Supplementary Information for

A Fully Differentiable Pore Network for Digital Reconstruction of Porous Media

Michael McKague[1], Mohammad Mehrnia[1], Mohammad Amin Sadeghi[1], Jeff Gostick[1,*]

[1] Department of Chemical Engineering, University of Waterloo, ON, Canada

[*]Corresponding author, jgostick@uwaterloo.ca

Network Size

To determine the size of network that is reasonably representative of an actual material, a Weibull distribution was fitted to the pore size distribution (psd) of the Berea image followed by taking a number of differently sized samples to see at what size the sample converges to the shape of the Berea psd. Figure S1 compares the shape of four differently sized samples from the true Weibull distribution (in red). The number of samples for reach represented a) 5^3, b) 8^3, c) 10^3, and d) 15^3 pore networks. Visually you can see how a 125 sized sample is not large enough to represent the actual pore size distribution well. Increasing the number of samples further, it becomes visually apparent that a 1000 sized sample is large enough to reasonably represent the shape of the true psd. While increasing the number of samples further improves the fit to the Weibull distribution, considering computational cost we decided to focus our work on optimizing a 10^3 network. It is worth mentioning that a network of this size has nearly three thousand throats and therefore would have a throat size distribution (tsd) that is even more representative of the true tsd.

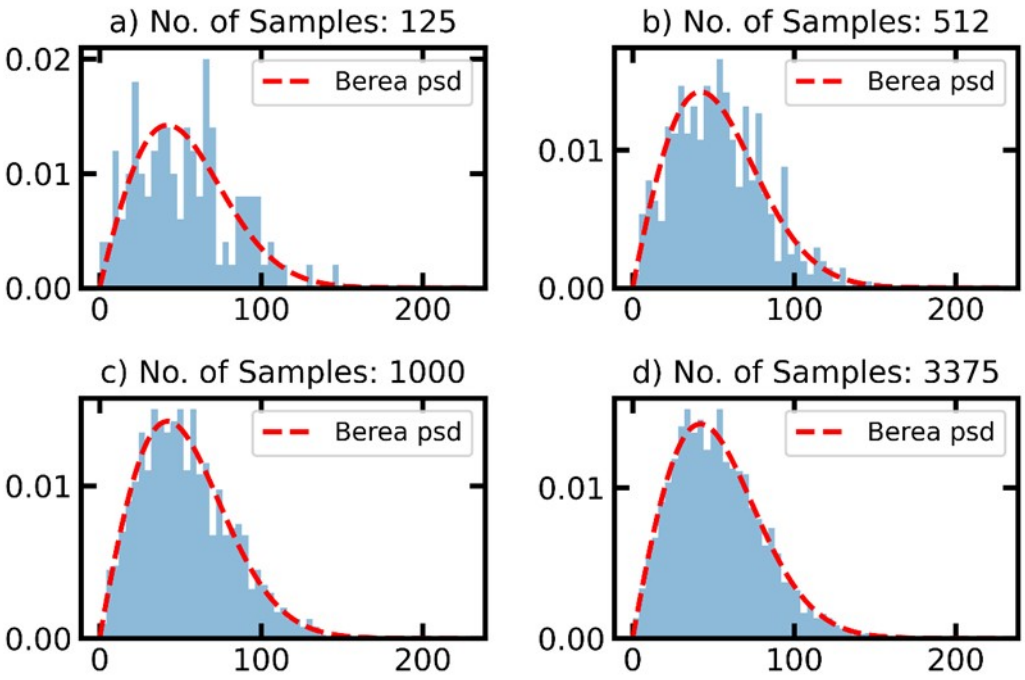

Figure S1 Different number of samples taken from a Weibull distribution fitted to the pore size distribution of the Berea image. We tried a) 125, b) 512, c) 1000, and d) 3375 number of samples.

Hydraulic Size Factor

In this work, the fitted pore networks assumed spheres and cylinders geometry with known hydraulic conductance models. The hydraulic conductance is dependent on the hydraulic size factor, λ_i^h, for element i in a pore-throat conduit. The hydraulic size factor for an element of arbitrary cross-section is given by equation (S1) where I_p^* is the specific polar moment of inertia and $A_i(x)$ is the cross-sectional area. The integral spans the entire length, l_i, of pore/throat element i.

$$1/\lambda_i^h = 16\pi^2 \int_0^{l_i} \frac{I_p^*}{A_i(x)^2} dx \qquad \textbf{(S1)}$$

In the spheres and cylinders geometry, both elements have a circular cross-section and therefore, both elements have the same specific polar moment of inertia. Therefore, we use the moment of inertia of a circle for both pores and throats as given by equation (S2).

$$I_p^* = \frac{1}{2\pi} \tag{S2}$$

Next, we take the integral of $1/A_i(x)$ with respect to x for both sphere and cylinder geometries. The throat is represented as a cylinder and therefore its cross-sectional area is constant along dimension x and its integral is given simply by equation (S3). Spheres, on the other hand, have a variable cross-sectional area and must be calculated knowing that the relationship between the cross-sectional area and distance x is $\pi\left(D_p^2/4 - x^2\right)$ to integrate and arrive at equation (S4).

$$\int_0^{l_t} \frac{1}{A_t(x)^2} dx = \frac{l_t}{\left(\pi D_t^2/4\right)^2} \tag{S3}$$

$$\int_0^{l_p} \frac{1}{A_p(x)^2} dx = \frac{4}{D_p^3 \pi^2}\left(\frac{2D_p L_p}{D_p^2 - 4L_p^2} + \tan^{-1}\left(\frac{2L_p}{D_p}\right)\right) \tag{S4}$$

Data Set

A total of 13 sample images obtained from the work by Dong and Blunt [1] were used to obtain mock experimental data for testing our optimization strategy. Table S1 summarizes some of the properties of the images including resolution, size, rock type, and porosity. The permeability in the x, y, and z directions determined from Lattice Boltzmann simulations were also included in the table. Figure S2 shows renderings, after digital reconstruction, for all 13 of the sampled images.

Table S1 Properties of sample images

Samples*	Resolution[a]	Size[b]	Rock Type[c]	Porosity (%)	K_x[d]	K_y[d]	K_z[d]	K_{avg}[d]
Berea	5.35	400^3	Sandstone	19.6	1659	1801	1872	1777
C1	2.85	400^3	Carbonate	23.3	1400	2355	1185	1647
C2	5.35	400^3	Carbonate	16.8	138	319	161	206
S1	8.68	300^3	Sandstone	14.1	1976	2513	2740	2410
S2	4.96	300^3	Sandstone	24.6	4270	4814	5252	4779
S3	9.10	300^3	Sandstone	16.9	723	1185	784	897
S4	8.96	300^3	Sandstone	17.1	415	597	573	528
S5	4.00	300^3	Sandstone	21.1	6474	6462	6223	6386
S6	5.10	300^3	Sandstone	24.0	13995	15065	18246	15769
S7	4.80	300^3	Sandstone	25.0	7621	10235	9349	9068
S8	4.89	300^3	Sandstone	34.0	15322	16298	15610	15743
S9	3.40	300^3	Sandstone	22.2	2357	2694	3319	2790
A1	3.85	300^3	Sandpack	42.9	6524	9293	9607	8475

