ElRock-Net: Assessing the Utility of Machine Learning to Initialize 3D Electric Potential Simulations

Bernard C. Chang[1,*], *Javier* E.Santos[2], *Rodolfo* Victor[3], and *Maša* Prodanović[1]

[1]Hildebrand Department of Petroleum and Geosystems Engineering, The University of Texas at Austin, Austin, TX 78722, USA
[2]Center for Nonlinear Studies and Computational Earth Science Group, Earth and Environmental Sciences Division, Los Alamos National Laboratory, USA
[3]Petrobras, Rio de Janeiro, Brazil

Abstract. Advancements in imaging technology are enabling accurate simulations of transport properties through the pore space of an imaged rock sample, albeit at a computational cost. Meanwhile, machine learning has emerged as an alternate tool for modelling transport properties that, once trained, use a fraction of the computational resources that traditional simulations require. However, machine learning models often fail to strictly enforce physical constraints of the system, leading to solutions that are less accurate than that of traditional solvers. Here we propose a novel hybrid workflow that combines machine learning and conventional simulation methods to combine their speed and accuracy. The workflow begins with a 3D, binary image of a sample. A trained convolutional neural network learns spatial relationships between a porous medium geometry and predicts the electric potential field through the medium. A validated finite difference solver uses the predicted field as input and fine-tunes it to obtain a deterministic result. The proposed workflow provides reductions in computational cost without sacrificing solution accuracy on unseen samples despite not having a fully trained model.

1 Introduction

Quantifying electrical properties in porous media is an important method for characterizing and monitoring subsurface systems. It has applications in hydrocarbon reservoirs, carbon capture and storage [1], hydrogeology [2], and mineral exploration [3]. The electrical response of these systems aids in inferring the composition of the material and its phase distributions.

For example, in well-log interpretation of hydrocarbon reservoirs, the resistivity log is a particularly useful indicator of *in situ* fluid saturation estimations because of its sensitivity to the varying phase conductivities [4]. Conventional interpretation models generally rely on the assumption that saline formation water is the only conductive phase — that hydrocarbon and the solid matrix do not readily allow electric current to flow. Correlating the resistivity to fluid saturations are crucial to accurately assessing the viability of a reservoir.

Fundamentally, electrical resistivity is an intrinsic property of a material that quantifies how strongly it conducts electric current. The overall resistivity response is subject to depositional and diagenetic processes that dictate the structure of the pore space and the phases that fill the pore space (assuming that the grains are not conductive). More specifically, resistivity measurements are most impacted by the electric tortuosity of the connected conductive region and variations in cross-sectional area of the conducting path [5]. The physical processes that result in these geometric heterogeneities can exist on multiple length scales, extending down to the nano- and micron-scales.

Small scale (i.e., nanometer-to-micron-scale) heterogeneities can have a significant impact on the total electrical response of the system, arguably more so than for fluid flow [6]. Physical interpretations of larger length-scale correlation components, such as the Winsauer tortuosity factor [7] or the cementation exponent in Archie's equations [8], often attempt to account for these pore-scale heterogeneities [9]. However, quantifying electrical behavior on the pore scale is critical to making inferences on larger domains.

Recent improvements in imaging technology have led to the observation of many porous media processes. Digital, 3D image data acquired through high-resolution imaging, such as computed tomography (CT), micro-computed tomography (μCT), and focused ion beam scanning electron microscopy (FIB-SEM) reveal details of porous media structures on various length scales. Analyzing rock properties based on these digital images are collectively known as *digital rock physics.* Assuming a minimum representative elementary volume (REV) and sufficient resolution, direct simulations on these image samples provide an accurate picture of physical processes on the pore scale and the ability to upscale to larger length scales. Further, breakthrough innovations in storage, hardware, and software have provided the capability to process data, run simulations, and analyze results on increasingly large images. Digital Rocks Portal (DRP) [10] is highlighted here as an open repository that contains a wide range of porous media samples. It provides simulation and machine learning training data used in this work and many others [11, 12, 13].

* Corresponding author: bcchang@utexas.edu

Although imaging advancements have contributed to improved understanding of pore-scale processes, the computational resources required to handle the enlarged image sizes are likewise increasing. Conservatively, computational demands scale to the cube of the side length of the discretized domain. This becomes problematic when performing simulations on structurally heterogeneous samples, such as in carbonate rocks. Oftentimes, achieving a minimum REV (if possible) necessitates increasing the size of the image sample beyond a system's hardware capabilities. This renders performing direct simulations on REVs of complex systems exceedingly difficult, typically resulting in long compute times on supercomputers.

