Deep Rock: fluid transport properties through disordered media with convolutional neural networks

Pawel Gniewek*
(Dated: July 15, 2019)

I. BACKGROUND

Machine Learning (ML) methods, and recently very popular Deep Learning (DL) in particular [1, 2], turned out to be successful in handling complex problems. After a period of a stagnation, the deep learning attracted an attention of researchers from the multiple fields, upon unanticipated success in image processing in 2012 [3]. Later on, DL proved to be very successful in competing professional human players in video games [4–6], or board games like chess, Go, or Shogi [7–9]. At the same time, the potential of deep neural networks has been recognized in the fields of image segmentation [10, 11], language processing [12, 13], and medicine [14, 15]. It is not a surprise that DL methods found many applications in physical sciences. For instance, both *supervised* and *unsupervised* ML methods are used in the problems such as predicting crystal structures [16, 17] and their stability [18], approximating density functional [19] and correlation energies [20], classifying and discovering phases and phase transitions in statistical models [21–27] as well classifying protein classes [28] or drugs activity [29], learning ground states and thermodynamics of many-body systems [30], approximating wave functions for quantum many-body problems in and out of equilibrium [31, 32], and estimating liquid crystals properties [33].

From this perspective, it seems natural to use ML to study fluid transport properties. Despite the obvious industrial and economic importance [34, 35], applications of DL to the turbulent flow [36–40] or liquid transport in heterogeneous materials [41–46] are only at their wake. ML/DL approaches have an advantage of being agnostic about (i) underlying order parameters that control the fluid flow and (ii) resulting constitutive relations quantifying the fluid transport [47–56].

In this work, we make the first steps towards a broader application of deep learning to geophysics, water resources research, and potentially petrology. However, before the advances machine learning methods can unleash their full potential, some fundamental aspects need to be understand. First, it needs to be shown that deep neural networks posses the capacity of extracting patterns from the finite amount of heterogeneous samples and can make quantitative predictions of the material's properties. Even though the first task may be feasible, the machine learning methods requires an immense amount of data in order to be able to extract the patters required to deliver quantitative predictions. This however is one of the hurdle in geophysical research, where the costs of obtaining the samples are high, and the data availability is often too scarce [57–59] to robustly train models. This is one of the reasons why in the industrial and academic applications, methods such as Lattice-Boltzmann methods [], Finite Volume method [], or Pores-network model [60–65] gained so much popularity. These methods can provide a quantitative results, however computational costs and the expertise required to perform these simulations, significantly limit the number of potential practitioners and beneficiaries. That is why the methods such as deep learning could bridge the gap between the need from the academic and industrial community and the scarcity of the data. Thus, in this work we aim to show that training DL models with a limit amount of data is feasible with a very good accuracy.

In this manuscript, we consider only a single phase flow since this is of a principal relevance in hydrology, and petrology and this value sets the maximum permeability (absolute permeability) for the multi-phase flow. First, we introduce data generation process. Next, we describe the architecture of the convolutional neural network. Then, we show the neural network training process and the prediction accuracy. Finally, we point out the future research directions and possible applications.

II. METHODS

A. Generation of 2D porous material

2D porous medium is often modeled using overlapping squares deposited on a regular lattice [67–70]. In this work we generate samples of a heterogeneous porous material by depositing aligned squares of the size k (in lattice units) on the 2D lattice with periodic boundary conditions, Fig 1(a)-(c). The percolation model presented here corresponds to the "percolation of voids" model and it was studied before by Koza et al. [66], and it interpolates between the site percolation of voids on a regular lattice ($k = 1$) and the continuum percolation of voids of aligned squares ($k \to \infty$). We identify a cluster that percolates the packing in a direction x_i as a cluster that spans two opposite sides of the systems, Fig.1(d). Periodic boundary conditions are assumed in the perpendicular direction, $x_j \perp x_i$, (so called the spanning cluster [71]) for lattice-Boltzmann calculations, Fig.1(d).

