs based on DNS

Stephan Gärttner[1] · Faruk O. Alpak[2] · Andreas Meier[1] · Nadja Ray[1] · Florian Frank[1,3]

Abstract

In recent years, convolutional neural networks (CNNs) have experienced an increasing interest in their ability to perform a fast approximation of effective hydrodynamic parameters in porous media research and applications. This paper presents a novel methodology for permeability prediction from micro-CT scans of geological rock samples. The training data set for CNNs dedicated to permeability prediction consists of permeability labels that are typically generated by classical lattice Boltzmann methods (LBM) that simulate the flow through the pore space of the segmented image data. We instead perform direct numerical simulation (DNS) by solving the stationary Stokes equation in an efficient and distributed-parallel manner. As such, we circumvent the convergence issues of LBM that frequently are observed on complex pore geometries, and therefore, improve the generality and accuracy of our training data set. Using the DNS-computed permeabilities, a physics-informed CNN (PhyCNN) is trained by additionally providing a tailored characteristic quantity of the pore space. More precisely, by exploiting the connection to flow problems on a graph representation of the pore space, additional information about confined structures is provided to the network in terms of the maximum flow value, which is the key innovative component of our workflow. The robustness of this approach is reflected by very high prediction accuracy, which is observed for a variety of sandstone samples from archetypal rock formations.

Keywords Digital rock · Neural networks · Deep learning · Permeability · Porous media

Mathematics Subject Classification (2010) 05C21 · 68T07 · 76D07 · 76M10 · 76S05

1 Introduction

Artificial neural networks can accelerate or replace classical methods for estimating various hydrological and petrophysical properties of artificial and natural rock [50]. In [6], deep neural networks were successfully used to approximate tomography operators to reconstruct wave velocity models from seismic data. Moreover, convolutional neural networks (CNNs) were used in [30] to determine effective electrical parameters in a homogenized model for electric conductivity.

Likewise, CNNs have proven their ability to provide fast predictions of scalar permeability values directly from images of the pore space of geological specimens [23, 35, 53]. Finally, CNNs were successfully used to replace standard solution schemes to predict effective permeabilities in 2D multiscale flow simulations [18]. This paper aims at contributing to the deployment of machine learning techniques in effective permeability prediction by presenting a finite-element-based forward simulation approach as well as introducing a novelly considered characteristic quantity for physics-informed neural network (PhyCNN) models. We use the term PhyCNN throughout this paper to underline the incorporation of an additional physically motivated input parameter to the network. This approach is to be carefully distinguished from implementing physical laws into the loss-function as performed in [11] to estimate flow velocity fields.

Specimens of natural rock are typically obtained by microcomputed tomography (μCT) scanning [10, 52] or

✉ Nadja Ray
  ray@math.fau.de

✉ Florian Frank
  frank@math.fau.de

Extended author information available on the last page of the article.

indirectly from 2D colored images [5]. As an important characteristic quantity of porous media, permeability measures the ability of a fluid to travel through a considered pore space. However, in general, the precision and generality of the estimates performed by neural networks heavily depend on the quality of the underlying training data set, i. e., the accuracy of the forward simulations in our case, as shown in [56]. In terms of permeability prediction, the labeling process of geological specimens is connected to the computation of 3D stationary flow fields of a single-phase fluid within the pore space. For permeability prediction under regular geometries, a broader set of methods is available as described and compared in [49].

In the literature, various methods are available to compute the stationary flow on complex geometries as constituted by the pore space of porous media, a thorough comparison of which is found in [54]. Most commonly, lattice Boltzmann methods (LBM) are used to solve the flow problem on a discrete modeling basis, cf. [27]. In this approach, a transient interacting many-particle system is driven to equilibrium state, heavily exploiting the inherent parallelism of the underlying mathematical structure. For a detailed description of LBM fundamentals and numerics, we refer to [25]. Yet, since a global equilibrium has to be reached, complex geometries containing thin channels may cause LBM to converge slowly or even diverge [23, 37], especially in the case of very constricted and therefore typically under-resolved pore morphologies. This behavior is frequently observed with common pore-scale flow LBM implementations that rely on the conventional single-relaxation-time Bhatnagar–Gross–Krook (BGK) scheme [8]. However, the novel multiple-relaxation-time (MRT) scheme addresses this problem to a great extent (e. g., [3]) at the cost of a moderate additional computational overhead. Having stated that, residual discretization errors stemming from the explicit nature of time integration in LBM schemes still remain in the numerical solution. Especially for under-resolved pore morphologies, these errors may be amplified causing divergence of the numerical flow solution. We further note that classical bounce-back rules used for implementing boundary conditions tend to develop oscillations that might locally dominate fluid behavior in scenarios with thin channels, cf. [55]. Severe limitations arising from simplistic implementations of LBM restrict the training data sets to ones derived from sphere-packs instead of real CT data in many publications, see for instance [37]. On the other hand, industrial-grade state-of-the-art LBM solvers are typically proprietary without public access to the code base for academic research purposes (e. g., [3, 47]). Moreover, a comprehensive permeability computation benchmarking study [39] demonstrated that not all of the simulation methods deliver a good compromise of accuracy against computational performance indicating that there exists a clear need for accurate and computationally efficient alternative methods for permeability computation.

In practice, LBM simulations are often aborted after a maximal number of iterations in case one or more convergence criteria (typically linked to the relative changes in the velocity and/or computed permeability) cannot be met [23]. As such, training sets for neural networks can be artificially filtered by the numerical characteristics of the forward simulation, potentially resulting in biased data. To improve the generality and quality of our PhyCNN training sets, we base our machine learning data set on the stationary Stokes equation by performing direct numerical simulations (DNS) on the pore geometry [42]. More precisely, our forward simulation is based on a distributed-parallel Stokes solver utilizing the finite element library MFEM [4]. As studied in [49] for simple cylindrical obstacles, such DNS approaches (in this case FEM) deliver permeability values comparable to the ones from LBM simulation. However, our implementation successfully alleviates the drawback of an impractically large number of iterations to obtain the desired accuracy on complex geometries. As such, our approach allows overcoming prior restrictions in setting up representative training sets including also confined and complex structures. Moreover, no artificial data augmentation schemes such as pore space dilation are needed in our approach to increase the number of training data samples or enhance the porosity range covered.

Likewise, novel strategies are employed to our neural networks. More precisely, we exploit the concept of PhyCNNs, where the CNN is provided with additional (physics-related) input quantities to improve the reliability and the accuracy of its predictions [46, 53]. As shown in [44], carefully chosen specific quantities derived from the pore space such as connectivity indices can deliver reasonable approximation quality for permeability estimation. As illustrated in [6], also training a network solely on previously extracted features from the raw data may lead to satisfactory prediction quality.

The outstanding performance of our methodology is achieved by considering the *maximum flow* value, a graph-network derived quantity being highly correlated to the target permeability value and simple to compute. As such, we solve maximum flow problems on a graph representation of the pore space based on [12]. Thus, we approximate the Stokes flow through the pore space by an abstract flow through a graph. By using the scalar quantity of maximum flow as a second input to our neural network, we additionally provide our CNN with information that reflects possible thin, channel-like structures. As discussed in [48],
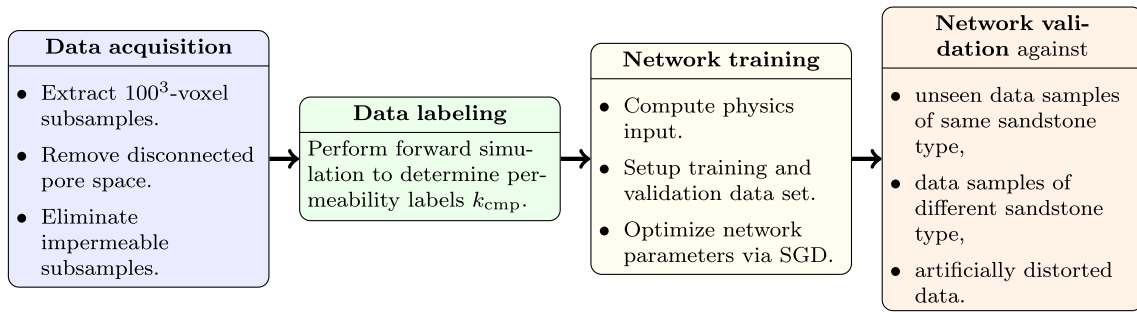
Fig. 1 Overall workflow in flow chart representation

graph representations have been shown highly capable of characterizing the pore-space connectivity in fractured rock and allow for a convenient way to deduce topological quantities of interest. We demonstrate that involving the maximum flow value in our newly designed PhyCNN in combination with the DNS-based forward simulation approach, our methodology delivers superior prediction accuracy and robustness compared to what is found in the literature.

The paper is organized as follows: In Section 2, we describe the sampling and preprocessing procedure of sandstone specimens including the forward simulation. Subsequently, Section 3 is dedicated to the network architecture used in our study. Finally, we validate the training performance of our PhyCNN on different types of sandstone in Section 4.

2 Methodology and data preparation

In this section, we describe the workflow and methodology by which we acquire the data set necessary to train and validate a CNN using a supervised learning approach. The preprocessing includes the selection and preparation of a set of pore-space geometries in form of voxel sets and the labeling with their computed permeability value k_{cmp}. Our complete workflow is presented in Fig. 1. We note that all steps except for the data labeling procedure (green box) are implemented in Matlab 2021a [31].

2.1 Sampling and preprocessing

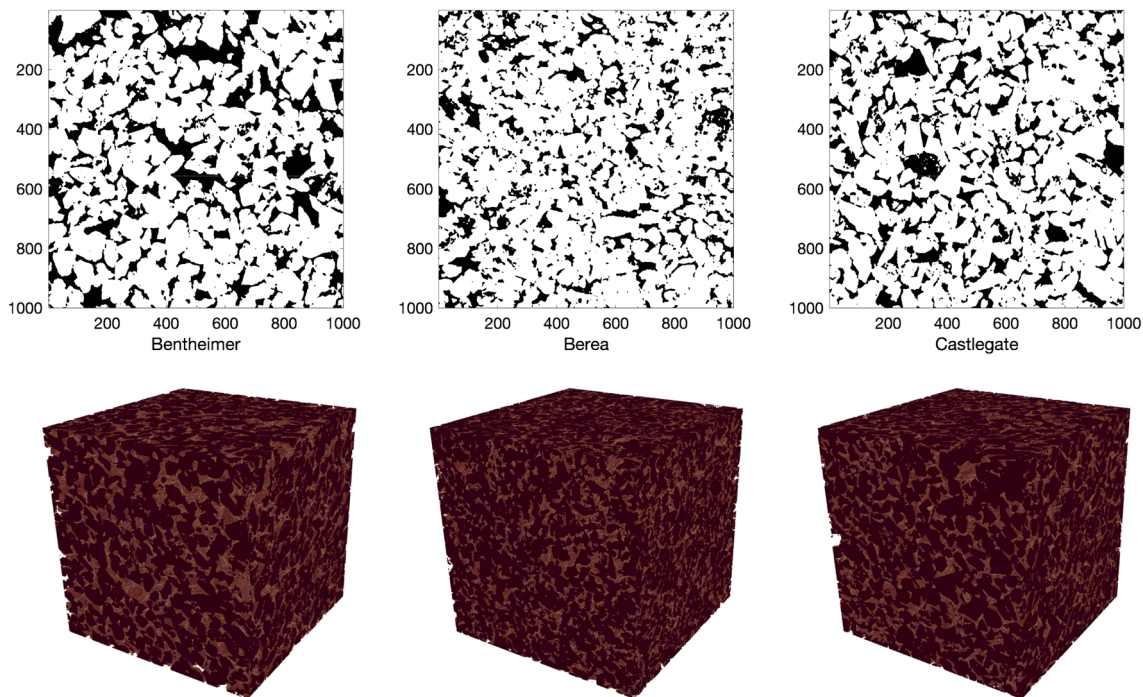The training procedure of our PhyCNN is based on a segmented X-ray μCT scan of a Bentheimer sandstone sample,