[a] *The resolution of the image is the length of one side of a voxel in μm*

[b] *The size of the image is the number of voxels in the image*

[c] *The sample is one of three rock types: sandstone, carbonate, and sandpack materials*

[d] *The permeability measured using Lattice Boltzmann on each image in the x, y, and z directions recorded in units of mD*

**The sample images were obtained from the work by Dong and Blunt titled, "Pore-network extraction from micro-computerized-tomography images" (2009)*

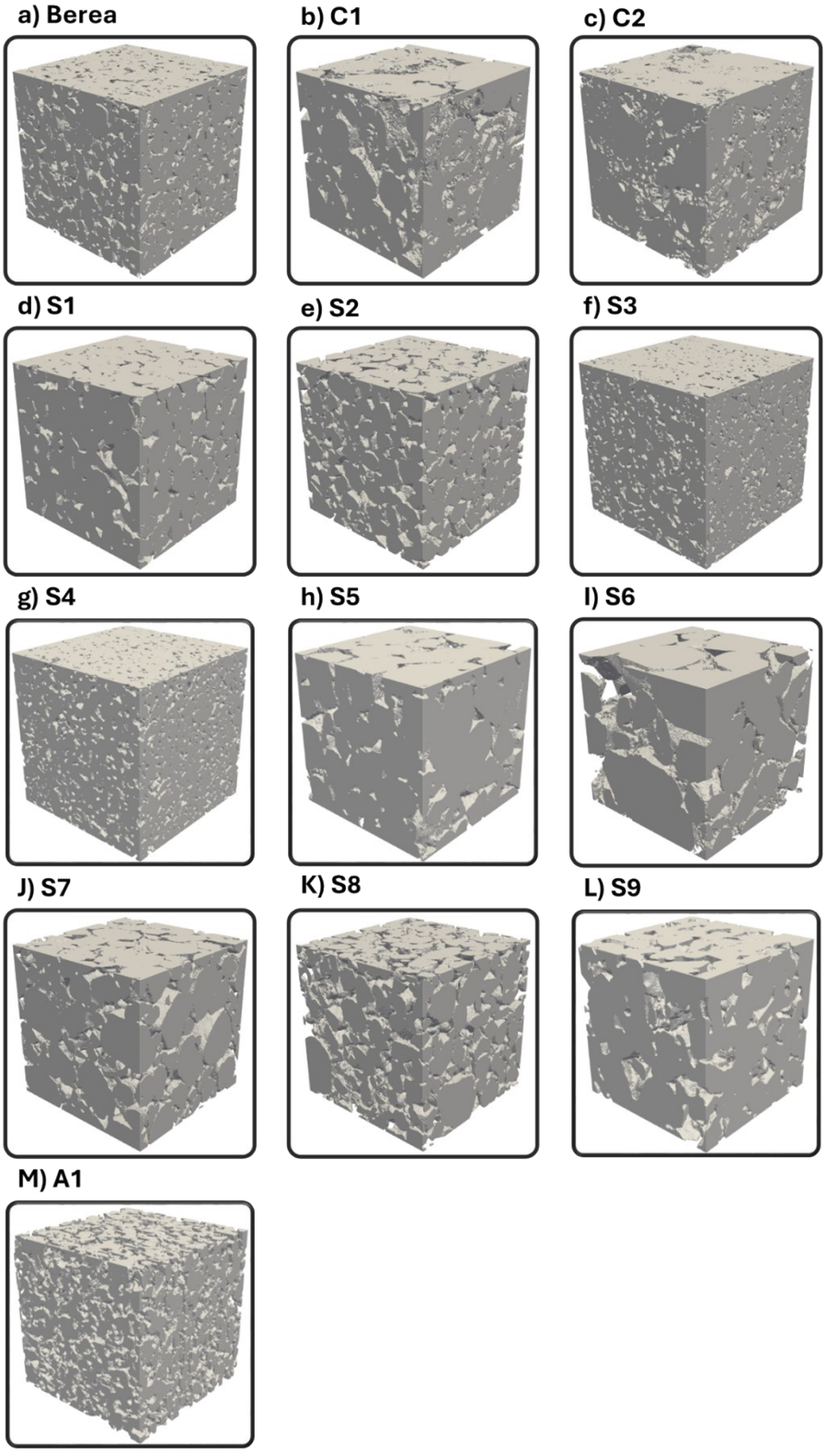

Figure S2 Digital renderings for all 13 sampled images

We confirmed our simulated data by comparing results from the Berea image to real experimental data. The work by Churchel et al. includes permeability and porosimetry data for a number of Berea sandstone samples that we could use for comparison. While the properties of Berea sandstone can vary significantly from one sample to the next, Churchel et al. found a relationship between permeability and breakthrough pressure [2]. **Figure S3** shows the relationship between permeability and breakthrough pressure. The black markers represent real experimental data from the work by Churchel et al. while the grey line is the fitted model. The red marker is the breakthrough pressure and permeability obtained from direct simulations on the Berea image itself. See how the simulated results follows the trend that was observed experimentally by Churchel et al.

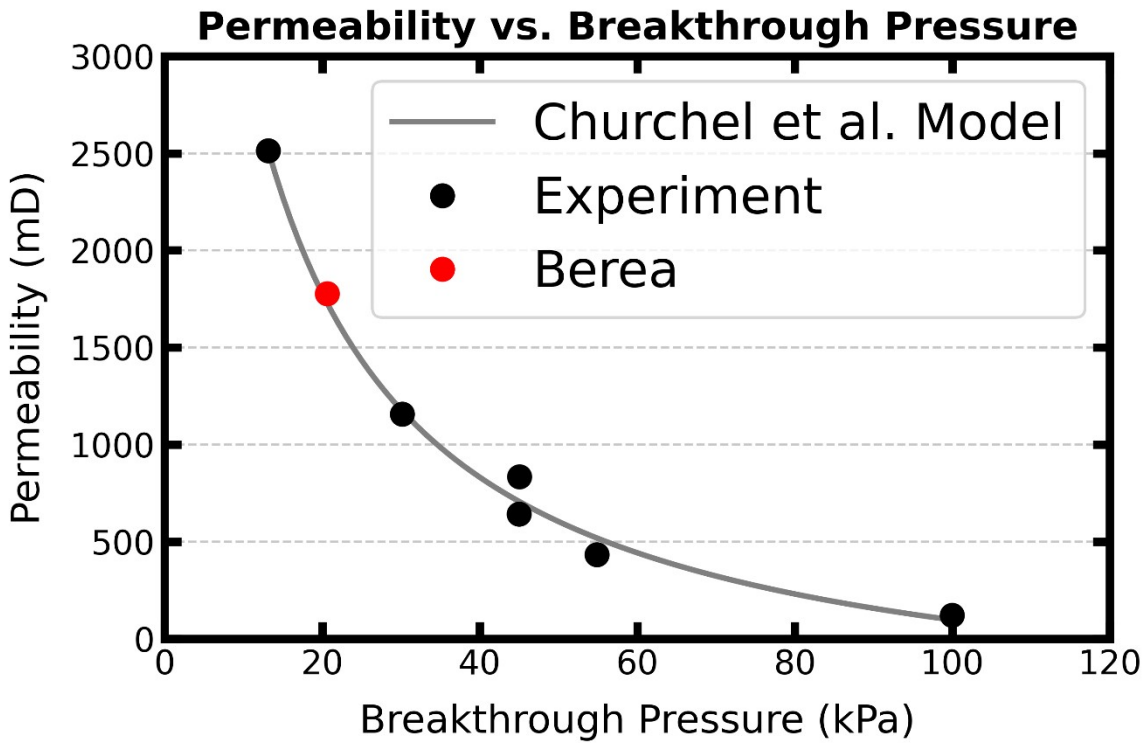

Figure S3 Plot showing the permeability versus breakthrough pressure relationship found experimentally by Churchel et al. for samples of Berea sandstone [2]. The experimental data is plotted using a black marker while the red marker shows the data obtained from simulation on the Berea image from the data set.