* gniewko.pablo@gmail.com

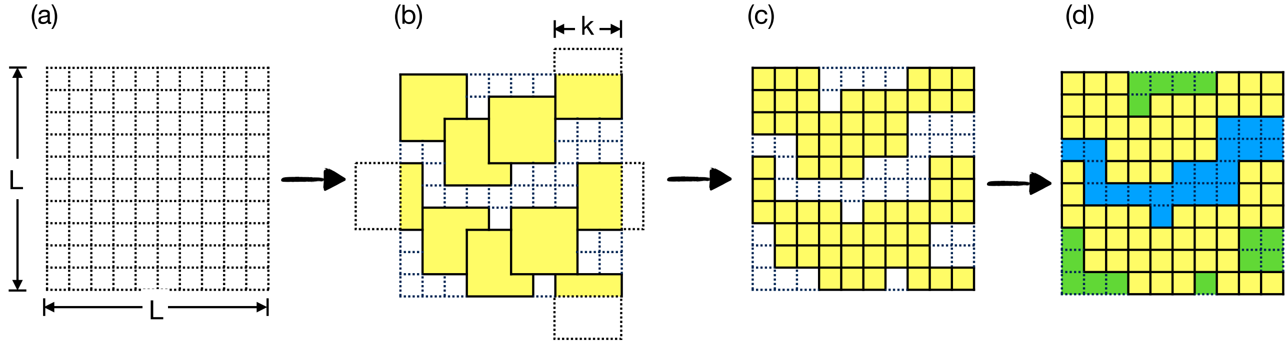

FIG. 1. A schematic of the procedure used to generate 2D porous material: (a) An empty 2D lattice with the dimension $L \times L$, and the lattice size $l = 1$. (b) Identical square, with a linear dimension k are deposited on the lattice. The squares are aligned with the lattice grid. (c) Sites occupied by the deposited squares are not permeable to liquid (d) A percolating cluster is identified that spans two opposite sides of the lattice. Scheme adapted from Koza et al. [66]

B. Permeability calculations with lattice-Boltzmann simulations

Permeability of the porous media is usually calculated using the Darcy's law:

$$q_i = -\frac{1}{\mu}\kappa_{ij}\left(\partial_j p - \rho b_j\right) \tag{1}$$

Packings generated in Section II A are structurally isotropic. To avoid distinguishing between any direction, we assume that the effects of the body forces can be neglected ($\partial_j p \gg \rho b_j$). Additioanlly, to avoid finite size effects [72], material's permeability is calculated as $\langle\kappa\rangle = \frac{1}{d}\mathrm{tr}\left(\kappa_{\mathrm{ij}}\right)$, where $d = 2$ in this paper. Thus, the permeability can be taught as a part of the proportionality constant in the relation describing the obstruction of the laminar flow through the material, and can be calculated knowing the stationary flow field through the sample.

Velocity field of the fluid flow through the 2D packings is solved with the lattice-Boltzmann (LB) method [73] using the D2Q9 lattice. This method was proven to be successful in studies of liquid flow in 2D porous materials [51, 67–69, 72–77]. Although D2Q9 lattice is commonly used in 2D LB simulations, mass transport may be better represented with D2Q5 lattice, especially close to the percolation threshold [78]. LB method provides us a solution to the Navier-Stokes equation for the flow in low Reynolds numbers limit. The LB method is using a velocity distribution function rather than velocity and pressure fields and is numerically more stable than the Finite Element Method at the irregular boundaries that are inevitable in porous materials [73]. To ensure better numerical stability for the complex geometry of the pores, we use multiple relaxation times (MRT) to solve linearized Boltzmann equation with LB method [79].

Permeability of the packing and the flow field are resolved by setting a pressure difference ΔP between two opposite sides of the simulation box, sufficiently small to keep the flow in the incompressible and laminar regimes (Stokes flow). Every simulation is performed for periodic boundary condition (PBC) in a direction perpendicular to the pressure gradient. In the direction of the pressure gradient, the system is open and the boundary conditions are set by pressure difference [58, 80]. No-slip boundary condition is applied to the solid material boundaries.

The flow fields obtained from LB simulations for each lattice site, $u(\mathbf{r})$, are further used to calculate the permeability (and potentially could be used to calculate other characteristics such as tortuoisty [75]). Permeability is calculated for a given flow filed as $\kappa = \eta\left\langle u(\mathbf{r})\right\rangle/\nabla P$. Permeability is given in lattice units (for conversion to physical units follow Latt [81]). All the LB simulations are performed with PALABOS [82].

C. Architecture of the neural network

The input images are processed as a 2D binary masks, and padded with periodic boundaries for before processing with the first convolutional layer in the Deep-Rock architecture. The architecture of the Deep-Rock model is a deep convolutional neural network with a single neuron output for regression for permeability prediction. The architecture consist of two parts (i) 5 2D-convolution layers with (3x3) kernels and (ii) two dense (fully-connected) layers; c.f. Figure 2. Each convolutional layer is proceeded with periodic boundary padding and followed by dropout [83] and Average Pooling [84], Figure 2. Average pooling was used as we found that Max-Pooling [85] works better for classification problems and not particularly well for regression type of tasks.

Finally, both convolutional and dense layers use ReLU activation function [86]. The only exception is the last neuron, where linear activation function is used, *c.f.*