Fig. 2 1000^2 2D slices and 1000^3 3D pore-space μCT image of all samples considered in this study, illustrating characteristic pore features for Bentheimer, Berea, and Castlegate sandstone

see Fig. 2, with experimentally measured porosity $\phi_{\mathrm{exp}} = 22.64\,\%$ and permeability $k_{\mathrm{exp}} = 386\,\mathrm{mD}$, provided by [32, 33]. Further characteristic quantities for this sandstone as well as two other sandstone types used below for validation purposes are listed in Table 1. Bentheimer sandstone is known to exhibit a broad range in pore volume distribution and high pore connectivity in comparison to other types of sandstone [20]. As such, this sample is expected to contain a representative collection of geometrical and topological properties of the pore space in natural rocks. The data set used in this paper is derived from a $1000\times1000\times1000$ binary voxel image, in which each voxel either belongs to the pore space ("fluid voxels") or the rock matrix ("solid voxels"). The voxel edge length is $2.25\,\mu\mathrm{m}$ yielding an overall cube side length of $2.25\,\mathrm{mm}$.

In the first step, we extract subsamples of $100\times100\times100$ voxels from the Bentheimer sample. Henceforth, we use the term 'subsample' to refer to segmented μCT-scan pieces of this specific size. For subsample extraction, we make use of the sliding frame technique. This approach is commonly used to further exploit a given data set beyond the partition into disjoint subsets [43]. More precisely, we sweep a $100\times100\times100$ voxel frame along the coordinate axes of the original 1000-voxel cube and displace it by steps of 50 voxels (half a subsample size) resulting in a total of $19^3 = 6859$ subsamples. Although data are sampled redundantly, the set of obtained subsamples can be regarded independently when training neural networks [23]. Furthermore, by rotating the original $100\times100\times100$ voxel subsamples by 90° around the y and z axis, the number of extractable data is increased further by a factor of three. As such, artificial data augmentation techniques like erosion and dilation of the pore space from the μCT image are not required here for obtaining a sufficient and representative amount of training data. Even though we can produce $3\cdot19^3 = 20\,577$ subsamples by the method above, we select only the first 10 000 ones, since this number is sufficient to train our PhyCNN. In particular, this number of available data samples exceeds that of similar studies using natural

rock sample for training, cf. [23, 37]. However, using artificially generated pore-space geometries, even larger data sets including 90 000 data point are available in the literature, cf. [35]. More precisely, these data are obtained from stochastic computer models, evading the necessity of expensive imaging and segmentation of real rock.

Second, fluid voxels belonging to disconnected pore space with respect to the x direction possibly occurring within the subsamples are turned into solid voxels. As disconnected pores do not contributed to Stokes flow being driven from the inhomogeneous boundary conditions (1c) placed on opposing sides of the subsample, this procedure maintains permeability properties while facilitating their calculation. To this end, a simple graph walking algorithm is exploited to identify connected subdomains of the pore space. Starting from a random fluid node, neighboring nodes within the fluid domain are successively added until the scheme converges. A thorough description of this algorithm is found in [24].

By projecting the voxels of the connected pore space onto the x axis, we conclude whether each subsample has a nontrivial permeability. Subsamples with zero permeability are excluded from the later workflow (the maximum flow value is exactly zero if and only if the permeability is zero—therefore no training on such data samples is necessary, cf. Section 3.2.2). By encoding the simplified pore geometries in 1-bit raw format, memory consumption is 125 kB per subsample. Accordingly, our whole library of data samples allocates only 1.25 GB of disk space. Therefore, it is manageable on standard personal computers and file read/write access is reasonably cheap.

We note that a subsample of size 225 μm is too small to be a representative elementary volume (REV) for most sandstone types, cf. [20, 32]. As such, computed effective properties of one subsample cannot be expected to represent the whole segmented porous medium out of which it was extracted. On the other hand, the obtained training data set for our PhyCNN is therefore expected to be highly diverse, i. e. to contain highly permeable as well as narrow and confined pore geometries. Consequently, this setup is well suited to underline the robustness and prediction quality of our proposed methodology. Moreover, hierarchical neural networks have proven to be a powerful tool to leverage the performance of CNNs predicting permeability to samples of REV-size (in the order of 500^3 voxels), cf. [38]. Therefore, we believe that our proposed methodology also benefits existing REV-scale models.

Table 1 Characteristics quantities of sandstone samples used in this paper: experimentally determined porosity, ϕ_{exp} ($\pm0.5\%$), porosity computed from the μCT scan, ϕ_{cmp}, experimentally determined permeability, k_{exp} ($\pm10\%$), mean capillary diameter, MCD, and interior surface area, A_{cmp}, computed via the Matlab function `isosurface`

Type	ϕ_{exp}	ϕ_{cmp}	k_{exp} [mD]	MCD [μm]	A_{cmp} [mm]
Bentheimer	22.64%	26.72%	386	30.0	355
Berea	18.96%	21.67%	121	22.3	284
Castlegate	26.54%	24.67%	269	24.7	335

First four quantities are provided in [32]

2.2 Forward simulation

To apply a supervised learning approach as outlined in Section 3, each subsample within the data set derived by the methods of Section 2.1 needs to be labeled with

a computed permeability k_{cmp} that we use as the reference value.

To this end, in Section 2.2.1, we perform flow simulations on the pore space. More precisely, for each of the 10 000 subsamples, a stationary flow field along the x direction is computed by solving the Stokes equations on the union of fluid voxels Ω for the fluid velocity $\boldsymbol{u}$ and pressure p. The discretization uses arbitrary order (stable) Taylor–Hood or reduced-order stabilized Taylor–Hood mixed finite elements. In this paper, we consider voxel meshes only, yet the usage of unstructured grids by using obvious respective discrete spaces is straight forward.

In Section 2.2.2, the (absolute, scalar) permeability is computed by averaging the pressure gradient and velocity field across the subsample in x direction, cf. Section 2.1. Apparently, as the data set contains y and z-rotated versions of each subsample, this relates to the determination of the permeability with respect to all three main axes. By accurate bookkeeping, the diagonal permeability tensor of each initial subsample is retrievable.

Finally, in Section 2.2.3, we justify our choice of discretization parameters, i. e., mesh refinement level and finite element spaces used to produce the permeability values k_{cmp} to train and validate our PhyCNN.

We note that—despite the finite resolution of the µCT images—the voxel scans are henceforth considered to represent the actual (ground-truth) pore geometry. Hence, errors arising from imaging are neglected as they pose an independent part of the total methodology that is beyond the scope of this paper.

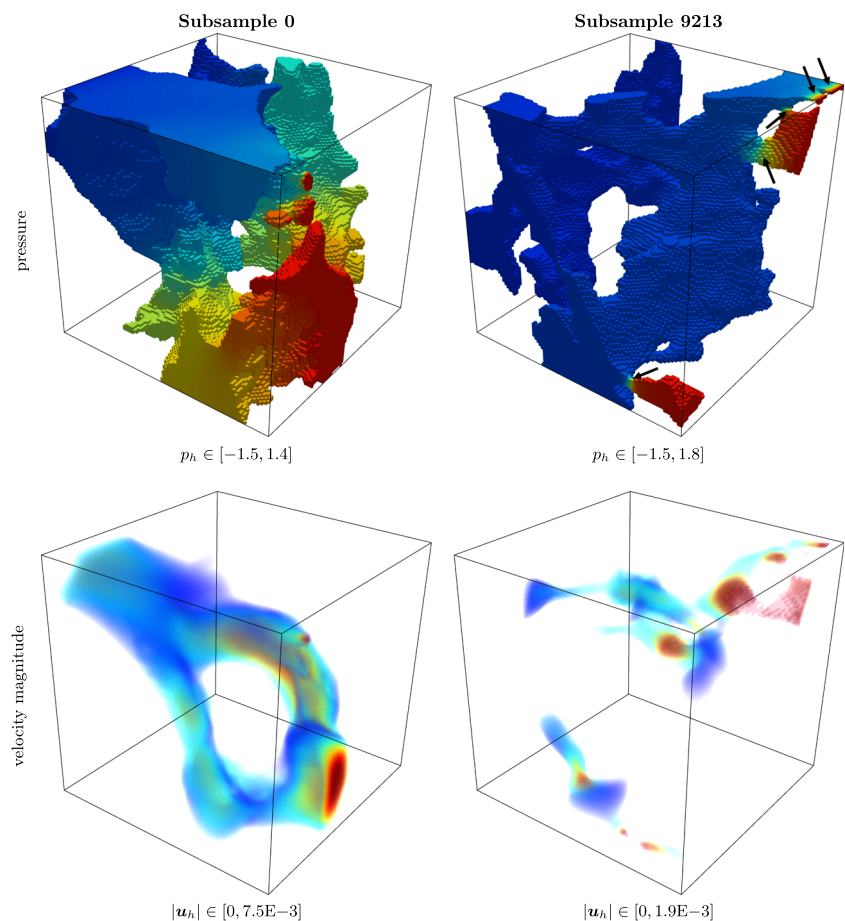### 2.2.1 Computation of the flow field

We consider the stationary Stokes equation for a Newtonian fluid in the nondimensionalized form,

$$-\frac{1}{\mathrm{Re}}\Delta\boldsymbol{u} + \nabla p = \boldsymbol{0} \quad \text{in } \Omega, \tag{1a}$$

$$\nabla\cdot\boldsymbol{u} = 0 \quad \text{in } \Omega, \tag{1b}$$

where $\Omega \subset (0,1)^3$ is a domain that consists of the union of fluid voxels of a considered $100{\times}100{\times}100$ voxel subsample (i. e. the subsample is inscribed into the unit cube). In Eq. 1, $\boldsymbol{u} = \boldsymbol{u}(x, y, z)$ denotes the (dimensionless) fluid velocity, $p = p(x, y, z)$ the (dimensionless) pressure, and $\mathrm{Re} := \rho\, U_{\mathrm{c}} L_{\mathrm{c}}/\mu$ the Reynolds number of the system with characteristic length L_{c} [m], characteristic velocity U_{c} [m s^{-1}], fluid density ρ [kg m^{-3}], and fluid viscosity μ [Pa s]. Note that the permeability k_{cmp} [m^2] is

Fig. 3 Examples of geometries used for the training process, with pressure fields *(top)* and velocity magnitudes *(bottom)*. Subsample 0 *(left)* exhibits moderate pressure gradients due to the wide and highly conductive channels in the pore space. Contrarily, in Subsample 9213 *(right)* Stokes pressure is dominated by thin structures leading to an ill-conditioned problem, i. e., small changes in the diameter of narrow pores have drastic impact on the permeability. Thin pore throats are indicated by black arrows in the pressure plot