The breakthrough pressure in **Figure S3** is determined by finding the point of maximum slope on the saturation versus capillary pressure curve, drawing a tangent at that point, and extending the tangent to find the x-intercept. The pressure at which the tangent crosses the x-axis is the breakthrough pressure. Figure S3 shows this procedure applied to two different saturation curves. The black curve was obtained by performing a mercury invasion simulation directly on the Berea image while the blue curve was obtained by simulation on the extracted network obtained from SNOW. The reason why the SNOW intrusion curve is shifted to the left is because we used equivalent diameters obtained from the network extraction while the image-based invasion used inscribed sizes from the distance transform. Using both curves to obtain breakthrough pressure, it was found that the simulation on the extracted network using equivalent diameters matched experimental data the best. Therefore, the extracted network and its equivalent diameters were used to obtain porosimetry data for all images in the data set.

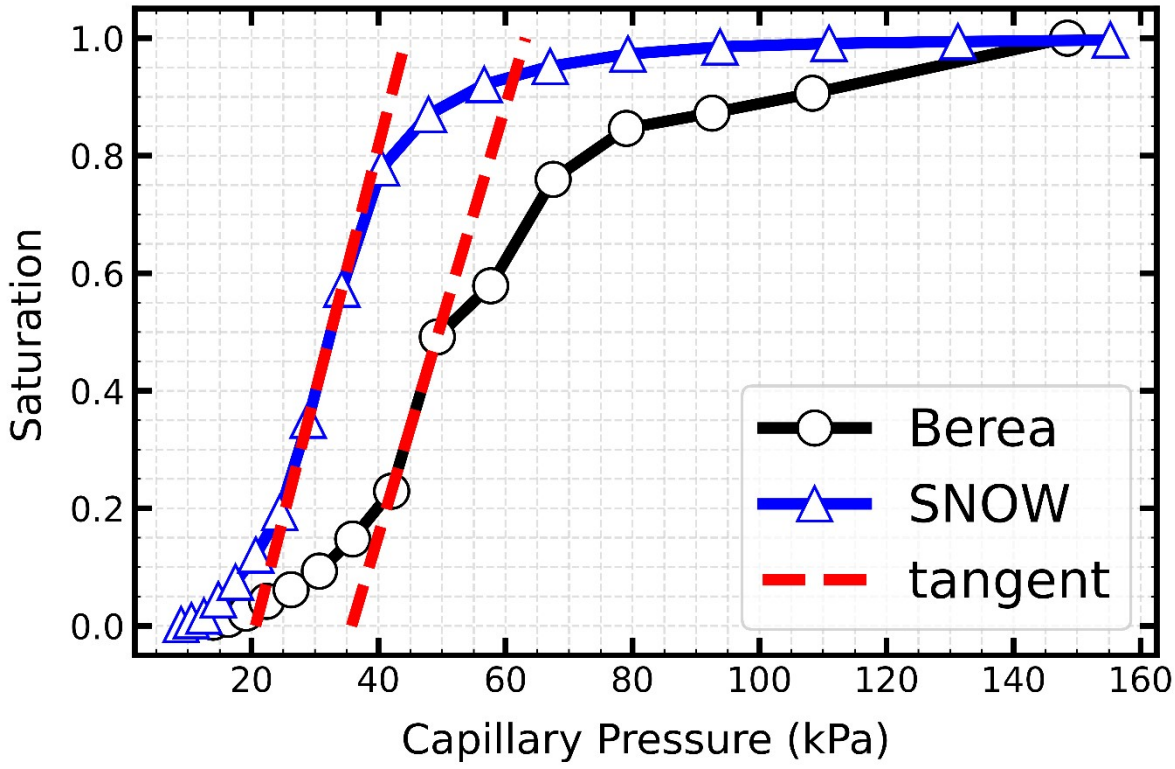

Figure S4 Saturation versus capillary pressure curves obtained from direct image-based simulation (black) and from simulation on the extracted network using equivalent diameters (blue). The tangent at the point of maximum slope is drawn for both saturation curves using a dotted red line. Where the tangent intercepts the x-axis is the breakthrough pressure.

Results

The geometric properties of a regular 10^3 pore network were optimized to match permeability and saturation data obtained for all samples. Figure S5 shows the resulting saturation curves after optimization (purple) compared to the target saturation (green) for all samples. The geometric properties were initially guessed by assuming the bundle of tubes model. The blue saturation curve represents the saturation obtained when using pore sizes from the bundle of tubes model. Similarly, **Figure S6** shows the resulting permeability before (blue) and after optimization (purple) for all samples. The exact same properties were assumed for optimization of all 13 samples including the number of iterations, learning rate, smoothing factor, and the way the initial guess was found. Finally, **Figure S7** compares the pore size distributions for all samples obtained from a SNOW network extraction (light blue), after optimization (purple), and from the bundle of tubes model (blue).