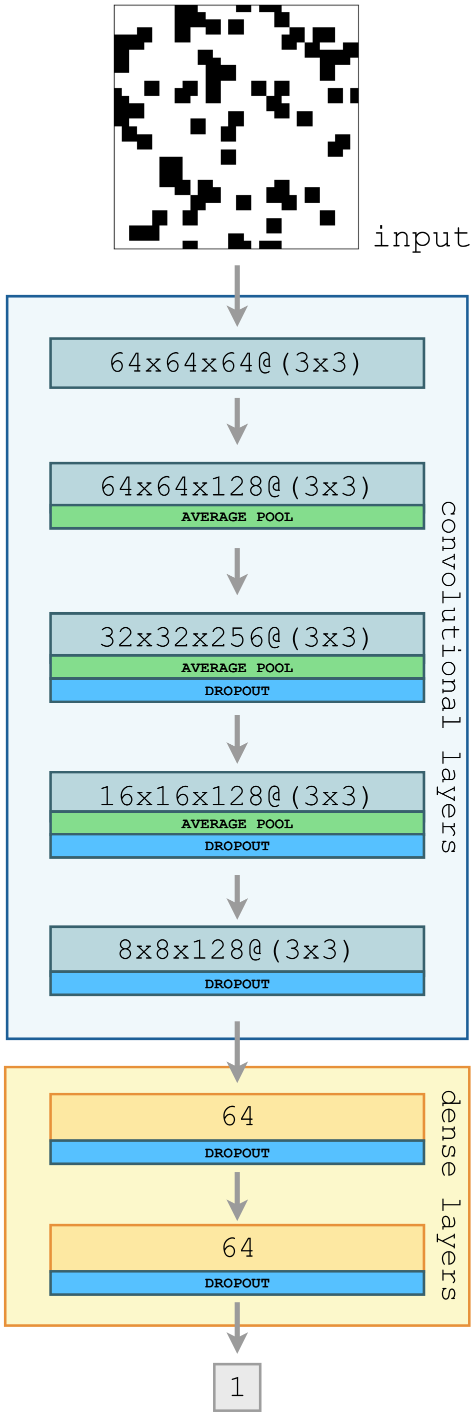

FIG. 2. The architecture of the Deep-Rock network. The architecture is designed to process 2D binary images with a set of convolutional (light blue box) and dense (light orange box) layers. Dropout layers are executed with the rates [0.25, 0.2, 0.15, 0.15, 0.1] (from top to bottom). More details can be found in Section II C.

bottom of Figure 2. The architecture is implemented with Keras library [87] and source codes are available on GitHub [88]. The network is trained on 4288 (void) percolating packings generated with the protocol described in Section II A and tested on 1073 independent packings. Mean-square-error (MSE) is used as a cost function (for the log-permeabilites). The analysis of the results is for the parameterization that scores the best on the training dataset, see Section **??**.

1. *Visualization of CNN filters*

Visualisation and interpretation of the convolutional neural networks is an ongoing research topic [89–93]. The filters used in this work are relatively small (3x3), so a simple visualisation of these kernels may not give us a good idea about the what features a given kernel is sensitive to. Thus, we visualize filters of the hidden layers by creating an input (an image of a porous material sample) that maximize the activation of the filters in different layers of our trained network [94]. To create such an input image we start with a random image $\mathbf{x}_0$ - a randomly chosen image of black and white pixels (represented by 0s and 1s). Next, we perform a forward pass using the input image $\mathbf{x}_0$, in order to compute the activation of a filter i in a layer j, $a_i^{[j]}(\mathbf{x})$. Then, we perform a backpropagation [95] (a backward pass), in order to obtain a gradient of the activation of $a_i^{[j]}(\mathbf{x})$ in respect to the previous layers' activation in CNN. At the end of the backward pass, we are left with the gradient of $a_i^{[j]}$ in respect to $\mathbf{x}$, $\nabla_x a_i^{[j]}(\mathbf{x})$. Next, we update the image with the gradient ascent step: $\mathbf{x}_{n+1} = \mathbf{x}_n + \alpha \nabla_x a_i^{[j]}(x)$, where α is a learning rate and n is an iteration number [93, 94]. In this work we chose $\alpha = 1.0$. We repeat this procedure for each kernel (filter) in a given layer j for $n^* = 40$ times, to get an image that activates the kernel $\mathbf{x}^* = \mathbf{x}_{40}$.

III. RESULTS

A. Packing preparation and percolation

The percolation probability thorough the packing at the porosity ϕ, generated as in Section II A, for given L and k scales with the system size L as:

$$P_{L,k}(\phi) = f_k\Big(\left(\phi - \phi_k^c\right) L^{1/\nu}\Big) \tag{2}$$

where ν is the correlation length exponent, f_k is a scaling function, and ϕ_k^c is the critical porosity. The critical porosity itself depends on the system size, and we define $\phi_k^c(L)$ as the porosity at which $P_{L,k}(\phi) = 0.5$. This values have been obtained by fitting percolation probability function $P_{L,k}(\phi)$ to a curve with a sigmoid shape, similarly to Koza et al. [66]

$$P_{L,k} \approx \frac{1}{2}\mathrm{erfc}\Big\{ \left[\phi - \phi_k^c(L)\right] / \Delta(L) \Big\} \quad (3)$$

where $\Delta(L)$ is a system dependent width of the percolation transition. The percolation theory assumes that the correlation length (the scale separating heterogeneity along the percolating cluster) is smaller than the linear size of the system L. Near the percolation threshold, the correlation length ξ_k scales as $\xi_k \propto (\phi - \phi_k^c)^{-1/\nu}$. In a consequence, ξ_k approaches infinity as the percolation threshold is approached, and in the system of a finite size L, ξ_k may be lager than L. When $\xi_k > L$, the apparent (i.e. for the finite system size) percolation threshold $\phi_k^c(L)$ is smaller than for the infinite system and depends on the system size (in the first order) as $\phi_k^c - \phi_k^c(L) \propto L^{-1/\nu}$ [52, 71, 96]. Thus, having estimated percolation thresholds for finite sizes, the value for $L \to \infty$ can be extrapolated, c.f. Table I. In this work, we performed this calculation to check the consistency with the previous work [66]. The packings through which the void space percolates in at least on directions are used in the lattice Boltzmann calculations, Section **??**.