Subsample 0 — $p_h \in [-1.5, 1.4]$

Subsample 9213 — $p_h \in [-1.5, 1.8]$

$|\boldsymbol{u}_h| \in [0, 7.5\mathrm{E}{-}3]$

$|\boldsymbol{u}_h| \in [0, 1.9\mathrm{E}{-}3]$

invariant with respect to Re (see below). The data set is constructed in such a way that there are connected fluid voxels (i. e. sharing a common face) reaching from the $x=0$ plane to the $x=1$ plane, cf. Figure 3, since impermeable subsamples were excluded from the workflow in the 'data acquisition' step, cf. Figure 1 and Section 2.1. The flow field is driven by a pressure gradient in x direction induced by the boundary condition

$$\left(\frac{1}{\mathrm{Re}}\nabla\boldsymbol{u} - p\,\mathbf{I}\right)\boldsymbol{n} = \boldsymbol{e}_x \quad \text{on } \Gamma_{\mathrm{N}}, \tag{1c}$$

where $\Gamma_{\mathrm{N}} := \left\{(x, y, z) \in \partial\Omega \mid x \in \{0, 1\}\right\}$ and $\boldsymbol{e}_x$ denoting the unit vector in x direction. On the remaining boundary $\Gamma_{\mathrm{D}} := \partial\Omega \setminus \Gamma_{\mathrm{N}}$, no-slip boundary conditions are prescribed,

$$\boldsymbol{u} = \boldsymbol{0} \quad \text{on } \Gamma_{\mathrm{D}}. \tag{1d}$$

The weak formulation of Eq. 1 is discretized by generalized Taylor–Hood pairs of spaces, $\mathbb{Q}_{\ell+1}^3/\mathbb{Q}_\ell$, where $\mathbb{Q}_\ell$ denotes the local space of polynomials of degree at most ℓ in each variable x, y, z, cf. [13, 14]. For $\ell = 0$, the pressure space $\mathbb{Q}_0 = \mathbb{P}_0$ is discontinuous and consists of elementwise constants. The respective "reduced Taylor–Hood pair" $\mathbb{Q}_1^3/\mathbb{P}_0$ requires stabilization (see below) due to a lack of discrete inf-sup-stability [13]. Let $\boldsymbol{\phi}_i = \boldsymbol{\phi}_i(x, y, z) : \Omega \to \mathbb{R}^3$, $i = 1, \ldots, n$ and $\psi_i = \psi_i(x, y, z) : \Omega \to \mathbb{R}$, $i = 1, \ldots, m$ denote the basis functions for the global discrete spaces for velocity and pressure, respectively. The discrete velocity $\boldsymbol{u}_h = \boldsymbol{u}_h(x, y, z)$ and discrete pressure $p_h = p_h(x, y, z)$ then have the representation

$$\boldsymbol{u}_h = \sum_{i=1}^{n}[\boldsymbol{x}^{\boldsymbol{u}}]_i\,\boldsymbol{\phi}_i, \qquad p_h = \sum_{i=1}^{m}[\boldsymbol{x}^p]_i\,\psi_i \tag{2}$$

with degree-of-freedom vectors $\boldsymbol{x}^{\boldsymbol{u}} \in \mathbb{R}^n$, $\boldsymbol{x}^p \in \mathbb{R}^m$, which are unique solutions of the linear system

$$\begin{bmatrix} \mathbf{A} & \mathbf{B}^{\mathrm{T}} \\ \mathbf{B} & -\mathbf{C} \end{bmatrix}\begin{bmatrix} \boldsymbol{x}^{\boldsymbol{u}} \\ \boldsymbol{x}^p \end{bmatrix} = \begin{bmatrix} \boldsymbol{b}^{\boldsymbol{u}} \\ \boldsymbol{0} \end{bmatrix} \quad \Longleftrightarrow: \quad \mathcal{A}\boldsymbol{x} = \boldsymbol{b} \tag{3}$$

with right-hand side $\boldsymbol{b}^{\boldsymbol{u}} \in \mathbb{R}^n$, $[\boldsymbol{b}^{\boldsymbol{u}}]_i := \int_{\Gamma_{\mathrm{N}}} \boldsymbol{e}_x \cdot \boldsymbol{\phi}_i$. The sparse blocks in $\mathcal{A}$ are the vector-Laplacian matrix $\mathbf{A} \in \mathbb{R}^{n,n}$ and the divergence matrix $\mathbf{B} \in \mathbb{R}^{m,n}$,

$$[\mathbf{A}]_{i,j} := \int_\Omega \nabla\boldsymbol{\phi}_i : \nabla\boldsymbol{\phi}_j, \qquad [\mathbf{B}]_{k,j} := -\int_\Omega \psi_k \nabla\cdot\boldsymbol{\phi}_j,$$

and $\mathbf{C}$ is a stabilization matrix that is required only for lowest order $\ell = 0$ to guarantee the full rank of $\mathcal{A}$. We choose $\mathbf{C}$ as in [13], (3.84), in which case, $\mathbf{C}$ can be interpreted as a pressure-Laplacian discretized by cell-centered finite volumes [16]. For $\ell > 0$, $\mathbf{C}$ is set to zero. In either case, $\mathcal{A}$ is symmetric and indefinite with n positive and m negative eigenvalues.

In order to solve the saddle point system (3) efficiently, a preconditioned MINRES method is applied, as it is the best choice of Krylov subspace methods for symmetric indefinite systems [51]. The chosen precondition operator for $\mathcal{A}$ in Eq. 3 is the symmetric and positively definite block-diagonal matrix

$$\mathcal{P} := \mathrm{diag}\,(\mathbf{A}, \mathbf{W})$$

with $\mathbf{W}$ being the pressure-mass matrix $[\mathbf{W}_{k,l}] := \int_\Omega \psi_k\,\psi_l$. The action of the inverse $\mathcal{P}^{-1}$ in each Krylov iteration is approximated block-wise by one V-cycle of an algebraic multigrid method (Boomer AMG from the Hypre library [15]). Since $\mathbf{W}$ is spectrally equivalent to the (negative) Schur complement $\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^{\mathrm{T}}$ of $\mathcal{A}$ (for $\ell > 0$), and due to the utilization of AMG, the number of Krylov iterations required to reach a given relative tolerance is bounded independently of the mesh size [7, 34, 36] (however, it highly depends on the geometry of the domain). MINRES with preconditioning as described above belongs to the state-of-the art Stokes solvers in the high-performance computing context [19]. In Section 2.2.3, we discuss appropriate choices regarding mesh size and discretization order.

Figure 3 illustrates Stokes velocity and pressure fields for two exemplary subsamples exhibiting qualitatively highly different pore-space geometries such as wide pore throats and narrow channels. We will emphasize the impact of highly and merely permeable rock samples on the behavior of DNS-based permeability computations and PhyCNN predictions throughout the paper.

2.2.2 Permeability estimation

We deduce the permeability value k_{cmp} of interest from the previously calculated Stokes velocity $\boldsymbol{u}$ and pressure p (we suppress the discretization index h in this section). In the Stokes Eq. 1, the inflow and outflow boundaries are defined as

$$\Gamma_{\mathrm{in}} := \{(x, y, z) \in \partial\Omega \mid x = 0\},$$
$$\Gamma_{\mathrm{out}} := \left\{(x, y, z) \in \partial\Omega \mid x = 1\right\}$$

and thus disjoint subsets of Γ_{N}. From the solution $(\boldsymbol{u}, p)$ of Eq. 1, we compute the approximated permeability $k_{\mathrm{cmp}}\,[\mathrm{m}^2]$ by the classical Darcy law, which reads in nondimensionalized form,

$$Q = -\mathrm{Da}\,\mathrm{Re}\,(P_{\mathrm{out}} - P_{\mathrm{in}}), \tag{4a}$$

with (dimensionless) volume flow rate Q, and (dimensionless) area-averaged inflow and outflow pressures, P_{in}, P_{out}, given by [28]

$$Q := \int_{\Gamma_{\mathrm{out}}} \boldsymbol{u}\cdot\boldsymbol{n}, \quad P_{\mathrm{in}} := \frac{1}{|\Gamma_{\mathrm{in}}|}\int_{\Gamma_{\mathrm{in}}} p, \quad P_{\mathrm{out}} := \frac{1}{|\Gamma_{\mathrm{out}}|}\int_{\Gamma_{\mathrm{out}}} p.$$

From Eq. 4a, k_{cmp} is derived from the Darcy number

$$\mathrm{Da} := \frac{k_{\mathrm{cmp}}}{L_{\mathrm{c}}^2}, \qquad (4b)$$

where L_{c} is the characteristic length, in our case, the edge length of a 100^3-voxel subsample, i. e., $L_{\mathrm{c}} = 225\,\mu\mathrm{m}$. We choose Re equal to one, since Re does not influence the permeability value k_{cmp} (a rescaling of $\boldsymbol{u}$ by Re^{-1} implies a rescaling of Q by Re^{-1} and therefore, Re cancels out in Eq. 4a).

Formula (4a) determines an approximation of the *scalar* permeability in x direction by area averaging. If this approach was applied to all three principal directions, it yielded a diagonal permeability tensor. In [22], a volume-averaged approach is proposed that is capable of determining the full permeability tensor. Application to x direction only, yields a column vector, whose entries are in general non-trivial. Since we want to train our PhyCNN with one scalar permeability value only, we decided to utilize the area-averaging approach in this study.

2.2.3 Choice of discretization parameters

In this section, we investigate the influence of mesh refinement and choice of polynomial order on the solution quality by comparing the computed permeability k_{cmp} obtained from Eq. 4a to the analytical value k_{ana}. In order to have analytical results available, we restrict our considerations to viscous flow through rectangular channels $\Omega_{a,b}$. This is supposed to constitute a sufficient benchmark, since the approximation quality of the computed permeability k_{cmp} is dominated by the discretization error in narrow pore throats due to high local gradients.

For the Stokes Eq. 1, consider a rectangular channel

$$\Omega_{a,b} := (0,1) \times \left(\frac{1}{2} - \frac{a}{2}, \frac{1}{2} + \frac{a}{2}\right) \times \left(\frac{1}{2} - \frac{b}{2}, \frac{1}{2} + \frac{b}{2}\right) \subset \mathbb{R}^3$$

of width $a \in (0,1)$ and height $b \in (0,1)$. An analytical expression for its permeability k_{ana} is [9]

$$k_{\mathrm{ana}} := \frac{K_\infty \cdot \min(a,b)^3 \cdot \max(a,b)}{12}, \qquad \text{where}$$

$$K_j := 1 - \sum_{n=1}^{j} \frac{1}{(2n-1)^5} \cdot \frac{192}{\pi^5}$$

$$\cdot \frac{\min(a,b)}{\max(a,b)} \tanh\left((2n-1)\frac{\pi}{2}\frac{\max(a,b)}{\min(a,b)}\right),$$

denoting the limit of K_j for $j \to \infty$ by K_∞. By application of the triangular inequality, $0 < \min(a,b) \leq \max(a,b)$, and $|\tanh(x)| < 1 \; \forall x \in \mathbb{R}$, we obtain the following approximation error bound:

$$|K_\infty - K_j| \leq \sum_{n=j+1}^{\infty} \frac{1}{(2n-1)^5} \cdot \frac{192}{\pi^5}.$$

Piecewise application of Jensen's inequality to the convex function $(2x-1)^{-5}$ finally yields the estimate:

$$|K_\infty - K_j| \leq \frac{192}{\pi^5} \int_{j+0.5}^{\infty} \frac{1}{(2x-1)^5} \, \mathrm{d}x$$

$$= \frac{192}{128} \cdot \frac{1}{\pi^5 j^4} \approx 0.0049 \, j^{-4}.$$

As such, we obtain at least six significant digits using K_{10} for the calculation of the analytical reference permeability k_{ana}.