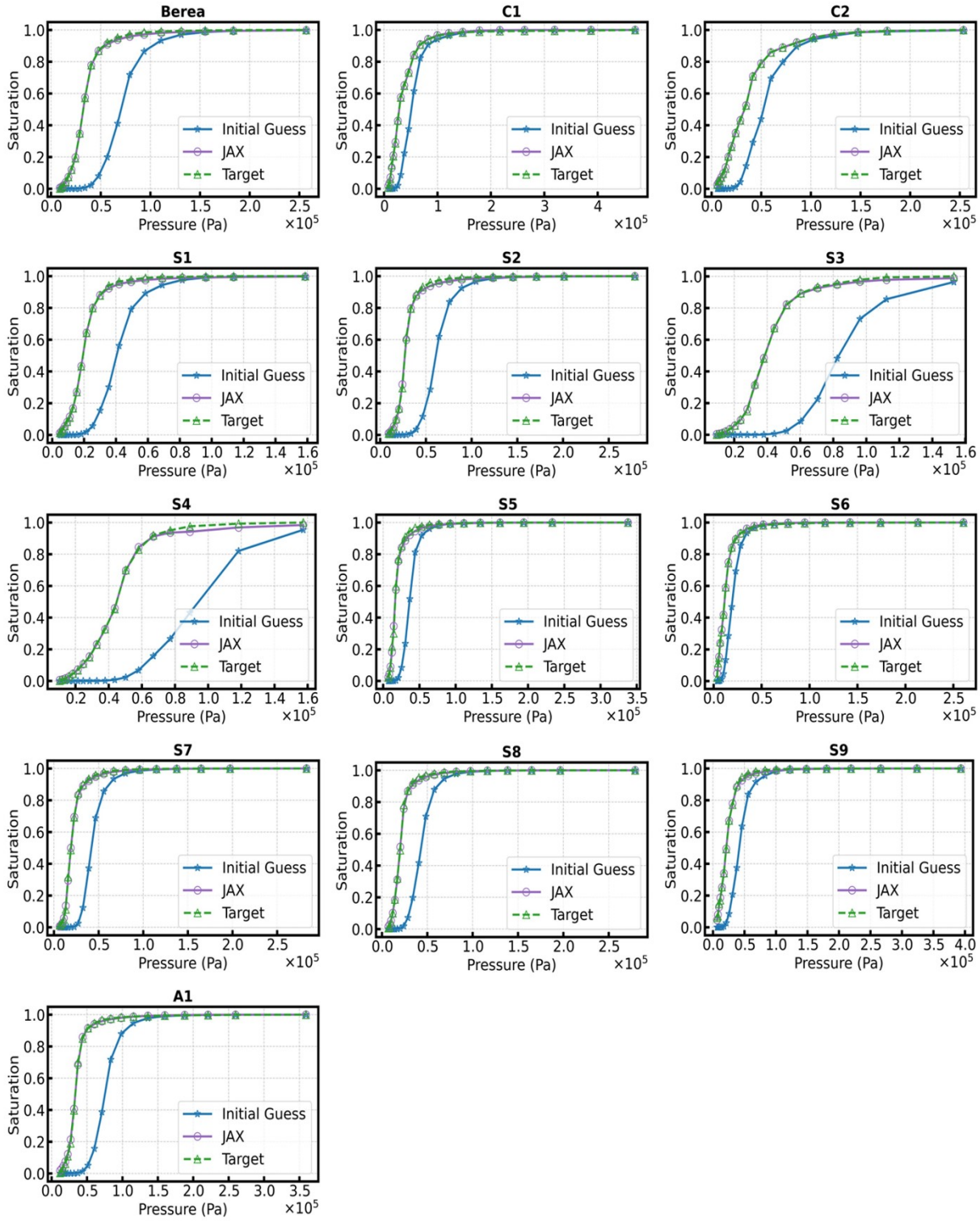

Figure S5 The saturation curves of all 13 samples before optimization (blue) and after optimization (purple) compared to the target saturation curve obtained from image-based simulation (green).

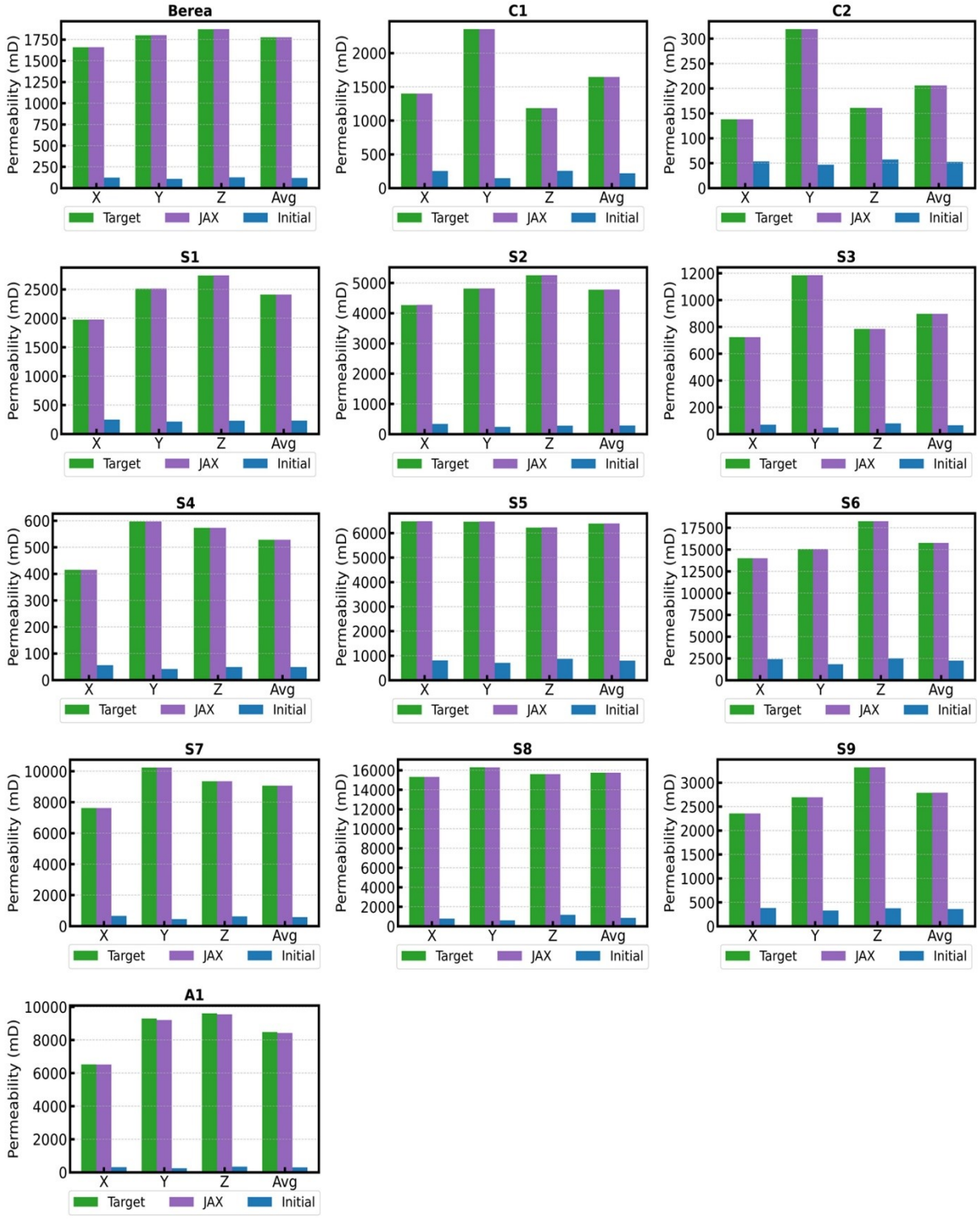

Figure S6 The permeability of all 13 samples prior to optimization (blue) and after optimization (purple). The green bar shows the target permeability. The permeability was measured and optimized in all three directions.