L	$k = 2$	$k = 4$
16	0.4392(8)	0.3800(5)
32	0.4506(1)	0.3564(7)
64	0.4592(6)	0.3780(5)
128	0.4700(6)	0.3922(6)
256	0.4800(2)	0.4112(8)
512	0.4822(6)	0.4121(8)
1024	0.4857(2)	0.4128(5)
∞[66]	0.4868(1)	0.4172(2)

TABLE I. Percolation thresholds calculated from Eq. (3), for different system sizes L, and $k = 2, 4$. The asymptotic values for $L \to \infty$ are taken from Koza et al. [66].

B. Model training

We trained Deep-Rock architecture on 4288 percolating packings, and tested it on 1073 packings with periodic boundary padding (and non-periodic boundary padding for comparison), Fig. 3(a). We can see that the loss function value steadily goes down for the training set (orange lines in Fig 3(a)), but the prediction quality saturates after 20-40 epochs and then the network tends to over-fit. Standard way of dealing with this kind of overfitting would be to use more data. However in this work, it seems that some regularization can be a more appropriate approach. For the further model inspection, we choose the model that provides on the training data-set MSE=0.01179. This value is better than the best fit to Kozeny-Carman model [47, 48].

The K-C model is widely used [51, 56, 58, 59, 67–69, 74, 75, 80, 97–103]. To compare KC model to the CNN prediction, we fitted KC model, that relates material permeability and reads:

$$\kappa = C_\kappa(\phi - \phi_c)^\alpha \frac{\phi^2}{(1-\phi)^2} \quad (4)$$

where ϕ is material porosity, ϕ_c is percolation threshold, α encompass physics of the fluid transport process, and C_κ is a numerical constant. We treat these parameters as free, and found them with fitting to be $\phi_c = 0.3821$, $\alpha = 1.3208$, with the resulting MSE=0.01795, Fig. 3 (b). ConvNet

C. ConvNet filters visualisation

It is sometimes not clear why neural networks work at the accuracy they do. On the other hand, some very simple physics-based models such as Pore-network model, can match the accuracy of the more sophisticated methods [63–65].

In Figure 4 we visualize the weights of the first convolutional layer. These filters are usually the most interpretable on the first convolutional layer, which is looking directly at the raw pixel data, but it is possible to also show the filter weights deeper in the network. The weights are useful to visualize because well-trained networks usually display nice and smooth filters without any noisy patterns. Noisy patterns can be an indicator of a network that has not been trained for long enough, or possibly a very low regularization strength that may have led to over-fitting - Figure 3 (a). - Comment more on what these filters seem to be doing.

IV. DISCUSSION

On one hand, Deep Learning has the advantage over physics-based computational methods, because they can suffer from some problems when evaluating transport properties in a discrete domain, close to the critical point [104, 105]. On the other hand, in order to train a robust DL model, a large training dataset it required. Since the experimental data is scarce, we may leverage the idea of transfer learning [], where some key features are extracted (by a neural network) from the large set of heterogeneous samples (for example samples that are artificially generated, like in our work), and then the model is then retrained for a particular application were only the small amount of data is available. Another challenge to overcome in DL is the issue of the impact of the resolution of the input data on the final fluid transport properties prediction. A popular approach is to obtain Micro-CT scans of geological samples, and then represent the material as a set of voxels (or pixels for 2D materials) that are permeable or impermeable to the hypothetical liquid. This