Table 2 lists the relative error for the computed permeability k_{cmp} obtained from Eq. 4a on $\Omega_{0.06,0.03}$ using different polynomial orders ℓ and global mesh refinement levels. As expected, both refinement and higher-order discrete spaces consistently reduced the error with respect to the analytical solution. The best cost-to-approximation-quality ratio is achieved by $\mathbb{Q}_2^3/\mathbb{Q}_1$ elements ($\ell = 1$) on the original grid. In particular, this choice poses a significant improvement in accuracy over the use of lowest-order spaces $\mathbb{Q}_1^3/\mathbb{Q}_0$ ($\ell = 0$) with two global mesh refinement levels.

Table 2 Computed permeabilities k_{cmp} of a channel with 6×3 voxel rectangular cross section for different polynomial orders ℓ of the pressure space (cf. Section 2.2.1) and mesh refinement levels

Order ℓ	Ref. level	k_{cmp}	$\dfrac{\lvert k_{\mathrm{cmp}} - k_{\mathrm{ana}}\rvert}{k_{\mathrm{ana}}}$	m (DOF $\boldsymbol{u}_h$)	n (DOF p_h)
0	0	8.44E − 8	8.82E − 2	8 484	1 800
0	1	9.06E − 8	2.22E − 2	54 873	14 400
0	2	9.22E − 8	4.62E − 3	390 975	115 200
1	0	9.25E − 8	1.20E − 3	54 873	2 828
1	1	9.26E − 8	4.06E − 4	390 975	18 291

As in Eq. 2, m and n are the numbers of DOF for velocity $\boldsymbol{u}_h$ and pressure p_h, respectively. The relative tolerance of MINRES is set to 1.0E − 6, which has shown to be sufficient for three significant digits in the permeability value using a small scouting test set

The advantage of higher order discretizations presented above for narrow rectangular channels seems to generalize to our actual geological samples, which include confined structures. For Subsample 0, cf. Figure 3, where the main flow channels are wide with respect to the voxel resolution, k_{cmp} is hardly affected by increasing the polynomial order p from zero to one (4% deviation). On the other hand, Subsample 9213 (Fig. 3) exhibiting narrow and branchy structures experienced a significant relative increase in computed permeability of 43%. Therefore, we approve the increased computational complexity of lowest order classical Taylor–Hood elements ($\ell = 1$) for permeability labeling in our forward simulations, cf. Section 2.2.1. We emphasize that our scheme does not require mesh refinement, which is typically required in LBM simulations of natural rock.

This completes the methodology description for data acquisition and yields a training set suitable to train our neural network for permeability prediction. We conclusively note that the solver converged for every subsample in the database particularly maintaining the generality of our training set.

2.3 Evaluation metrics

In our study, we use the following well-known metrics to characterize and compare different measures of a data set statistically.

First, we define the standard deviation σ by

$$\sigma = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N} \left(t_i - \bar{t}\right)^2}$$

for a number N of real-valued data t_i and arithmetic mean value $\bar{t}$, measuring the expected deviation from $\bar{t}$. In order to quantify the correlation between different measures of a data set, we further introduce the coefficient of determination R^2

$$R^2 = 1 - \frac{\sum_{i=1}^{N} (t_i - y_i)^2}{\sum_{i=1}^{N} (t_i - \bar{t})^2}$$

for targets t_i and corresponding predictions y_i, encoding the share of variance in the targets that is covered by the predictions.

Finally, we define the mean-squared error MSE by

$$\mathrm{MSE} = \sum_{i=1}^{N} \frac{(t_i - y_i)^2}{N}.$$

3 Machine learning model

In this section, we motivate and present our suggested neural network architecture for permeability prediction. Therefore, we provide a brief introduction to the class of convolutional neural networks in Section 3.1 as the basis for our specific network architecture. For further reading, we refer to [2]. In Section 3.2, this model is augmented by an additional physical input quantity, which is derived in Section 3.2.2 in detail. For simplicity, we refer to the 100^3-voxel geological subsamples derived in Section 2.1 as (data) samples in the context of supervised machine learning.

3.1 Convolutional neural networks

In the following, we discuss the building blocks required to set up an artificial neural network tailored to the task of permeability predictions from 3D binary images of pore-space geometries. In a broader sense, we start with the more general concept of feed-forward neural networks.

As indicated by the name, feed-forward neural networks (FFNs) process their input data by successively propagating the input information through a finite number of layers. Each of these layers $i \in \{0, \ldots, L\}$ consists of N_i neurons, where the indices 0 and L correspond to the network's input and output layer, respectively. Typically, the action of a layer $1 \leq i \leq L$ on input data $x_{i-1} \in \mathbb{R}^{N_{i-1}}$ is given as an affine-linear transformation, followed by application of a nonlinearity (activation function) σ_i:

$$x_i = \sigma_i \left(W_i x_{i-1} + b_i \right).$$

We refer to $W_i \in \mathbb{R}^{N_i, N_{i-1}}$ as the weight matrices and to $b_i \in \mathbb{R}^{N_i}$ as the biases. Their combined entries are the learnable parameters of the network.

Convolutional neural networks (CNNs) constitute a subclass of FFNs exhibiting a dedicated internal structure specialized to the interpretation of image data. Furthermore, CNNs naturally incorporate translational invariance with respect to the absolute spatial position of features within the input data. In the following, we give a short introduction to this kind of networks. For a broader overview of common network designs and layer types, we refer to the respective literature, e. g. [2]. A list of the different layer types used in our study is given in Table 3.

Typically, CNNs consist of several convolutional (`conv`) lower layers (filter layers) including pooling operations, followed by `dense` neuron layers. As indicated by the name, convolutional layers perform the discrete analog of a convolution, where the convolution kernel is a learnable parameter of the network. As such, they possess a highly structured weight matrix W and are therefore superior to fully connected layers in terms of computational complexity, especially for highly localized

Table 3 Layer structure of our PhyCNN

Block	Layers	Learnables
input1	image input $100\times100\times100$	−
conv1	conv(32,5) − BN − LeakyReLU(0.1) − MP(5,5)	4 096
conv2	conv(64,5) − BN − LeakyReLU(0.1) − MP(5,5)	256 192
conv3	conv(128,3) − BN − LeakyReLU(0.1) − MP(2,2)	221 568
dense1	dense(64) − LeakyReLU(0.1)	65 728
concat1	concat(dense1, input2(physics input))	−
dense2	dense(32) − LeakyReLU(0.1)	4 161
output	regression(1)	−

The physical scalar input2 is expanded to dimension 64×1 and concatenated with the output of 'dense1' using a depth concatenation layer concat1. Nomenclature:

conv(N, K):	convolutional layer with N channels and $K \times K$ kernel size
BN:	batch normalization layer
MP(N, P):	maxPooling Layer, size N stride P
dense(N):	dense layer with N neurons
regression(N):	regression layer with N neurons
LeakyReLU(α):	leaky rectified linear unit, slope α on negative inputs
concat(in1,in2):	concatenation layer

kernels (sparsity). This restriction is justified in our application scenario, since most (low-level) geometry features are based on highly localized correlations among neighboring voxels.

MaxPooling (MP) layers are used to reduce the resolution of the output image. Presenting a natural bottleneck, the most significant pieces of information are selected and passed to the subsequent neuron layers. Application of several different filters to the same input within each layer of the network extracts increasingly complex characteristic features of the pore geometry. Finally, this information is interpreted by the subsequent dense neural layers.

Convergence of the training procedure is improved by exploiting additional batch normalization (BN) layers, calibrating the mean and spread of the provided inputs. Throughout the network, we apply LeakyReLU nonlinearities in an elementwise manner. For a parameter $\alpha \in (0, 1)$, the LeakyReLU(α) function is defined by

$$\text{LeakyReLU}(\alpha; x) := \max(0, x) - \alpha \max(0, -x).$$

Due to the slope of this nonlinearity being bounded away from zero, we circumvent the *dying neuron problem* occurring in deep neural networks with ReLU nonlinearities [29], which arise formally by setting $\alpha = 0$. Furthermore, backpropagation during the learning phase is facilitated. Our implementation is conducted using the Deep Learning Toolbox in Matlab R2021a [31].

3.2 Physics-informed convolutional neural networks

In the following, we use the building blocks for general CNNs as introduced in Section 3.1 and derive a network architecture specifically tailored to our application in permeability prediction. Therefore, we first give an introduction to PhyCNNs and thereafter discuss the choice of meaningful additional input quantities to the network.

3.2.1 General architecture

As discussed above, standard CNNs are required to extract all important features and information solely from the input image data to predict the correct output. Nevertheless, it is possible to provide the network with additional input derived from the image data in a preprocessing step, leading to the field of PhyCNNs. An exemplary schematic representation of such a network structure is provided in Fig. 5. As the additional physical input is typically not of image type (and hence does not exhibit spatially correlated features), it omits the convolutional layers and is directly coupled to the upper dense layers. As such, PhyCNNs predict by considering both the extracted geometry features as well as the provided physical input quantities. To achieve the desired increase in prediction accuracy and robustness, the additional data must be well-chosen and complement the insights gained by the CNN from the pure image data. Depending on the quality and quantity of the extracted features as well as the specific application, these alone may even provide sufficient information to the network to accurately perform predictions, cf. [6], underlining the potential of physics-informed strategies.

This approach has been successfully applied to the prediction of permeabilities from pore-space geometries: In [37], the Euclidean distance map, maximum inscribed sphere radii, and *time of flight* maps are inserted as inputs
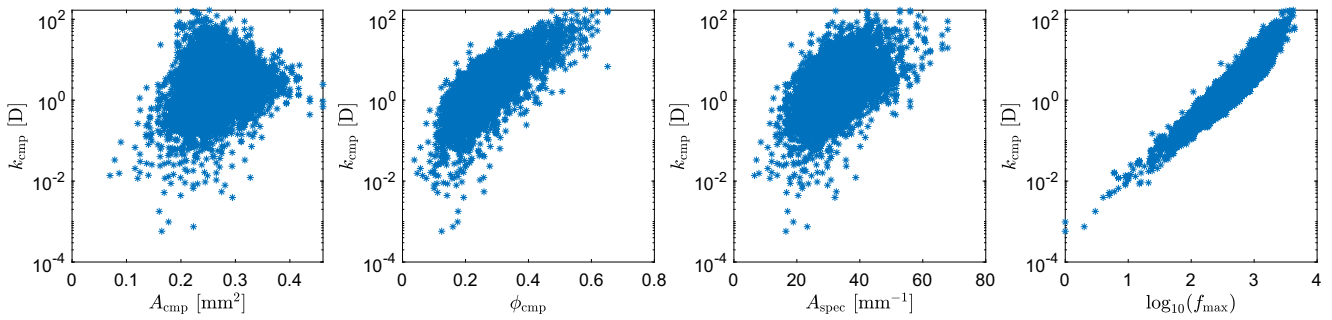
Fig. 4 Correlation of different physical measures of the underlying geometry with the computed permeability values k_{cmp} for Bentheimer data samples. From left to right: Computed surface area A_{cmp} ($R^2 =$ 0.055), computed porosity ϕ_{cmp} ($R^2 = 0.547$), specific surface area A_{spec} ($R^2 = 0.455$), maximum flow on graph representation f_{max} ($R^2 = 0.869$). The given R^2 values refer to a linear regression model

to determine the overall fluid field. In [53], porosity and surface area have been used to increase predictions performance for 2D image data. Similar observations have been made on 3D data where the additional consideration of porosity and tortuosity significantly reduced the number of outliers, especially when training on small data sets, cf. [45]. The specific choice of these parameters is related to the Kozeny–Carman formula estimating permeability from porosity and surface area / tortuosity. As indicated in [41], the proposed relation deteriorates in quality for increasingly complex geometries. However, [46] found porosity to be the most prominent input factor for their CNN among a set of various derived quantities such as coordinate number or mean pore size. Furthermore, characteristics obtained by pore network modeling have shown low general accuracy [46].