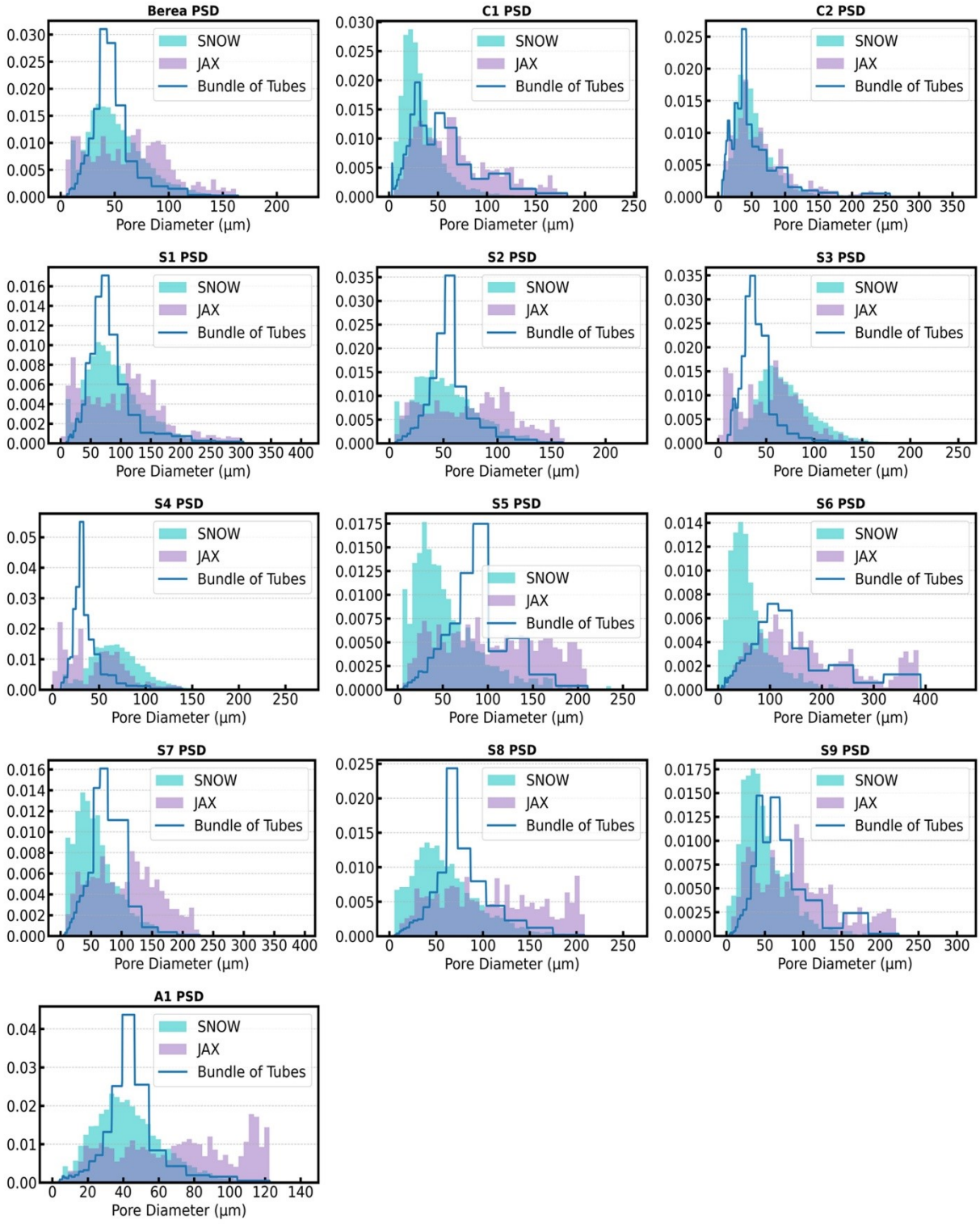

Figure S7 The pore size distributions for all 13 samples before optimization (blue line) and after optimization (purple) are shown. The pore sizes from the network extraction (blue) are also shown as these represent the ground truth for pore sizes obtained from the image.

After optimization, the pore networks that fit the experimental data were used to obtain stochastic networks of arbitrary size. Figure S8 shows the resulting pore networks for three of the samples: a) Berea, b) C2, and c) A1. The spatial correlations of the pore and throat sizes were learned by fitting the hyperparameters of a Gaussian Process (GP) model to the original 10^3 network. Table S2 shows the hyperparameters that were used including the signal variance and length scale for both GP models (i.e. pore diameter and aspect ratio) of Berea, C2, and A1 samples. The learned spatial correlations were then used to generate larger networks of arbitrary size with the same properties as the original network. Learning spatial correlations to generate arbitrarily sized networks was tested on three of the 13 samples and the complete set of results are shown for all three samples tested. Figure S9 shows the importance of using GP to obtain spatial correlations for matching the target properties.

Table S2 Hyperparameters used by gaussian process models to find spatial correlations for pore diameters and throat aspect ratios

Sample	*GP model*	σ^2	l
Berea	*Pore Diameter*	*0.121*	*0.080*
	Aspect Ratio	*0.041*	*0.122*
C2	*Pore Diameter*	*0.070*	*0.075*
	Aspect Ratio	*0.033*	*0.125*
A1	*Pore Diameter*	*0.208*	*0.083*
	Aspect Ratio	*0.058*	*0.116*

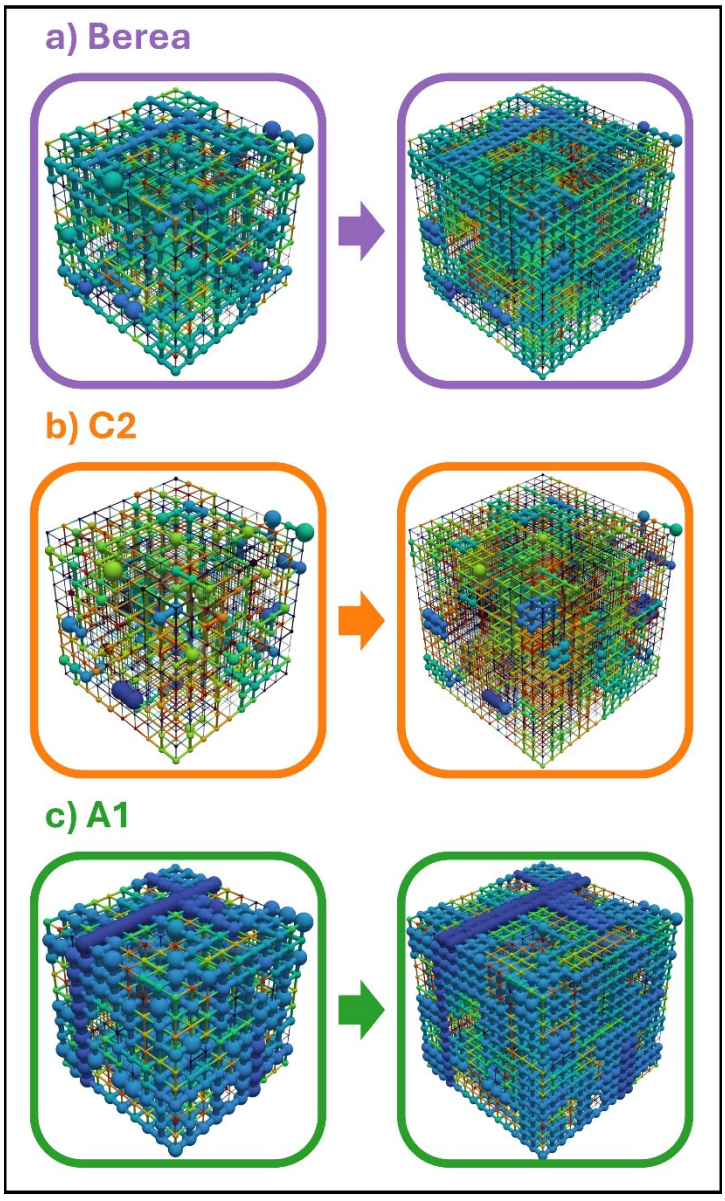

Figure S8 Optimized pore networks for a) Berea, b) C2, and c) S5 samples were scaled to a larger 15^3 network while maintaining learned spatial correlation of pore and throat sizes.