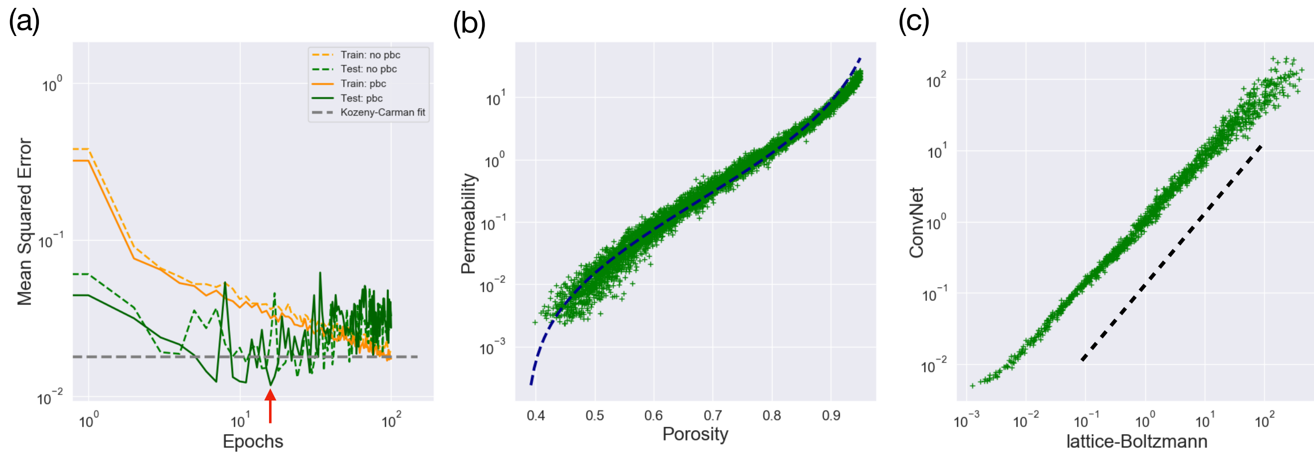

FIG. 3. (a) Progress of the Deep-Rock architecture training: for periodic boundary condition image padding (solid lines) and non-periodic boundary condition (dashed lines). Training set consists of 4288 images, and the test set consist of 1073 images. Weights for the best performing network on the training set are saved (denoted by a red arrow). Loss function value for the best weights is MSE=0.01179. Data for the non-periodic boundary padding is given for comparison. Horizontal grey dashed-line gives the error for the optimal fit of the Kozeny-Carman model. Note: Log-log scale. (b) The optimal fit (dashed blue lines) of the Kozeny-Carman model $\kappa \propto (\phi - \phi_c)^\alpha \phi^2/(1-\phi)^2$(Eq. 4), where $\phi_c = 0.3821$, $\alpha = 1.3208$ (MSE= 0.01795). (c) Quantitative comparison of the performance of the best trained network and LB simulations. Black dashed line gives the linear dependence with slope 1.

however presents a challenge to both the physics-based and machine learning methods for whichtheir numerical performance depends on the processing and handling the digitalized data [57–59, 61, 106–115] (especially close to the critical point [116]). Thus, the further research needs to address how degradation of the images resolution impacts the performance of the pre-trained neural network.

V. ACKNOWLEDGMENTS

I thank Tomasz Konopczynski for help with the Keras (https://keras.io/) implementation of the periodic boundary conditions padding.

VI. APPENDIX

A. Source Code Availability

All of the source codes of the *Deep Rock Project* are available on GitHub [88].

[1] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, in *Proceedings of the IEEE* (1998) pp. 2278–2324.

[2] Y. LeCun, Y. Bengio, and G. E. Hinton, Nature **521**, 436 (2015).

[3] A. Krizhevsky, I. Sutskever, and G. Hinton (Curran Associates, Inc., 2012) pp. 1097–1105.

[4] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, in *NIPS Deep Learning Workshop* (2013).

[5] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, Nature **518**, 529 (2015).

[6] H. v. Hasselt, A. Guez, and D. Silver, in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16 (2016) pp. 2094–2100.

[7] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, Nature **529**, 484 (2016).

[8] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, and D. Hassabis, Nature **550**, 354 (2017).