As indicated by our data set statistics in Fig. 4, porosity and surface area are only weakly correlated with the computed permeability k_{cmp} for our complex 3D setting. For a given surface area A_{cmp}, we observe data samples covering four orders of magnitude in their respective permeability values, about half the range is observed for fixed porosities ϕ_{cmp}. Hence, drastic increases in accuracy cannot be expected by regarding these quantities as additional physical input in our case. Instead, we construct another more specific physical quantity that can be computed efficiently from the input data using graph algorithms included in Matlab [31]. More precisely, we solve a surrogate maximum flow problem on a graph representation of the pore space. In Section 3.2.2, we give a thorough description of this chosen quantity that specifically encodes information about the sample that is difficult to access by standard CNNs.

Our final network architecture is presented schematically in Fig. 5 using the hyper-parameters of Table 3 and Fig. 5. The structure is based on the findings of [23],
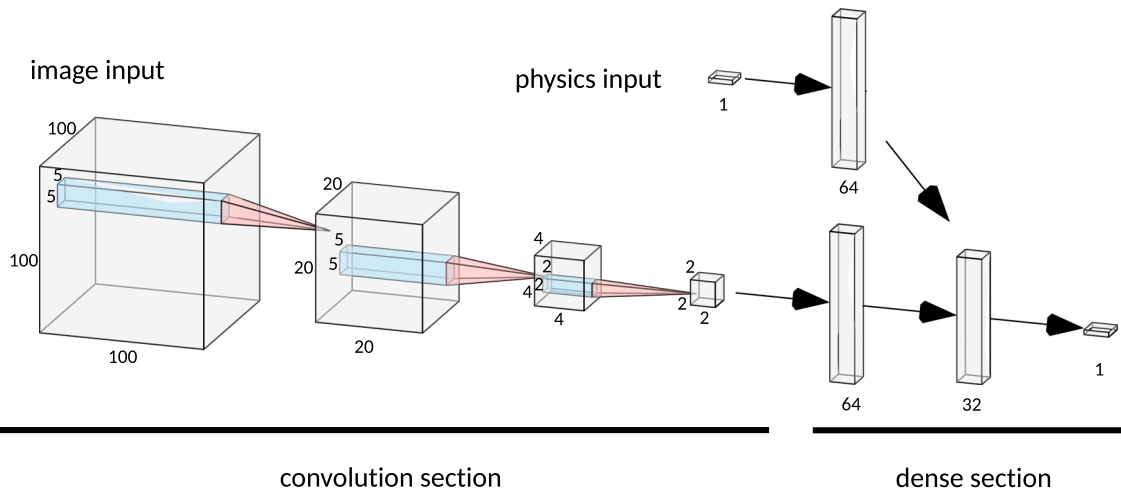

Fig. 5 Schematic representation of physics-informed convolutional neural networks (PhyCNNs). The image input is processed by convolutional layers, condensing extracted features in cubes of size 2^3. The scalar valued physical input circumvents the image convolution layers and is directly fed into the subsequent dense layers. As such, the interpretation of the given data performed in the network's upper `dense` layers relies on both information sources. Numbers at the layers denote their dimension (number of neurons). Note that the cubic convolutional kernels and their dimension are illustrated in 2D for clarity. Graphics produced using NN-SVG [26]

where the hyper-parameters of a purely convolutional neural network for permeability prediction have been optimized using a grid search algorithm. Since this optimization is specific to the data set considered, slight manual adjustments further increased accuracy on our data. More specifically, we decrease the bottleneck constituted by the final convolutional layer and improve convergence by adding batch normalization layers and relaxed cut-offs as nonlinearities (LeakyReLU). Furthermore, the maximum flow value $f_{\max}$ is treated as an additional scalar physical input quantity which is duplicated to a vector of length 64 and subsequently concatenated with the first dense layer of the network. By doing so, we balance both inputs to the second dense layer which results in a more uniform weight distribution. As a result, the network's upper dense layers are able to perform predictions from the extracted image data as well as the additional scalar input. For details on the parameters used within the network's training procedure, see Section 4.1. A pretrained version of our network is available as part of the toolbox RTSPHEM [17] on GitHub.

3.2.2 Maximum flow problems on graphs

In the field of optimization, maximum flow problems aim at identifying the maximum flow rate a network of pipes is able to sustain. Let the network be described by an undirected graph $G(N, E, \omega)$, where the nodes N correspond to distribution nodes, edges E to connections by pipes, and the weights ω encode the maximal capacity of each pipe. For any two nodes $n_1 \neq n_2 \in N$ (source and sink), we can determine the maximum flow between those nodes allowed by the network. For a thorough introduction to graph algorithms and maximum flow problems, we refer to [12]. In this sense, we aim at approximating relevant properties of the physical flow between two opposite sides of a sample by solving flow problems on a suitable graph network.

To do so, we consider the pore space of our segmented image data (stemming from geological specimens) to be

a graph by identifying each voxel as a node and each neighboring relation of voxels via a common face as an edge. This procedure is illustrated exemplarily in Fig. 6 for a 2D structure. For simplicity, all edge weights ω_i are assumed to be one. Note that this graph representation still comprises multiple standard characteristics of the pore space. For example, porosity translates to the number of nodes in the graph divided by the total number of voxels; the number of surface elements is approximately determined by $6 \operatorname{card}(N) - 2 \operatorname{card}(E)$. Moreover, we add a further node n_{in} connected to each voxel of the sample's inflow face, as well as n_{out} being connected to each voxel of the sample's outflow face, cf. Fig. 6. As such, we can approximate the permeability determination problem by a maximum flow problem through G between the nodes $n_{\mathrm{in}}, n_{\mathrm{out}}$.

For the samples considered in this paper, the computational effort for calculating $f_{\max}$ using Matlab's built-in routine maxflow with about one second per sample on a single CPU core, also cf. Section 4, is reasonably small compared to the solution of stationary Stokes equations as described in Section 4.3.

By the min-cut max-flow theorem, cf. [12], the maximum flow problem allows for another interesting interpretation. In the setup introduced within this section, $f_{\max}$ corresponds to the minimal number of edges that need to be deleted from G to obtain a disconnected graph such that inflow and outflow faces of the sample are contained in different connected components, cf. Fig. 6. Hence, we obtain information about the connectivity of the pore space with respect to the direction of interest. More precisely, $f_{\max}$ classifies structures containing thin channels regarding their restricting effects on fluid flow. This behavior is comprehensible using the geometries illustrated in Fig. 3: Exhibiting $f_{\max}$ of 1230, the permeability of Subsample 0 is governed by wide channels. On the other hand, exhibiting $f_{\max}$ of 55, narrow pore throats hamper the fluid flow of Subsample 9213.

We further note that for moderate tortuosity, $f_{\max}$ is approximately proportional to the volume of the channel
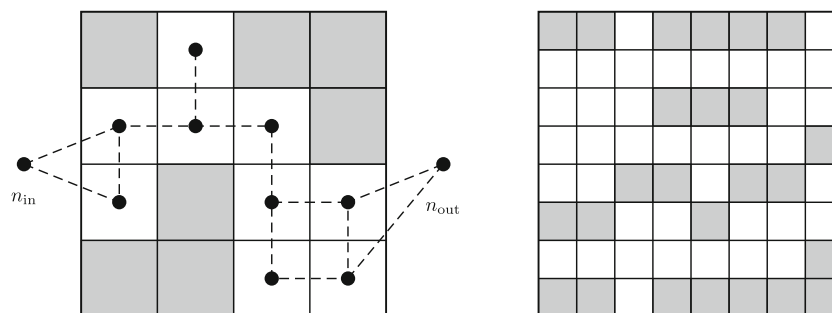

Fig. 6 Left: Derivation of the graph representation from a pore space illustrated by a 2D 4×4 pixel image. Nodes are depicted as circles, edges as lines. Gray voxels refer to the solid matrix. The maximum flow value $f_{\max}$ of the depicted graph is one. Right: More complex 8×8 pixel setup exhibiting $f_{\max}$ of three w.r.t. the horizontal axis and zero w.r.t. the vertical axis

needed to transport the maximum flow through the sample. As such, by exclusively regarding the subgraph of G actively used by the maximum flow, we cut off parts of G that do not effectively contribute to fluid flow. Hence, the determination of f_{max} can be understood as approximating the porosity with respect to the pore space participating in fluid transport. That way, we naturally improve the permeability estimation based on *classical* porosities.

Plotting f_{max} against the calculated permeabilities k_{cmp}, we conclude an almost linear relationship between both quantities in the logarithmic scale. Using mean-square linear regression, we obtain the approximate functional relation

$$k_{cmp} \sim 50625 \cdot (f_{max})^{1.407} \cdot 10^{-8.183} \quad [D], \quad (5)$$

cf. Fig. 4. On the displayed logarithmic scale, this regression yields an approximation quality of $R^2 = 0.8692$, cf. Section 2.3. Apparently, the derived quantity is strongly correlated to the target permeability k_{cmp}. Therefore, the permeability estimation using formula (5) is considered a suitable initial guess, which the CNN is able to improve by relating to features extracted from the pore geometry.