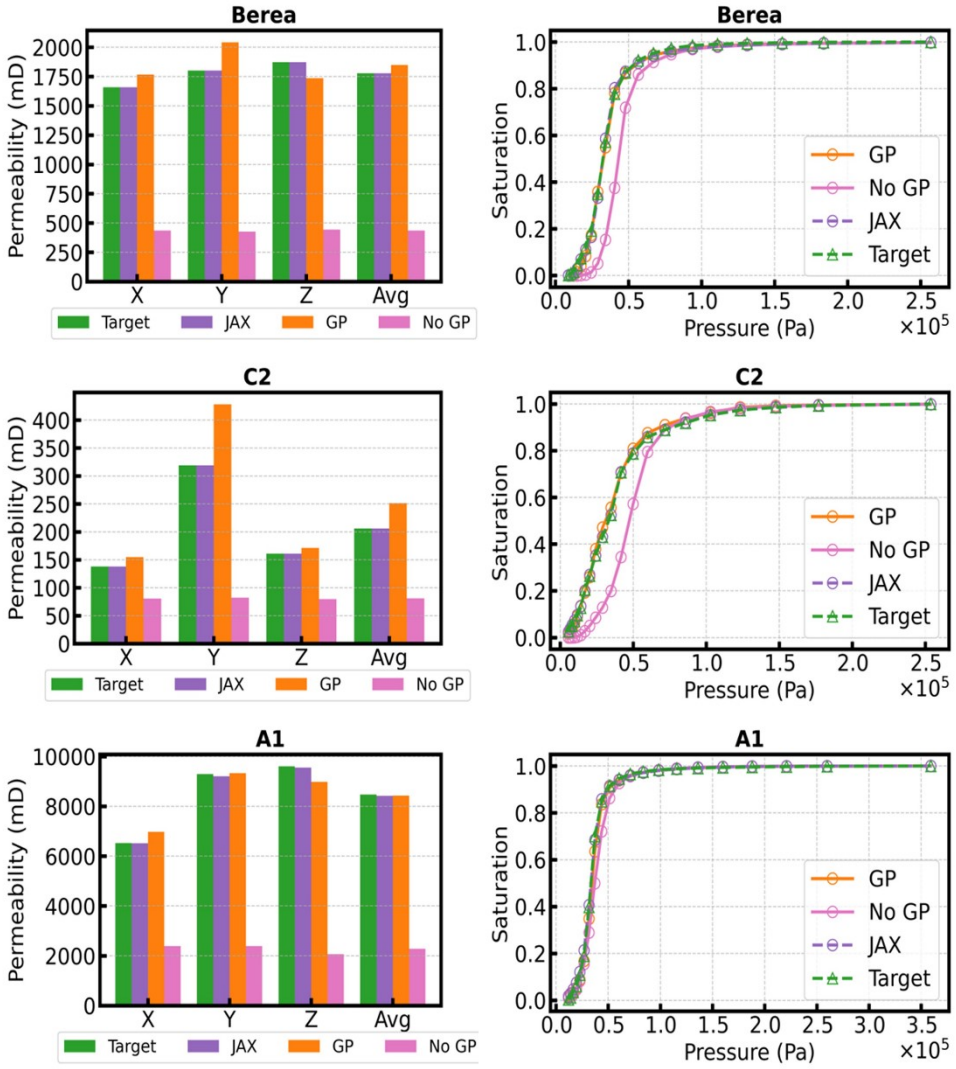

Figure S9 The resulting saturation and permeability before (pink) and after (orange) gaussian process was used to learn spatial correlation of optimized network.

Effect of Network Size on Gradient

Figure S10 shows the effect of network size on the gradient distribution. Notice how increasing the size of the network ever so slightly from 5^3 to 7^3 to 9^3 can have a significant effect on the gradient distribution by shifting its size to the left. As the network gets larger the impact of changing the diameter of just one pore becomes smaller and smaller. Knowing this was important when selecting a learning rate, in that the learning rate chosen should be informed in some way by the size of the network.

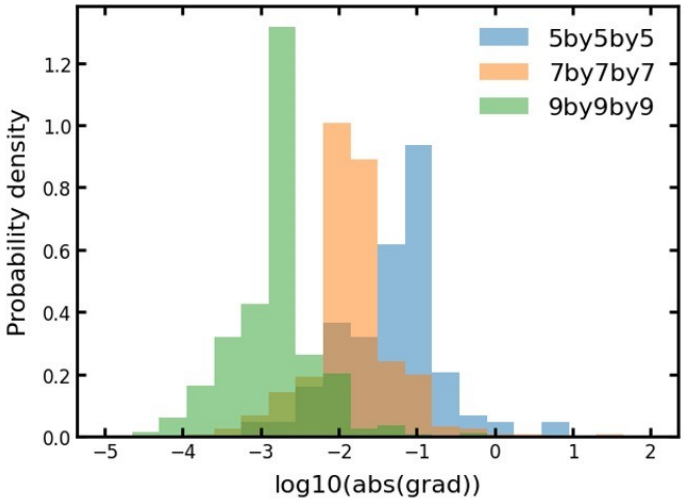

Figure S10 The effect of network size on the log of the gradient

Smoothing Factor

Figure S11 shows the effect of changing the smoothing factor on the final loss for the Berea sandstone sample. You can see that significant variations (approximately ±25%) in the smoothing factor cause little fluctuations in the final loss except at small smoothing factors. As the smoothing factor becomes smaller, gradient estimation becomes more difficult and hence optimization. The smoothing factor was chosen so that it could model the same physical behavior we see without the smoothing factor (see Figure S12) while also making gradient estimation possible. Figure S12 shows what happens after removing the smoothing factor. See how the PNM without smoothing factor (red) still matches the target data.