[9] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran,

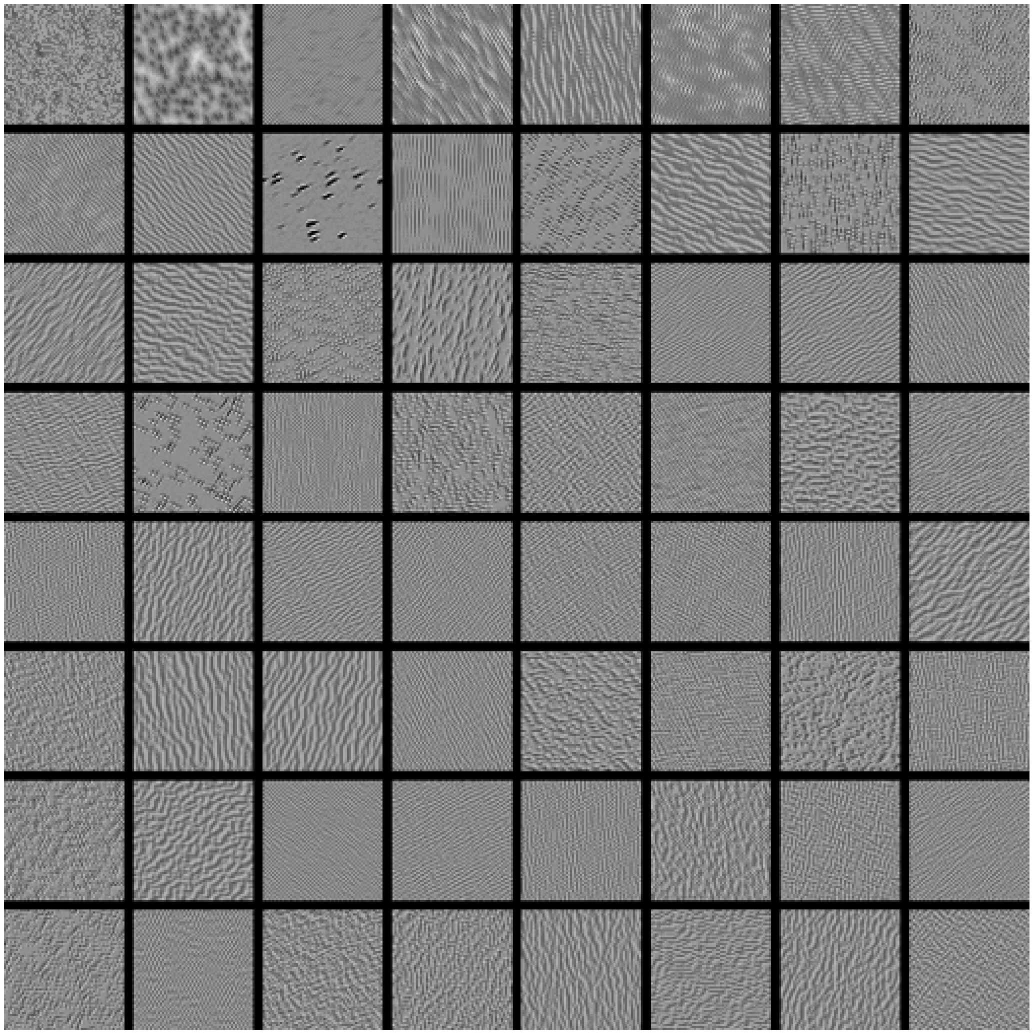

FIG. 4. Filters visualisation in the first convolutional layer of Deep-Rock architecture (Fig.2). The visualisations are given in the descending order (left-to-right, top-to-bottom) in terms of the network activation, see Section **??** for more details.

T. Graepel, T. Lillicrap, K. Simonyan, and D. Hassabis, Science **362**, 1140 (2018).

[10] O. Ronneberger, P.Fischer, and T. Brox, in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, LNCS, Vol. 9351 (2015) pp. 234–241.

[11] K. He, G. Gkioxari, P. Dollár, and R. Girshick, in *Proceedings of the International Conference on Computer Vision (ICCV)* (2017).

[12] T. Mikolov, K. Chen, G. Corrado, and J. Dean, CoRR **abs/1301.3781** (2013).

[13] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, in *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13 (2013) pp. 3111–3119.

[14] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, Nature **542**, 115 (2017).

[15] P. Rajpurkar, A. Y. Hannun, M. Haghpanahi, C. Bourn, and A. Y. Ng, CoRR **abs/1707.01836** (2017), arXiv:1707.01836.

[16] J. Graser, S. Kauwe, and T. Sparks, Chemistry of Materials **30**, 3601 (2018).

[17] K. Ryan, J. Lengyel, and M. Shatruk, Journal of the American Chemical Society **140**, 10158 (2018).

[18] W. Ye, C. Chen, Z. Wang, L.-H. Chu, and S. Ong, Nature Communications **9**, 3800 (2018).

[19] G. Hegde and R. C. Bowen, Scientific Reports **7**, 42669 (2017).

[20] L. Cheng, M. Welborn, and T. I. Miller, (2019), arXiv:1901.03309.

[21] L. Wang, Phys. Rev. B **94**, 195105 (2016).

[22] K. Ch'ng, J. Carrasquilla, R. G. Melko, and E. Khatami, Phys. Rev. X **7**, 031038 (2017).

[23] J. Carrasquilla1 and R. G. Melko, Nature **13**, 431 (2017).

[24] W. Hu, R. R. P. Singh, and R. T. Scalettar, Phys. Rev. E **95**, 062122 (2017).