4 PhyCNN training and workflow performance

In this section, information concerning our PhyCNN's training procedure is provided. Furthermore, we evaluate the quality of our resulting predictions on three different types of sandstone as well as a challenging artificially deformed series of geometries. Finally, we compare the computational efficiency of our Stokes solver and the PhyCNN.
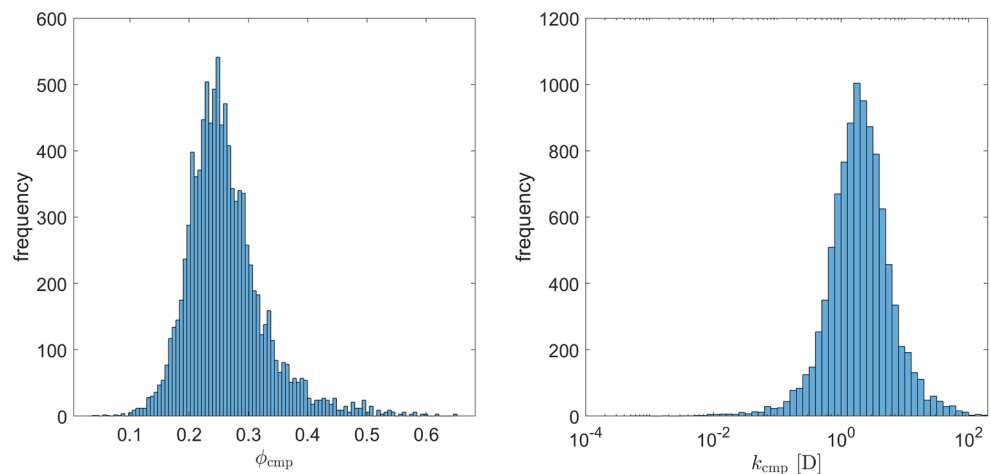
4.1 PhyCNN training

As the data points obtained by the forward computation are quite sparse for extremely high and low permeability values (Fig. 7), we disregarded all data samples ranging outside the interval of [50 mD, 50 D]. As a result, we improve the quality of predictions within the permeability range where the availability of data is favorable and reliable. Splitting the remaining data set according to a 90%/10% key, 8 876 samples are used for training the network, another 987 for validation. In order to accurately account for the labels covering three orders of magnitude, a logarithmic transformation was applied before training as done in [21]. Using a standard mean-squared-error (MSE) loss in the regression as given in Section 2.3, this results in measuring the error in relative deviation rather than absolute deviation.

The training was performed over 15 epochs using a stochastic gradient descent (SGD) optimizer with momentum 0.9. An almost constant validation loss over the last epochs indicated convergence of the network. Starting from an initial learning rate of 0.0020, the step size has been decreased by 60% every four epochs.

4.2 Prediction quality

The network's predictive performance is illustrated in Fig. 8 via regression plots. Our PhyCNN achieved coefficient of determination (R^2) values of 96.33% on training data and 93.22% on validation data, cf. Table 4. As such, the network accomplished very high accuracy with only weak tendency to overfitting. Furthermore, the plots in Fig. 8 show a low number of outliers, underlining the stability of our approach.

In order to further stress the robustness, we additionally validate our PhyCNN on samples originating from different types of sandstone that were not used for training. Using the data provided by [32, 33], 200 additional subsamples were extracted from each Berea and Castlegate

Fig. 7 Computed porosity ϕ_{cmp} and permeability k_{cmp} distribution among the total of 10 000 Bentheimer samples. Mean porosity is 26.20% with standard deviation $\sigma = 0.067$, mean permeability is 3930 mD with $\sigma = 8319$ mD

Fig. 8 PhyCNN accuracy on validation a) and training data b) originating from Bentheimer sandstone. The related R^2 values are 0.8820, 0.9289 in the natural scale and 0.9322, 0.9633 in the logarithmic scaling. Images c) and d) show our PhyCNN's accuracy on the Berea and Castlegate test sets, respectively
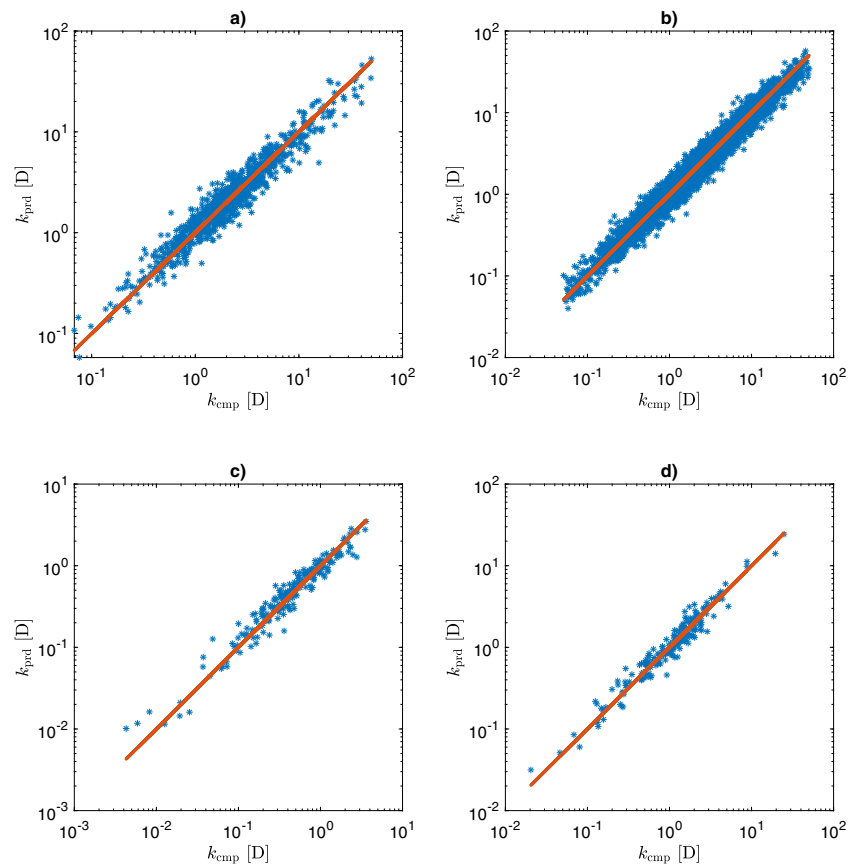
μCT scan, see Table 1, using the same methods as described in Section 2. Achieving R^2 values of 89.43% and 94.13% (logarithmic) on Berea as well as 94.55% and 94.51% (logarithmic) on Castlegate, the networks proves excellent generalization properties across different types of pore geometries, cf. Figs. 2, 8 and Table 1. As such, the network seems in fact to perform slightly better on Berea and Castlegate data samples than on the validation data set of Bentheimer. However, Table 4 marks a very high standard deviation for the permeability

Table 4 Overview prediction quality

	Measure	Bentheimer (train.)	Bentheimer (val.)	Berea	Castlegate
Data set	mean k_{cmp} [mD]	3417.7	3628.8	660.9	1661.0
	$\sigma(k_{cmp})$ [mD]	4948.7	5233.2	657.8	2442.2
PhyCNN(f_{max})	R^2	92.89%	88.20%	89.43%	94.55%
	R^2 log	96.33%	93.22%	94.13%	94.51%
PhyCNN(ϕ-σ)	R^2	85.65%	73.55%	72.81%	74.86%
	R^2 log	91.27%	79.57%	58.81%	75.12%
plain CNN	R^2	78.92%	73.11%	64.54%	62.66%
	R^2 log	84.48%	77.97%	58.36%	74.12%
plain f_{max}	R^2	57.78%	57.67%	42.82%	56.46%
	R^2 log	85.15%	85.03%	82.65%	81.68%

In the first block, mean permeability and standard deviation $\sigma(k_{cmp})$ are listed for the respective data sets. Subsequently, R^2 values in natural and logarithmic scale for the training and validation Bentheimer data as well as for Berea and Castlegate sandstone are presented as achieved by our PhyCNN, i.e., the maxflow-informed CNN (PhyCNN(f_{max})), the porosity/surface-area-informed CNN (PhyCNN(ϕ-σ)), and a plain CNN without additional inputs. Finally, the additional input quantity f_{max} is considered in a power-law (linear fitting applied to logarithmic variables) regression model (plain f_{max})

in Bentheimer compared to the other sandstone types. As such, the latter data sets comprise less heterogeneity facilitating the network's prediction. We further emphasize that both Berea and Castlegate validation sets have not been screened to match the trained range [50 mD, 50 D]. More precisely, eleven Berea data samples exhibit a permeability k_{cmp} below 50 mD as well as two Castlegate data samples.

Moreover, Table 4 lists the prediction quality indices of other related CNNs for comparison. To clearly and consistently separate the different approaches investigated, we refer to the PhyCNN described in Section 3.1 more precisely as PhyCNN(f_{max}) within this comparison. In case of the PhyCNN(ϕ-σ), the general structure depicted in Fig. 5 is maintained while replacing the maxflow input variable by the porosity ϕ and surface area σ, and retraining the network, cf. Fig. 4. As the data of Table 4 show, the PhyCNN(ϕ-σ) cannot reach the accuracy of our PhyCNN(f_{max}). Moreover, robustness appears significantly lower since accuracy on other sandstone types than the training data decreases rapidly. Especially for Berea sandstone, the results are poor, which might be a direct consequence of the significantly different pore-space characteristics in comparison to Bentheimer, cf. Table 1. Furthermore, we also compare to the analogous network without any additional input quantities (plain CNN), resulting in slightly worse predictions than the PhyCNN(ϕ-σ). Finally, we note that both of these CNNs provide less accurate predictions on validation samples than f_{max} alone (plain f_{max}) being fitted to the Bentheimer training data set using a power law approach and measured with respect to R^2-log. As such, we conclude that f_{max} already contains highly relevant information regarding the samples' permeability delivering more accurate predictions in the logarithmic measure than standard CNN approaches. However, our PhyCNN(f_{max}) improves the results of pure f_{max} by extracting further information directly from the pore-space, cf. Fig. 5.

Finally, we investigate the generalization limits of our PhyCNN by subjecting it to an artificially distorted data set covering a challengingly wide permeability and porosity range. To achieve that, we applied a level-set-based algorithm already used in [18] to erode or dilate the pore space of Subsample 0 displayed in Fig. 3. More precisely, the pore space is contracted with a uniform level-set velocity directed perpendicularly to the pore walls until all flow channels collapse, i.e. the data sample becomes impermeable. The same number of deformation steps is also used to expand the pore volume of Subsample 0. Each of those steps corresponds to a constriction/expansion of the pore space by a single additional voxel layer on average.

Subsequently, we compute the permeabilities k_{cmp} using the Stokes solver and k_{prd} using the PhyCNN of the series of pore spaces and compare the results in Fig. 9. The data show an almost perfect match for expanded pore spaces even for porosities up to 70%. Using linear interpolation, we estimate k_{cmp} to exceed the trained permeability range [50 mD, 50 D] beyond $\phi_{cmp} = 57.33\%$. Therefore, we conclude that our PhyCNN is capable of properly characterizing also highly permeable samples. On the other hand, predictions remain reasonably accurate for strongly confined geometric structures. Three samples in Fig. 9 show porosities below 9.08%, which approximately refers to the lower end of the trained permeability range for this example. These exhibit f_{max} of 4, 16, and 30, respectively, referring to an almost disconnected pore space. Since the flow channels narrow in these cases to only very few voxels in diameter, errors from the μCT scan as well the discretization of Stokes Eq. 1 may become non-negligible. As such, these samples may leave the current operating limits of digital rock physics, resulting in a significant systematic overestimation of permeability compared to lab experiments [40].
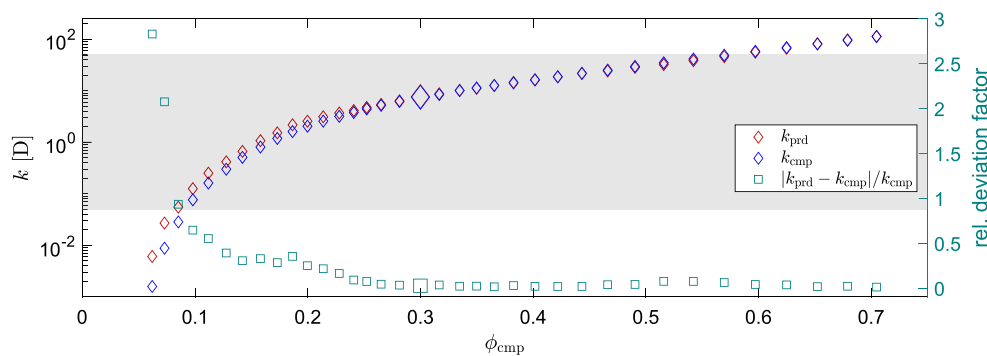

Fig. 9 Testing generalization ability on artificially distorted data samples. PhyCNN prediction performance on eroded and dilated pore spaces exhibiting a large range of porosities ϕ_{cmp}. For each manipulated geometry, Stokes simulation data k_{cmp} (red) and CNN prediction k_{prd} (blue) are compared. Furthermore, we provide relative prediction errors. Markers referring to the original, not manipulated data sample are increased in size. The permeability range spanned by our PhyCNN's training data set is highlighted in gray

4.3 Computational performance

In this section, we provide computational performance indicators for the forward simulations as performed using the method described in Section 2.2 as well as the ones obtained for estimations using the PhyCNN. Subsequently, we compare the forward simulation run times for the generation of 10 000 data samples including both permeability computation approaches on the same compute cluster to estimate the actual speed up. All subsequent specifications of computation times refer to the wall time.