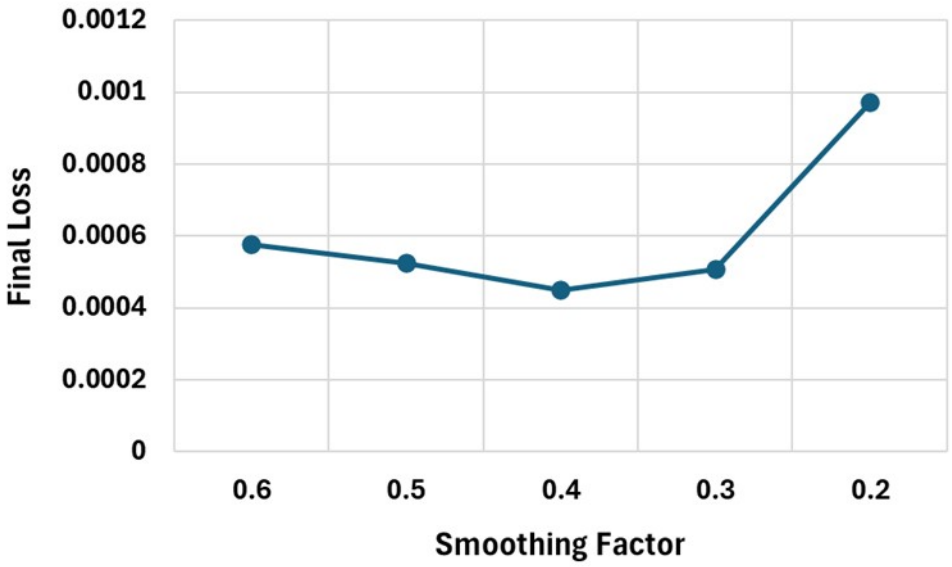

Figure S11 Sensitivity of smoothing factor on final loss for Berea sample

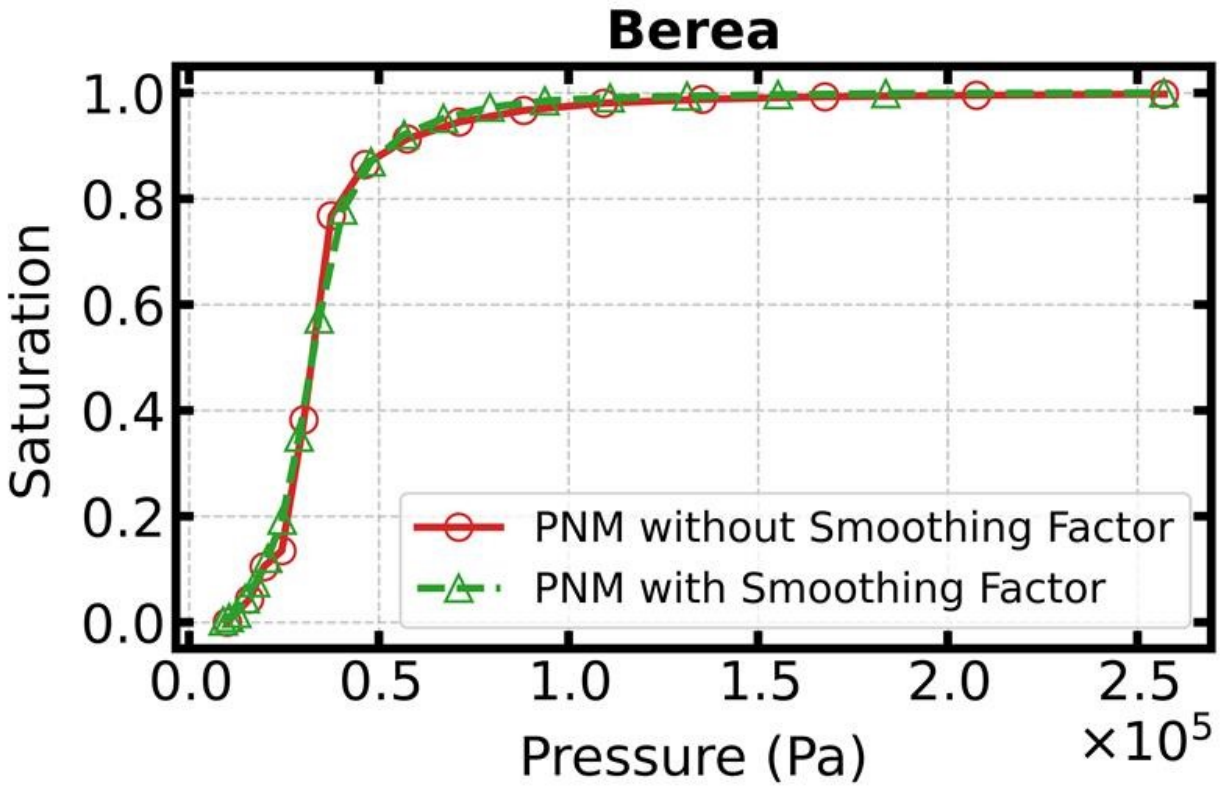

Figure S12 Resulting porosimetry curve after optimization and removing smoothing factor

Effect of Initial Guess

Gradient descent optimization is largely affected by the initial guess. Here, we explore the effect that changing the initial guess has on the final solution and resulting pore size distribution. The thinking is that if we use a "better" initial guess then we can obtain a closer match to the actual pore size distribution. Figure S13 shows the effect of changing the initial guess for throat aspect ratio and the effect this has on the final fitted network's pore size distribution. Plot (a) shows the original initial guess for throat aspect ratio which is normally distributed with an average of 0.4 and standard deviation of 0.05. Plot (b) shows the new or "better" initial guess as a Weibull distribution with shape of 5 and scale of 1. Plots (c) and (d) show the resulting pore size distributions after fitting using original and new guesses respectively. It is apparent that the shape of the final pore size distribution is largely controlled by the initial guess.

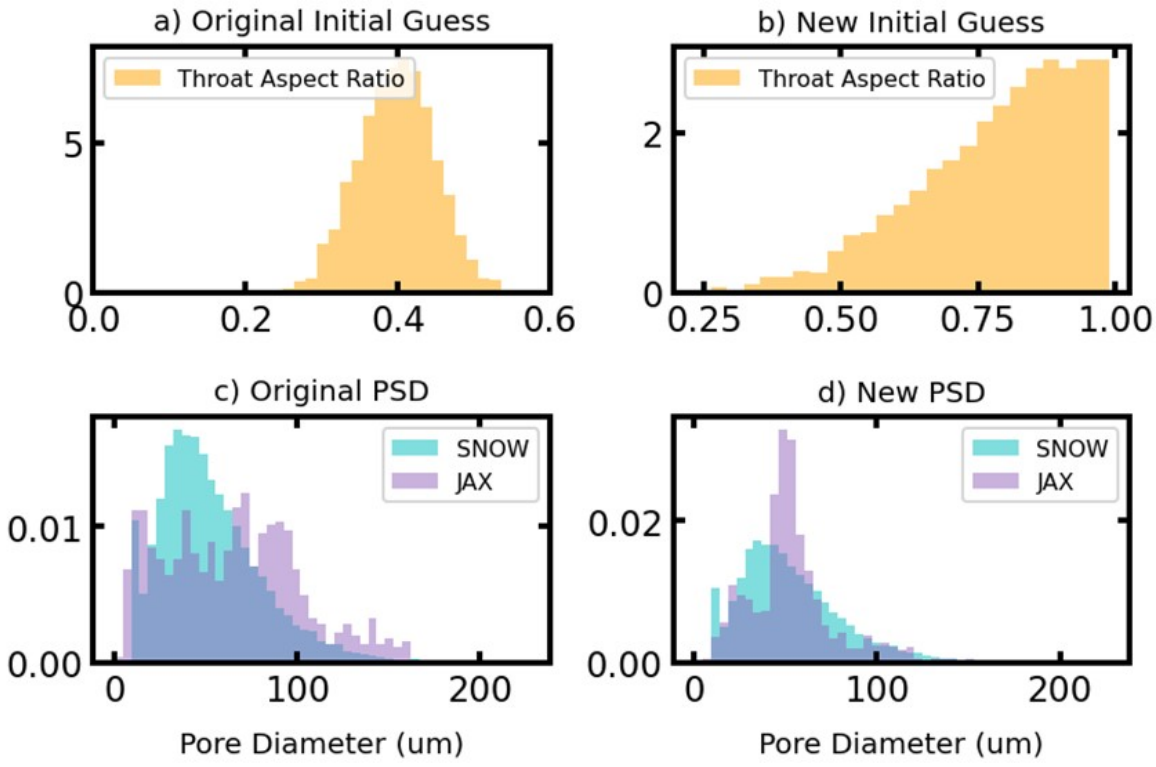

Figure S13 The effect initial guess can have on the resulting pore size distribution. Plots a) and b) are the original and new initial guesses respectively while plots c) and d) are the resulting pore size distributions from original and new initial guesses.