[25] S.-H. Li and L. Wang, Phys. Rev. Lett. **121**, 260601 (2018).
[26] B. F.A., J. Gomes, R. Sharma, F. Lee, and V. Pande, (2018), arXiv:1803.08993.
[27] R. Xu, W. Fu, and H. Zhao, (2019), arXiv:1901.00774.
[28] A. Amidi, S. Amidi, D. Vlachakis, V. Megalooikonomou, N. Paragios, and E. Zacharaki, ArXiv e-prints arXiv:1707.06017.
[29] I. Wallach, M. Dzamba, and A. Heifets, ArXiv e-prints arXiv:1510.02855.
[30] G. Torlai and R. G. Melko, Phys. Rev. B **94**, 165134 (2016).
[31] K. Mills, M. Spanner, and I. Tamblyn, Phys. Rev. A **96**, 042113 (2017).
[32] P. Broecker, J. Carrasquilla, R. Melko, and S. Trebst, Scientific Reports **7**, 8823 (2017).
[33] H. Y. D. Sigaki, R. F. de Souza, R. T. de Souza, R. S. Zola, and H. V. Ribeiro, Phys. Rev. E **99**, 013311 (2019).
[34] C. Shen, Water Resources Research (2017), 10.1029/2018WR022643.
[35] J. N. Kutz, Journal of Fluid Mechanics **814**, 14 (2017).
[36] T. Miyanawala and R. Jaiman, (2017), arXiv:1710.09099.
[37] O. Hennigh, (2017), arXiv:1705.09036.
[38] D. Stoecklein, K. Lore, M. Davies, S. Sarkar, and B. Ganapathysubramanian, Scientific Reports **7**, 46368 (2017).
[39] A. Farimani, J. Gomes, and V. Pande, (2017), arXiv:1709.02432.
[40] R. King, O. Hennigh, A. Mohan, and C. M., (2018), arXiv:1810.07785.
[41] N. Srisutthiyakorn, "Deep-learning methods for predicting permeability from 2d/3d binary-segmented images," in *SEG Technical Program Expanded Abstracts 2016* (2016) pp. 3042–3046.
[42] O. Arigbe, M. Oyeneyin, I. Arana, and M. Ghazi, Journal of Petroleum Exploration and Production Technology , 1 (2018).
[43] J. Wu, X. Yin, and H. Xiao, Science Bulletin **63**, 1215 (2018).
[44] O. Sudakov, E. Burnaev, and D. Koroteev, (2018), arXiv:1803.00758.
[45] M. Vasilyeva and A. Tyrylgin, (2018), arXiv:1810.01586.
[46] S. Mo, Y. Zhu, N. Zabaras, X. Shi, and J. Wu, (2018), arXiv:1807.00882.
[47] J. Kozeny, Sitzungsber Akad. Wiss. Wien. **136**, 271 (1927).
[48] P. C. Carman, Transactions-Institution of Chemical Engineers **15**, 150 (1937).
[49] B. Halperin, S. Feng, and P. Sen, Physical Review Letters **54**, 2391 (1985).
[50] S. Feng, B. Halperin, and P. Sen, Physical Review B **35**, 197 (1987).
[51] N. S. Martys, S. Torquato, and D. Bentz, Physical Review E **50**, 403 (1994).
[52] H. Daigle, Advances in Water Resources **96**, 43 (2016).
[53] a. M. G. Srisutthiyakorn, N, SEG International Exposition and 87th Annual Meeting , 3811 (2017).
[54] N. Nishiyama and T. Yokoyama, Journal of Geophysical Research: Solid Earth **122**, 6955 (2017).
[55] J. Koestel, A. Dathe, T. H. Skaggs, O. Klakegg, M. A. Ahmad, M. Babko, D. Gimenez, C. Farkas, A. Nemes, and N. Jarvis, Water Resources Research **54**, 9255 (2018).
[56] P. Gniewek and O. Hallatschek, Phys. Rev. E **99**, 023103 (2019).
[57] J.-F. Gaillard, C. Chen, S. Stonedahl, B. Lau, D. Keane, and A. Packman, Geophysical Research Letters **34**, L18404 (2007).
[58] C. Chen, A. I. Packman, and J.-F. Gaillard, Geophysical Research Letters **35**, L07404 (2008).
[59] C. Berg and R. Held, Transport in Porous Media **112**, 467 (2016).
[60] I. Fatt, Pet. Trans. AIME **207**, 144 (1956).
[61] M. J. Blunt, B. Bijeljic, H. Dong, O. Gharbi, S. Iglauer, P. Mostaghimi, A. Paluszny, and C. Pentland, Advances in Water Resources **51**, 197 (2013).
[62] H. Li, C. Pan, and C. T. Miller, Phys. Rev. E **72**, 026705 (2005).
[63] A. Q. Raeini, B. Bijeljic, and M. J. Blunt, Phys. Rev. E **96**, 013312 (2017).
[64] K. Alim, S. Parsa, D. A. Weitz, and M. P. Brenner, Phys. Rev. Lett. **119**, 144501 (2017).
[65] A. Q. Raeini, B. Bijeljic, and M. J. Blunt, Phys. Rev. E **97**, 023308 (2018).
[66] Z. Koza, G. Kondrat, and K. Suszczyński, Journal of Statistical Mechanics: Theory and Experiment **2014**, P11005 (2014).
[67] A. Koponen, M. Kataja, and J. Timonen, Physical Review E **54**, 406 (1996).
[68] A. Koponen, M. Kataja, and J. Timonen, Physical Review E **56**, 3319 (1997).
[69] M. Matyka, A. Khalili, and Z. Koza, Physical Review E **78**, 026306 (2008).
[70] A. Duda, Z. Koza, and M. Matyka, Physical Review E **84**, 036319 (2011).
[71] M. D. Rintoul and S. Torquato, Journal of Physics A: Mathematical and General **30**, L585 (1997).
[72] Z. Koza, M. Matyka, and A. Khalili, Physical Review E **79**, 066306 (2009).
[73] S. Succi, *The lattice Boltzmann equation: for fluid dynamics and beyond* (Oxford University Press, 2001).
[74] A. Cancelliere, C. Chang, E. Foti, D. H. Rothman, and S. Succi, Physics of Fluids A: Fluid Dynamics **2**, 2085 (1990).
[75] M. Matyka and Z. Koza, in *AIP Conference Proceedings 4* (AIP, 2012) pp. 17–22.
[76] M. Matyka, J. Gołembiewski, and Z. Koza, Phys. Rev. E **93**, 013110 (2016).
[77] M. Agnaou, D. Lasseux, and A. Ahmadi, Physical Review E **96**, 043105 (2017).
[78] M. Espinoza-Andaluz, A. Moyn, and M. Andersson, Computers Mathematics with Applications (2019), https://doi.org/10.1016/j.camwa.2019.02.012.
[79] K. N. Premnath and J. Abraham, Journal of Computational Physics **224**, 539 (2007).
[80] C. Chen, B. L. Lau, J.-F. Gaillard, and A. I. Packman, Water Resources Research **45**, W06416 (2009).
[81] J. Latt, Choice of units in lattice Boltzmann simulations (2008).
[82] .
[83] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, J. Mach. Learn. Res. **15**, 1929 (2014).
[84] S. Mittal, Neural Computing and Applications (2018).