Meanwhile, deep learning has emerged as an alternate tool for recognizing patterns from observational data. These algorithms have widespread applications in subsurface systems, including reservoir characterization, production optimization, and lithology interpretation and rock classification [14, 15, 16]. In the context of digital rocks physics, machine learning workflows have also been successful modeling some transport properties such as permeability and flow velocity [13, 17, 18]. However, these workflows often fail to strictly enforce the physical constraints of the system, leading to solutions that are less accurate than that of traditional solvers. Training accurate neural network models also require large quantities of data and are not easily generalized to complex, heterogeneous reservoir rocks.

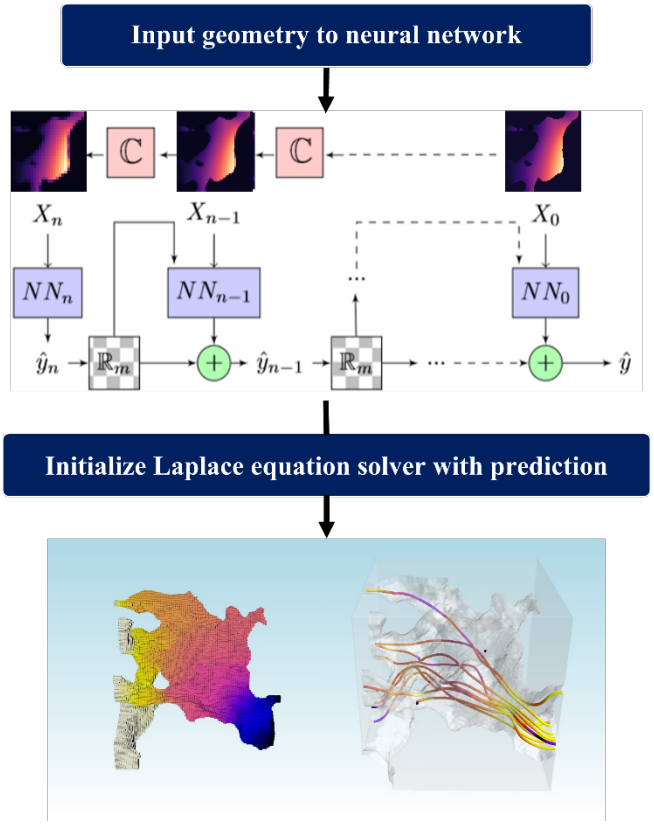

Fig. 1. Conceptual figure of the proposed framework. A trained neural network predicts the electric potential distribution through the pore space. A validated solver is then initialized with the prediction and obtains the potential and current fields through the medium.

Here we propose a framework similar to that of Wang et. al [19] for combining machine learning workflows and direct simulation methods (Fig. 1). A trained neural network extracts spatial relationships between a porous medium geometry and predicts the electric potential field through the medium. The predicted field is used as the initial condition for a validated Laplace equation solver and fine-tunes it to obtain a deterministic result. This framework combines the benefits of machine learning and direct simulation to reach a solution quickly without sacrificing accuracy. We also provide some guidelines and metrics for assessing the model predictions on unseen samples.

2 Methods and Materials

2.1 Dataset

Porous media in the subsurface is comprised of a wide range of lithologies. Because most traditional resistivity correlations were developed for clastic, sedimentary rocks, we direct assessment of the proposed framework to predictions on such lithologies.

To generalize our findings, we sample from open data benchmark on Digital Rocks Portal (DRP) [20, 21]. The catalog consists of 217 binary images spanning over 50 rock types and lithologies and were processed from 125 projects currently hosted on DRP. The catalog also includes results from flow and electrical simulations in addition to geometric and structural characterizations of the pore space. Subvolumes of 256^3 and 480^3 were extracted from the center of the original samples and oriented such that the direction of flow is parallel to the z-axis. The images were then preprocessed to ensure percolation and to eliminate all but the largest connected component of the pore space.

As part of this study, we sample four numerically-dilated sphere packs and four sandstone samples. The original, discretized sphere pack was reconstructed using the locations of the Finney packing of spheres [22]. Subsequent morphological operations (erosion and dilation) were performed to vary the porosity from 0.10 – 0.42. The five sandstone samples selected for this study originated from Neumann et al. on DRP [23] and include Parker, Leopard, Kirby, and Bentheimer sandstones with porosity values summarized in Table 1. We assume that the sandstone samples are clay-free. For both sandstones and sphere packs, we assume the solid matrix is non-conductive and the void space is fully saturated with conductive brine.