Each of the 10 000 forward simulations on Bentheimer sandstone was performed with classical Taylor–Hood elements on voxels ($\ell=1$, cf. Section 2.2.3) in parallel on 50 compute nodes of the Emmy compute cluster at RRZE [1], each being equipped with two Intel®Xeon 2660v2 'Ivy Bridge' processors (10 physical cores per chip, i. e., $50 \cdot 2 \cdot 10 = 1000$ cores total, no hyper-threading) and 64 GB of RAM. The linear solver (MINRES) converged within 969 iterations on average with a relative tolerance of $1.0\mathrm{E} - 6$ in the norm $\| \cdot \|_{\mathcal{P}^{-1}}$. Furthermore, meshing, assembly, and solution were performed within 2.35, 0.44, and 33.95 seconds on average per solver call, respectively. As such, the labeling procedure of all 10 000 data samples was completed within roughly 100 hours. We conclude that the forward simulation effort of our Stokes approach is roughly comparable to the LBM-based simulations applied in related publications, cf. [23].

Operating on a graphics cluster featuring two Intel®Xeon E5-2620v3 (6 cores each), two Nvidia® Geforce Titan X GPUs, and 64 GB of RAM, the Phy-CNN's training process terminated within two hours. Using 10 CPU cores and a single graphics chip on this machine, the permeabilities of the 10 000 elements data set of Bentheimer sandstone is estimated within 1294 seconds by our PhyCNN. More precisely, 867 seconds on CPU were spent on the graph flow problems related to $f_{\max}$ as described in Section 3.2.2 while 367 seconds were required to perform the subsequent network inferencing on GPU. Estimating the time required to solve the related Stokes problem (1) extrapolated from a subset of ten data samples indicates a total time of approximately 1469 hours. Accordingly, we conclude an acceleration factor of 4087 by using our PhyCNN on the stated hardware configuration and data set.

From the data presented above, we infer that the time invested in the network's training procedure including the calculation of $f_{\max}$ for the complete database is compensated after about 15 data samples computed beyond the training and validation data sets. This strongly underlines the capability of our approach to pose a time-saving yet accurate alternative to flow simulation-based permeability estimators on large data sets.

5 Summary and conclusions

In this work, we demonstrated the feasibility of direct numerical simulation (DNS) for flow through porous media to generate a library of computed permeability labels from 3D images acquired using specimens of natural rocks. Our distributed-parallel stationary Stokes solver achieved computational efficiency comparable to classical lattice Boltzmann method (LBM) implementations while successfully addressing convergence challenges that typically appear on complex 3D geometries including poorly resolved subdomains.

As a result of computations with the proposed numerically robust forward simulation algorithm, an unbiased (by computational artifacts) data set was constructed to train a convolutional neural network for accelerated permeability predictions. An easily-computable graph-based characteristic quantity of the pore space, namely the maximum flow value, was introduced to the neural network as another novel development in our work. As a result, this methodology enabled our machine learning model to achieve an R^2 value of 93.22% on the validation data set. Moreover, similar prediction qualities were found for types of sandstone rock that are different from the training samples, as well as for artificially generated voxel sets. These observations underline the robustness of our artificial intelligence augmented permeability estimation approach.

In a one-on-one comparison before accounting for the training-investment-related computational costs, the neural-network-based permeability estimation approach delivered a speed-up in excess of 4000 fold. On the other hand, computational results indicate that the proposed approach exceeds the performance of a purely DNS-based workflow beyond approximately 15 data samples when the training-related computational costs are accounted for. This indicates that the proposed artificial intelligence augmented permeability estimation workflow is viable for real-life digital rock physics applications.

Finally, we note that the refined methodology presented in this paper may generalize to larger sample sizes approaching the REV-scale. Recent publications suggest hierarchical multiscale neural networks [38] to alleviate the memory requirements of the training and inference procedure. Merging both approaches holds the potential of leveraging our current results to larger scale geological samples.

Notation

Symbols

A_{cmp} Computed interior surface area of µCT scan [mm^2].

A_{spec} Specific (w.r.t. material volume) interior surface area [mm^{-1}].

α Parameter in `LeakyReLU` nonlinearity.

f_{max} Maximum flow value.

k_{ana} Analytical permeability (all permeabilities in darcies D, millidarcies mD).

k_{cmp} Computed permeability, cf. (4).

k_{exp} Experimental permeability.

k_{prd} Predicted permeability.

ℓ Polynomial degree along principle axes ($\mathbb{Q}_\ell$ is the local space of polynomials of degree at most ℓ in each variable).

Ω Pore space, domain $\Omega \subset \mathbb{R}^3$ of the Stokes Eq. 1.

p Hydrostatic pressure on the pore scale (dimensionless), solution $p : \Omega \rightarrow \mathbb{R}$ of the Stokes Eq. 1.

ϕ_{cmp} Computed porosity.

ϕ_{exp} Experimental porosity.

R^2 R-squared value, coefficient of determination.

Re   Reynolds number (definition is irrelevant here, since Re cancels out in the computation for k_{cmp}, cf. (4)).

σ Standard deviation.

$\boldsymbol{u}$ Solenoidal fluid velocity on the pore scale (dimensionless), solution $\boldsymbol{u} : \Omega \rightarrow \mathbb{R}^3$ of the Stokes Eq. 1.

n Node $n \in N$ in an undirected graph $G(N, E, \omega)$, cf. Section 3.2.2.

Abbreviations CNN, Convolutional neural network; DNS, Direct numerical simulation; DOF, Degrees of freedom; FFN, Feed forward network; LBM, Lattice Boltzmann method; μCT, Microcomputed tomography; MSE, Mean squared error; PhyCNN, Physics-informed convolutional neural network; ReLU, Rectified linear unit; REV, Representative elementary volume; SGD, Stochastic gradient descent.

Data Availability The CNN code used in this paper is available as part of the porous media numerical toolkit RTSPHEM [17] on GitHub.

Declarations

Conflict of Interests The authors have no relevant financial or non-financial interests to disclose.

References

1. Emmy Compute-Cluster: RRZE, Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany (2021)

2. Aggarwal, C.C.: Neural Networks and Deep Learning, vol. 25, p. 497. Springer International Publishing, Berlin (2018)

3. Alpak, F.O., Gray, F., Saxena, N., Dietderich, J., Hofmann, R., Berg, S.: A distributed parallel multiple-relaxation-time lattice Boltzmann method on general-purpose graphics processing units for the rapid and scalable computation of absolute permeability from high-resolution 3D micro-CT images. Comput. Geosci. **22**(3), 815–832 (2018). https://doi.org/10.1007/s10596-018-9727-7

4. Anderson, R., Andrej, J., Barker, A., Bramwell, J., Camier, J.-S., Dobrev, J.C.V., Dudouit, Y., Fisher, A., Kolev, T., Pazner, W., Stowell, M., Tomov, V., Akkerman, I., Dahm, J., Medina, D., Zampini, S.: MFEM: A modular finite element library. Comput. Math. Appl. **81**, 42–74 (2021). https://doi.org/10.1016/j.camwa.2020.06.009

5. Araya-Polo, M., Alpak, F.O., Hunter, S., Hofmann, R., Saxena, N.: Deep learning-driven permeability estimation from 2D images. Comput. Geosci. **24**(2), 571–580 (2019). https://doi.org/10.1007/s10596-019-09886-9

6. Araya-Polo, M., Jennings, J., Adler, A., Dahlke, T.: Deep-learning tomography. Lead. Edge **37**, 58–66 (2018). https://doi.org/10.1190/tle37010058.1

7. Benzi, M., Golub, G.H., Liesen, J.: Numerical solution of saddle point problems. Acta Numer. **14**, 1–137 (2005). https://doi.org/10.1017/s0962492904000212

8. Bhatnagar, P.L., Gross, E.P., Krook, M.: A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems. Phys. Rev. **94**(3), 511–525 (1954). https://doi.org/10.1103/physrev.94.511

9. Bruus, H.: Theoretical Microfluidics. Oxford Master Series in Physics. OUP Oxford (2008)

10. Bultreys, T., Boever, W.D., Cnudde, V.: Imaging and image-based fluid transport modeling at the pore scale in geological materials: A practical introduction to the current state-of-the-art. Earth-Sci. Revi **155**, 93–128 (2016). https://doi.org/10.1016/j.earscirev.2016.02.001

11. Cheng, C., Zhang, G.-T.: Deep learning method based on physics informed neural network with resnet block for solving fluid flow problems. Water **13**(4), 423 (2021). https://doi.org/10.3390/w13040423

12. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C. Introduction to Algorithms, 2nd edn. The MIT Press, Cambridge (2001)

13. Elman, H.C., Silvester, D.J., Wathen, A.J.: Finite Elements and Fast Iterative Solvers: With Applications in Incompressible Fluid Dynamics. Numerical Mathematics and Scientific Computation. Oxford University Press, New York (2014)

14. Ern, A., Guermond, J.L.: Theory and Practice of Finite Elements. Applied Mathematical Sciences. Springer, Berlin (2004)

15. Falgout, R.D., Jones, J.E., Yang, U.M.: The design and implementation of *hypre*, a library of parallel high performance preconditioners. In: Bruaset, A.M., Tveito, A. (eds.) Numerical Solution of Partial Differential Equations on Parallel Computers, pp. 267–294 (2006). https://doi.org/10.1007/3-540-31619-1_8

16. Frank, F., Liu, C., Alpak, F.O., Rivière, B.: A finite volume/discontinuous Galerkin method for the advective Cahn–Hilliard equation with degenerate mobility on porous domains stemming from micro-CT imaging. Computat. Geosci. **22**(2), 543–563 (2018). https://doi.org/10.1007/s10596-017-9709-1

17. Gärttner, S., Frank, F.: RTSPHEM – reactive transport solver in porous homogenized evolving media. Department Mathematik, Friedrich-Alexander-Universität Erlangen-Nürnberg. https://doi.org/10.5281/zenodo.5166669, https://github.com/cupperfreeze/RTSPHEM/. Accessed 5 Dec 2022 (2021)