[85] M. Riesenhuber and T. Poggio, Nature Neuroscience **2**, 1019 (1999).

[86] X. Glorot, A. Bordes, and Y. Bengio, in *AISTATS*, JMLR Proceedings, Vol. 15, edited by G. J. Gordon, D. B. Dunson, and M. Dudk (JMLR.org, 2011) pp. 315–323.

[87] F. Chollet *et al.*, "Keras," `https://keras.io` (2015).

[88] P. Gniewek, "Deep-rock," `https://github.com/pgniewko/Deep-Rock` (2019).

[89] K. Simonyan, A. Vedaldi, and A. Zisserman, CoRR **abs/1312.6034** (2013).

[90] M. D. Zeiler and R. Fergus, Computer Vision (ECCV 2014) **8689** (2014).

[91] A. Mahendran and A. Vedaldi, in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015* (2015) pp. 5188–5196.

[92] J. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, in *ICLR (workshop track)* (2015).

[93] J. Yosinski, J. Clune, A. M. Nguyen, T. J. Fuchs, and H. Lipson, CoRR **abs/1506.06579** (2015).

[94] D. Erhan, Y. Bengio, A. Courville, and P. Vincent, *Visualizing Higher-Layer Features of a Deep Network*, Tech. Rep. 1341 (University of Montreal, 2009) also presented at the ICML 2009 Workshop on Learning Feature Hierarchies, Montréal, Canada.

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

[96] M. E. Fisher, Physics Physique Fizika **3**, 255 (1967).

[97] K. Ng, AIChE Journal **32**, 115 (1986).

[98] M. J. MacDonald, C.-F. Chu, P. P. Guilloit, and K. M. Ng, AIChE Journal **37**, 1583 (1991).

[99] G. Mavko and A. Nur, Geophysics **62**, 1480 (1997).

[100] A. Costa, Geophysical Research Letters **33** (2006).

[101] C. Arns, M. Knackstedt, and N. Martys, Physical Review E **72**, 046304 (2005).

[102] C. Jin, P. A. Langston, G. E. Pavlovskaya, M. R. Hall, and S. P. Rigby, Physical Review E **93**, 013122 (2016).

[103] P. Xu and B. Yu, Advances in Water Resources **31**, 74 (2008).

[104] S. Schnyder, M. Spanner, F. Hofling, T. Franosch, and J. Horbach, Soft Matter **11**, 701 (2015).

[105] M. Spanner, F. Höfling, S. C. Kapfer, K. R. Mecke, G. E. Schröder-Turk, and T. Franosch, Phys. Rev. Lett. **116**, 060601 (2016).

[106] B. Bijeljic, A. Raeini, P. Mostaghimi, and M. J. Blunt, Phys. Rev. E **87**, 013011 (2013).

[107] H. Andra, N. Combaret, J. Dvorkin, E. Glatt, J. Han, M. Kabel, Y. Keehm, F. Krzikalla, M. Lee, C. Madonna, M. Marsh, T. Mukerji, E. H. Saenger, R. Sain, N. Saxena, S. Ricker, A. Wiegmann, and X. Zhan, Computers & Geosciences **50**, 33 (2013).

[108] A. Raeini, M. Blunt, and B. Bijeljic, Advances in Water Resources **74**, 116 (2014).

[109] N. Alyafei, P. A. Raeini, A.Q., and M. Blunt, Transport in Porous Media **74**, 116 (2015).

[110] M. K. Misztal, A. Hernandez-Garcia, R. Matin, D. Müter, D. Jha, H. O. S, and J. Mathiesen, in *Front. Phys.*, Vol. 3 (2015).

[111] B. Muljadi, M. Blunt, A. Raeini, and B. Bijeljic, Advances in Water Resources **95**, 329 (2016).

[112] L. Mosser, O. Dubrule, and M. J. Blunt, Phys. Rev. E **96**, 043309 (2017).

[113] P. Tahmasebi, M. Sahimi, and J. Andrade, Geophysical Research Letters **44**, 4738 (2017).

[114] T. Ritschel, S. Schluter, J. Kohne, H.-G. Vogel, and K. Totsche, Water Resources Research **54**, 9033 (2018).

[115] N. Saxena, A. Hows, R. Hofmann, F. O. Alpak, J. Freeman, S. Hunter, and M. Appel, Advances in Water Resources **116**, 127 (2018).

[116] J. Liu and K. Regenauer-Lieb, Phys. Rev. E **83**, 016106 (2011).