Direct simulations often perform calculations on volume sizes exceeding 1000^3 voxels. However, memory constraints on the available graphical processing units (GPUs) severely limited the size of viable training data. The 200^3 subvolumes are therefore used for training data.

Table 1. Nomenclature, porosity, and simulated formation factor for sphere pack and sandstone samples.

Sample	Description	ϕ	F
SP0	Eroded SP	0.418	3.57 ± 2.21E-04
SP2	Dilated SP	0.293	6.48 ± 3.91E-04
SP4	Dilated SP	0.182	15.38 ± 4.42E-02
SP6	Dilated SP	0.099	98.71 ± 9.64E-01
SS4	Parker SS	0.268	11.00 ± 6.33E-02
SS8	Leopard SS	0.273	11.05 ± 2.22E-02
SS9	Kirby SS	0.211	21.91 ± 1.23E-1
SS11	Bentheimer SS	0.088	174.95 ± 4.12E0

2.2 Electrical Simulations

Traditional methods of relating electrical resistivity to fluid saturation (S_w) in clay-free reservoirs are commonly based on Archie's empirical equations [8].

$$R_t = R_w \frac{a}{\phi^m S_w^n}, \quad (1)$$

where R_t and R_w are the total resistivity and resistivity of formation water, a is the Winsauer tortuosity factor [7], ϕ is porosity, S_w is the brine saturation, and m and n are experimentally obtained cementation and saturation exponents.

The formation resistivity factor, F, is derived from Archie's equation and relates the sample's resistivity to its ϕ and a:

$$F = \frac{\sigma_w}{\sigma_o} = \frac{a}{\phi^m}, \quad (2)$$

where σ_w and σ_o are the conductivities of the brine and the fully brine-saturated formation, respectively. Physically, F can be considered a normalized conductivity that measures the influence of the pore structure on the conductivity of the sample.

We assume a reliable segmentation of a porous medium into solid and void space and use it to solve the generalized Laplace equation for the electric potential distribution through the medium,

$$\nabla \cdot (\sigma \nabla \varphi) = 0, \quad (3)$$

where φ is the scalar electric potential field.

The volume-averaged currents in the three coordinate directions are found using the solution to the finite difference representation of the Laplace equation. The total electric current, I, through the rock is defined as

$$I = \oint \sigma \nabla \varphi \cdot \hat{n} \, dA. \quad (4)$$

The electric current is then used to calculate the macroscopic conductivity of the rock

$$\sigma_{\text{rock}} = \frac{LI}{A\Delta V}, \quad (5)$$

where L is the sample length, A is the area of a slice orthogonal to the flow of electric current, and ΔV is the difference in macroscopic electric potential applied at the inlet and outlet. Finally, we use σ_{rock} in equation 2 to calculate F.

Every sample chosen for this dataset consists of one non-conductive grain phase and one conductive fluid phase. We, therefore, assign the conductivities of the grain and fluid phases to be 0 and 1, respectively. The potential value at the inlet and outlet slices are fixed and the electric potential is initialized as a linear gradient through the pore space. The normal component of the electric current density is set to zero at pore-grain boundaries.

We perform electrical simulations using Digital Rock Suite (DRS) [24]. To validate the code, we numerically calculated the electrical conductivities of 14 fluid configurations in a periodic, random, close-packing of spheres of uniform radius. The datasets, authored by McClure et al. [25], used the lattice Boltzmann method to determine equilibrium fluid configurations at different initial fluid saturations. We compare the DRS solution to that of DC3D.f, a similar, open-source code from the National Institute of Standards and Technology [26], The reported porosity of the sphere pack is 0.369 and has a computational domain size of 900^3 voxels. The code was validated using samples containing multiple fluids occupying the pore space. However, the scope of this study is limited to samples with a single occupying fluid. While electrical predictions in partially-saturated domains are desirable, accurate single-phase electrical property prediction has remained a challenge and must be better understood first.

As Fig. 2 shows, both solvers showed good agreement in their results. Generally speaking, as the wetting phase saturation decreases, the bulk wetting phase becomes disconnected and a negative deviation from Archie's law would be observed [27]. However, in this saturation regime, we would expect the resistivity to follow a linear trend in log-log space (Archie's law). The power law fit further supports validation of DRS as the fitting parameters, a and n, matched the expected values for a packing of spheres.