The initial guess can also effect the efficiency or route to the final solution. Figure S14 shows the loss upon each iteration using both the original (black) and new (red) initial guess. The final loss using the new initial guess reaches a value of 5.9×10^{-5}. This is roughly an order of magnitude better from the original. The sharp decline of the new initial guess indicates rapid training. Therefore, it is clear that the initial guess plays a significant role on both speed and convergence.

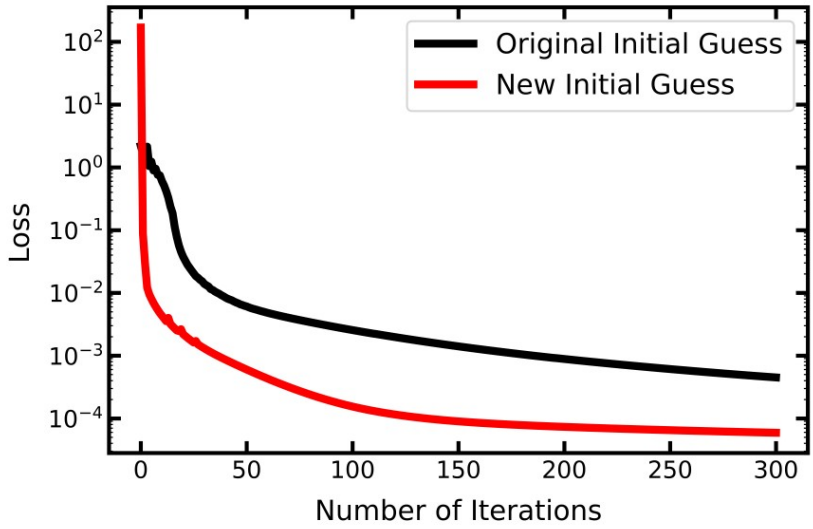

Figure S14 Loss as a function of number of iterations for original and new initial guess

Gaussian Process

Figure S15 illustrates better the process of using a small network to train a GP model for predicting pore sizes. First, both networks are scaled in all directions between 0 and 1. Next, a GP model is trained on the locations and sizes of the fitted network (blue pores). That GP model is used to predict the pore sizes at the locations of the enlarged network (red pores). Now, a second GP model is trained on the locations and throat sizes of the fitted network. Upon first attempt, we tried training this GP on throat coordinate data (i.e. the coordinate halfway between connected pores). However, this did not work because as we see from Figure S15 point A is closer to point B than it is to point C. We would prefer the throat at point A to match the throat size at point C because they point in the same direction. For anisotropic data, it is not merely the throat coordinate that matters but also the direction of the throat. By training the GP model to the coordinates of both connected pores, the model was able to learn not only the spatial relationship of throat sizes but also the anisotropic meaning of throat sizes.

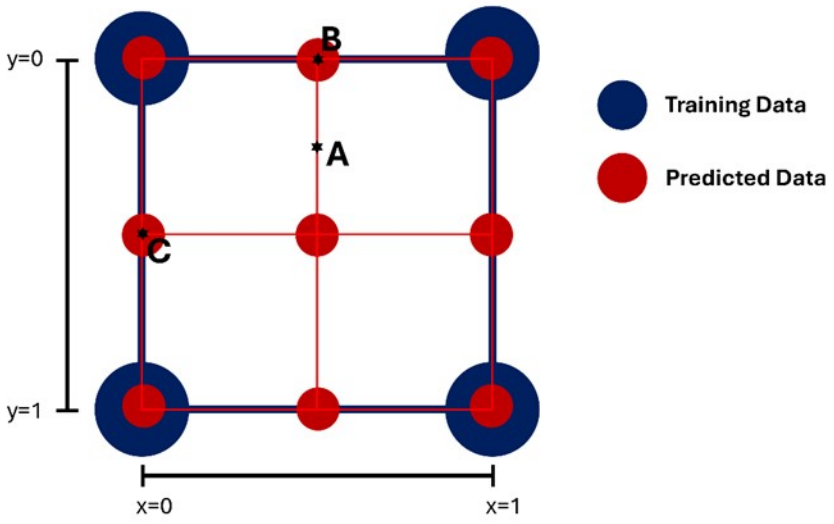

Figure S15 An illustration of how Gaussian Process is used to scale a simple 2 by 2 network to a 3 by 3 network. The locations of the fitted network used as training data is shown as blue dots while the locations of the predicted data is shown using red dots.

Code Snippet

Below is a brief `Python` code snippet written to help explain how gradient descent is performed using `JAX` and `Diffrax` packages. First, the gradient of the loss function is taken using `jax.grad()` which writes a computational graph to memory and returns a function, `grad_f`, that when called, computes the gradient using reverse-mode automatic differentiation. Then, a function, `dydt`, is defined that returns the negative of the gradient calculated by `grad_f` for any input. Finally, `dfx.diffeqsolve` is used as the main entry point to numerical integration as a way to perform gradient descent. Although other numerical integration schemes were tried, the Euler method was used because it is the simplest method that offers good control of a fixed learning rate. The span of integration `t0` and `t1` controls the number of integration steps to take while dt is the learning rate. Prior to integration, an initial set of weights is defined, here as random numbers spanning from 0 to 1, but in practice, an informed initial guess from the bundle of tubes model was used.

```
import diffrax as dfx
import jax
from functions import loss

f = loss()
grad_f = jax.grad(f)
```

```
def dydt(t, y):
    return -grad_f(y)

dt = 1
t0, t1 = (0, 10)
solver = dfx.Euler()
term = dfx.ODETerm(dydt)
w0 = jax.random.uniform(key, shape=(Np+Nt,))
solution = dfx.diffeqsolve(term, solver, t0=t0, t1=t1, dt0=dt, y0=w0)
w = solution.ys[-1]
```

References

[1] H. Dong, M.J. Blunt, Pore-network extraction from micro-computerized-tomography images, Phys. Rev. E. 80 (2009) 036307. https://doi.org/10.1103/PHYSREVE.80.036307/FIGURES/18/MEDIUM.

[2] P.L. Churchel, P.B. French, J.C. Shaw, L.L. Schramm, Rock properties of Berea sandstone, Baker dolomite, and Indiana limestone, (1991) 431–446. https://doi.org/10.2118/21044-MS.