18. Gärttner, S., Frolkovič, P., Knabner, P., Ray, N.: Efficiency of micro-macro models for reactive two-mineral systems. Multiscale Model. Simul. **20**(1), 433–461 (2022). https://doi.org/10.1137/20m1380648

19. Gmeiner, B., Huber, M., John, L., Rüde, U., Wohlmuth, B.: A quantitative performance study for Stokes solvers at the extreme scale. J. Comput. Sci. **17**, 509–521 (2016). https://doi.org/10.1016/j.jocs.2016.06.006

20. Gong, L., Nie, L., Xu, Y.: Geometrical and topological analysis of pore space in sandstones based on X-ray computed tomography. Energies **13**(15), 3774 (2020). https://doi.org/10.3390/en13153774

21. Graczyk, K.M., Matyka, M.: Predicting porosity, permeability, and tortuosity of porous media from images by deep learning. Sci. Rep. **10**(1). https://doi.org/10.1038/s41598-020-78415-x (2020)

22. Guibert, R., Nazarova, M., Horgue, P., Hamon, G., Creux, P., Debenest, G.: Computational permeability determination from pore-scale imaging: sample size, mesh and method sensitivities. Transp. Porous Med. **107**(3), 641–656 (2015). https://doi.org/10.1007/s11242-015-0458-0

23. Hong, J., Liu, J.: Rapid estimation of permeability from digital rock using 3D convolutional neural network. Comput. Geosci. **24**(4), 1523–1539 (2020). https://doi.org/10.1007/s10596-020-09941-w

24. Hopcroft, J., Tarjan, R.: Algorithm 447: efficient algorithms for graph manipulation. Commun. ACM **16**(6), 372–378 (1973). https://doi.org/10.1145/362248.362272

25. Krüger, T., Kusumaatmaja, H., Kuzmin, A., Shardt, O., Silva, G.: The Lattice Boltzmann Method, p. 720. Springer International Publishing, Berlin (2016)

26. LeNail, A.: NN-SVG: Publication-ready neural network architecture schematics. J. Open Source Softw. **4**(33), 747 (2019). https://doi.org/10.21105/joss.00747

27. Liao, Q., Jen, T.-C.: Computational Fluid Dynamics. IntechOpen, pp. 410. https://doi.org/10.5772/10585 (2011)

28. Liu, C., Frank, F., Alpak, F.O., Rivière, B.: An interior penalty discontinuous Galerkin approach for 3D incompressible Navier–Stokes equation for permeability estimation of porous media. J. Comput. Phys. **396**, 669–686 (2019). https://doi.org/10.1016/j.jcp.2019.06.052

29. Lu, L.: Dying ReLU and initialization: theory and numerical examples. Commun. Comput. Phys. **28**(5), 1671–1706 (2020). https://doi.org/10.4208/cicp.oa-2020-0165

30. Lu, X., Giovanis, D.G., Yvonnet, J., Papadopoulos, V., Detrez, F., Bai, J.: A data-driven computational homogenization method based on neural networks for the nonlinear anisotropic electrical response of graphene/polymer nanocomposites. Comput. Mech. **64**(2), 307–321 (2018). https://doi.org/10.1007/s00466-018-1643-0

31. MATLAB: Version 9.10.0.1602886 (R2021a). Natick, Massachusetts: The MathWorks Inc. (2021)

32. Neumann, R.F., Barsi-Andreeta, M., Lucas-Oliveira, E., Barbalho, H., Trevizan, W.A., Bonagamba, T.J., Steiner, M.: High accuracy capillary network representation in digital rock reveals permeability scaling functions. Sci. Rep. **11**(1). https://doi.org/10.1038/s41598-021-90090-0 (2021)

33. Neumann, R., Andreeta, M., Lucas-Oliveira, E.: 11 Sandstones: raw, filtered and segmented data. http://www.digitalrocksportal.org/projects/317, https://doi.org/10.17612/f4h1-w124 (2020)

34. Pearson, J.W., Pestana, J., Silvester, D.J.: Refined saddle-point preconditioners for discretized Stokes problems. Numer. Math. **138**(2), 331–363 (2017). https://doi.org/10.1007/s00211-017-0908-4

35. Prifling, B., Röding, M., Townsend, P., Neumann, M., Schmidt, V.: Large-scale statistical learning for mass transport prediction in porous materials using 90,000 artificially generated microstructures. Front. Mater. **8**. https://doi.org/10.3389/fmats.2021.786502 (2021)

36. Rozložník, M.: Saddle-Point Problems and their Iterative Solution. Nečas Center Series. Springer International Publishing, Berlin (2018)

37. Santos, J.E., Xu, D., Jo, H., Landry, C.J., Prodanović, M., Pyrcz, M.J.: PoreFlow-Net: A 3D convolutional neural network to predict fluid flow through porous media. Adv. Water Resour. **138**, 103539 (2020). https://doi.org/10.1016/j.advwatres.2020.103539

38. Santos, J.E., Yin, Y., Jo, H., Pan, W., Kang, Q., Viswanathan, H.S., Prodanovič, M., Pyrcz, M.J., Lubbers, N.: Computationally efficient multiscale neural networks applied to fluid flow in complex 3D porous media. Transp. Porous Med. **140**(1), 241–272 (2021). https://doi.org/10.1007/s11242-021-01617-y

39. Saxena, N., Hofmann, R., Alpak, F.O., Berg, S., Dietderich, J., Agarwal, U., Tandon, K., Hunter, S., Freeman, J., Wilson, O.B.: References and benchmarks for pore-scale flow simulated using micro-CT images of porous media and digital rocks. Adv. Water Resour. **109**, 211–235 (2017). https://doi.org/10.1016/j.advwatres.2017.09.007

40. Saxena, N., Hows, A., Hofmann, R., Alpak, F.O., Dietderich, J., Appel, M., Freeman, J., De Jong, H.: Rock properties from micro-CT images: Digital rock transforms for resolution, pore volume, and field of view. Adv. Water Resour. **134**, 103419 (2019). https://doi.org/10.1016/j.advwatres.2019.103419

41. Schulz, R., Ray, N., Zech, S., Rupp, A., Knabner, P.: Beyond Kozeny–Carman: Predicting the permeability in porous media. Transp. Porous Med. **130**(2), 487–512 (2019). https://doi.org/10.1007/s11242-019-01321-y

42. Siena, M., Hyman, J.D., Riva, M., Guadagnini, A., Winter, C.L., Smolarkiewicz, P.K., Gouze, P., Sadhukhan, S., Inzoli, F., Guédon, G., Colombo, E.: Direct numerical simulation of fully saturated flow in natural porous media at the pore scale: a comparison of three computational systems. Comput. Geosci. **19**(2), 423–437 (2015). https://doi.org/10.1007/s10596-015-9486-7

43. Sudakov, O., Burnaev, E., Koroteev, D.: Driving digital rock towards machine learning: Predicting permeability with gradient boosting and deep neural networks. Comput. Geosci. **127**, 91–98 (2019). https://doi.org/10.1016/j.cageo.2019.02.002

44. Sun, H., Al-Marzouqi, H., Vega, V.: EPCI: A new tool for predicting absolute permeability from computed tomography images. Geophysics **84**(3), F97–F102 (2019). https://doi.org/10.1190/geo2018-0653.1

45. Tang, P., Zhang, D., Li, H.: Predicting permeability from 3D rock images based on CNN with physical information. J. Hydrol. **606**, 127473 (2022). https://doi.org/10.1016/j.jhydrol.2022.127473

46. Tembely, M., AlSumaiti, A.M., Alameri, W.S.: Machine and deep learning for estimating the permeability of complex carbonate rock from X-ray micro-computed tomography. Energy Rep. **7**, 1460–1472 (2021). https://doi.org/10.1016/j.egyr.2021.02.065

47. Tölke, J.: Implementation of a lattice Boltzmann kernel using the compute unified device architecture developed by nVIDIA. Comput. Vis. Sci. **13**(1), 29–39 (2008). https://doi.org/10.1007/s00791-008-0120-2

48. Viswanathan, H., Hyman, J., Karra, S., O'Malley, D., Srinivasan, S., Hagberg, A., Srinivasan, G.: Advancing graph-based algorithms for predicting flow and transport in fractured rock. Water Resour. Res. **54**(9), 6085–6099 (2018). https://doi.org/10.1029/2017wr022368

49. Wagner, A., Eggenweiler, E., Weinhardt, F., Trivedi, Z., Krach, D., Lohrmann, C., Jain, K., Karadimitriou, N., Bringedal, C., Voland, P., Holm, C., Class, H., Steeb, H., Rybak, I.: Permeability estimation of regular porous structures: a benchmark for comparison of methods. Transp. Porous Med. **138**(1), 1–23 (2021). https://doi.org/10.1007/s11242-021-01586-2

50. Wang, Y.D., Blunt, M.J., Armstrong, R.T., Mostaghimi, P.: Deep learning in pore scale imaging and modeling. Earth-Sci. Rev. **215**, 103555 (2021). https://doi.org/10.1016/j.earscirev.2021.103555

51. Wathen, A.J.: Preconditioning. Acta Numer. **24**, 329–376 (2015). https://doi.org/10.1017/S0962492915000021

52. Wildenschild, D., Sheppard, A.P.: X-ray imaging and analysis techniques for quantifying pore-scale structure and processes in subsurface porous medium systems. Adv. Water Resour. **51**, 217–246 (2013). https://doi.org/10.1016/j.advwatres.2012.07.018. 35th Year Anniversary Issue

53. Wu, J., Yin, X., Xiao, H.: Seeing permeability from images: Fast prediction with convolutional neural networks. Sci. Bull. **63**(18), 1215–1222 (2018). https://doi.org/10.1016/j.scib.2018.08.006

54. Yang, X., Mehmani, Y., Perkins, W.A., Pasquali, A., Schönherr, M., Kim, K., Perego, M., Parks, M.L., Trask, N., Balhoff, M.T., Richmond, M.C., Geier, M., Krafczyk, M., Luo, L.-S., Tartakovsky, A.M., Scheibe, T.D.: Intercomparison of 3D pore-scale flow and solute transport simulation methods. Adv. Water Resour. **95**, 176–189 (2016). Pore scale modeling and experiments. https://doi.org/10.1016/j.advwatres.2015.09.015

55. Yang, Z.: Analysis of lattice Boltzmann boundary conditions. http://nbn-resolving.de/urn:nbn:de:bsz:352-opus-35079. PhD thesis. Konstanz: Universität Konstanz (2007)

56. Zhou, Y., Wu, Y.: Analyses on influence of training data set to neural network supervised learning performance. In: Jin, D., Lin, S. (eds.) Advances in Computer Science, Intelligent System and Environment, p. 19?25. Springer, Berlin (2011). https://doi.org/10.1007/978-3-642-23753-9_4

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

Affiliations

Stephan Gärttner[1] · **Faruk O. Alpak[2]** · **Andreas Meier[1]** · **Nadja Ray[1]** · **Florian Frank[1,3]**

Stephan Gärttner
gaerttner@math.fau.de

[1] Department Mathematik, Friedrich-Alexander-Universität Erlangen-Nürnberg, Cauerstraße 11, 91058 Erlangen, Germany

[2] Shell Technology Center, 3333 Highway 6 South, Houston, TX 77082, USA

[3] Math2Market GmbH, Richard-Wagner-Straße 1, 67655 Kaiserslautern, Germany