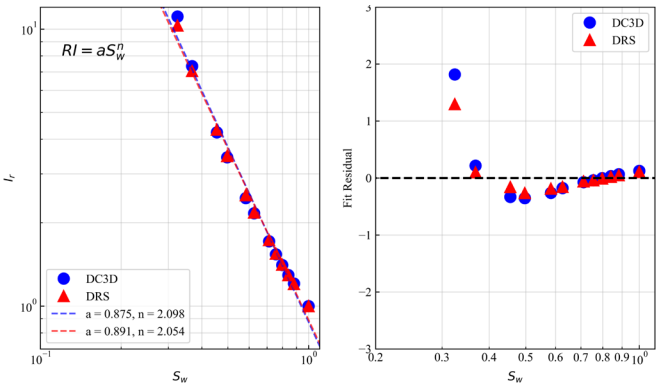

Fig. 2. Comparison of resistivity calculation results on a 900^3-voxel packing of spheres at various brine saturations. Both results follow a linear trend as described by Archie's equation; however, Digital Rock Suite performed better at smaller saturations. Each simulation uses 64 processors on the Stampede2 cluster in the Texas Advanced Computing Center (TACC).

In terms of run times, DRS was generally faster than DC3D.f and saw considerable improvement at smaller water saturations.

2.3 Machine Learning Model

Direct numerical simulations on images can accurately describe physical phenomena through complex geometries. However, fully deterministic calculations are often computationally demanding, particularly for large domains.

Another promising avenue is using data-driven models to estimate properties from 3D images. Machine learning (ML) has emerged as an effective tool for finding complex relationships in structured data. Making predictions with ML models is computationally inexpensive compared to traditional simulations.

For any neural network, the training phase is the most resource-intensive task. Typically, models are trained on GPUs because their hardware architecture enables them to handle more parallel processes than their CPU counterparts. However, most commercially-available GPUs host memory ranging from 2 – 24 GB, resulting in a constraint on the size of viable training data. This presents a bottleneck for network training in that training on images larger than the required minimum REV is often difficult or impossible with current hardware resources.

To circumvent this constraint, previous networks often employed patch-based learning [13, 19] in which the image is divided into small subvolumes. This was advantageous in that the network could train and predict on any arbitrarily-sized volume. However, patch-based learning assumes stationarity of the flow statistic and, therefore, depends on the statistical homogeneity of the pore space.

For the proposed framework, we adopt the MultiScale Network for hierarchical regression (MS-Net) [18], which is the first convolutional neural network (CNN) that successfully predicted permeability of imaged samples that included heterogeneities such as fractures and vugs. The network architecture was designed to capture morphological information across entire 3D volumes by using a hierarchical set of models, each responsible for learning increasingly coarse realizations of the input data. Santos et al. [18] originally used this network to predict the fluid flow velocity field and absolute permeability from a 3D binary image. Preliminary results indicate that the same network architecture can be adopted to predict the electric potential field.

The coarsening process involves averaging each group of 8 neighboring pixels (in 3D) to obtain a new image that is 8 times smaller than the original (Fig. 3). In this fashion, the model can see the image in its entirety, allowing it to recognize heterogeneity in the pore space. The hierarchical network architecture consists of a system of smaller neural networks. The individual neural networks train and predict on the same input domain, but at different resolutions. The features that each network learns at its associated scale are used to inform the next refined scale. As such, the coarsest image and each prediction is subsequently passed to increasingly fine resolution scales until the original image resolution is recovered.

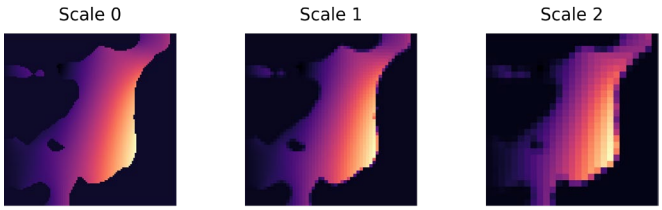

Fig. 3. Example coarsening of the ground-truth image, in this case the detrended electric potential. Average pooling is performed on each successive layer to reduce the size of the preceding image by 2^3. A system of networks are trained on each scale, with coarser images informing finer images.

The ground truth data for training comes from the output of the numerical solver, DRS. DRS initializes the electric potential through each slice of the image using a linear gradient between the inlet and outlet slice values (2 and 1, respectively). We would expect the final solution to present similarly, except with deviations due to the configuration of the input geometry and the applied boundary conditions. The underlying gradient can be removed (detrended) to aid in improving machine learning model performance by subtracting the initial linear gradient and only training with residual values of the electric potential. From a machine learning perspective, the underlying trend can distort the relationship between the explanatory and response variables, hindering a model's ability to learn. By detrending the data, we eliminate the need for the model to learn the underlying trend. This highlights anomalous regions where the potential value deviates from the underlying linear gradient initialization. The model then predicts these residual values to which the linear trend can then be re-added (retrended) to recover the original potential distribution.

In this framework, we train the neural network using only the binary image as input.

2.4 Assessment Metrics

Neural networks have difficulty generalizing to features they have not seen before [28]. Common approaches for remedying this are by either increasing the diversity of the training data or by developing more robust relationships by including more information about the data (e.g. geometric features, physics-informed learning, etc.). Additionally, predictions of transport fields do not necessarily obey the physical laws that govern such processes. Whether providing the network with more training data or employing the proposed framework, traditional simulations are still a necessary component to scientific research.

The assessment of the proposed framework leverages a phenomenon known as overfitting. Overfitting occurs when the network depends on specific details of the training set — in essence, the model fits to the noise of the training data. Theoretically, a completely overfit model will predict the training data with 100% accuracy. Overfitting is generally undesirable because it only performs well on the training data while making relatively poor predictions on unseen data.

The scope of this study is not to produce a generalizable model, but to assess the degree to which a model should be trained to see benefits in simulation run time. We, therefore, perform a sensitivity analysis in which models are

intentionally overfitted to the provided training data. These models are intended as the "best case" scenario in which the error in prediction for the supplied training sample is minimal. They are not intended to be generalized models that can be applied to predict on other domains. Each model is provided one training sample, is trained for approximately 30,000 epochs, and predicts on the same training sample. Intermediate model states are saved throughout training to track the model's progression. The intermediate and final predictions are supplied to DRS and the run times to simulation convergence are compared.

We explore nine metrics to assess the training-data prediction accuracy of the model throughout training. We calculated each metric directly on the predicted detrended field (model output) and on the manually re-trended field (input field supplied to DRS). These metrics are evaluated against the simulation run times to gain insights into the components important to the success of the proposed framework. When the ground-truth is known, such as in training and validation data for supervised learning, the most important metrics can be included as additional penalization criteria. When generalizing to unseen data, metrics that do not require a known solution can also be calculated to evaluate the quality of the prediction.

2.4.1 Norms

Neural network training is an optimization problem in which a loss function is minimized. Because loss functions are measures of prediction error, the vector norms of the difference between the labeled data and the predictions are commonly employed in loss functions. MS-Net uses the Mean Squared Error (MSE) loss function between the ground-truth (Y_i) and predicted ($\widehat{Y}_i$) fields,

$$MSE = \frac{1}{n}\sum_{i}^{n}\left(Y_i - \widehat{Y}_i\right)^2. \tag{5}$$

The use of MSE causes MS-Net to preferentially penalize larger magnitudes of $|Y_i - \widehat{Y}_i|$. In other words, the network pays more attention to model parameters that cause predictions to diverge further from the true field. For comparison, we also calculate the L_2 distance between Y_i and $\widehat{Y}_i$,

$$L_2 = \sqrt{\sum_{i}^{n}(Y_i - \widehat{Y}_i)^2}. \tag{6}$$

Alternatively, one may choose to use the mean absolute error (MAE), based on the L_1 distance, as the loss function. This loss function is less sensitive to outliers than MSE because the values are not squared. We, therefore, calculate the L_1 distance to the ground truth,

$$L_1 = \sum_{i}^{n}\left|Y_i - \widehat{Y}_i\right|. \tag{7}$$

2.4.2 Peak Signal-to-Noise Ratio

Peak signal-to-noise ratio (PSNR) is commonly used to quantify the "closeness" of an image to being an exact copy of another such as when filtering images. It is a voxel-by-voxel calculation, derived from MSE, that determines imperceptible differences between two images and is defined as

$$PSNR = 20\log_{10}\frac{\mathrm{MAX}_I}{MSE\left(Y_i, \widehat{Y}_i\right)}, \tag{8}$$

where MAX_I is the maximum possible voxel value of the 3D image. The PSNR value decreases as the MSE increases, implying large differences between the input images.

2.4.3 Histograms

Analyzing the histogram of $\widehat{Y}_i$ reveals important information about the training progression and areas that the model has difficulty training. Comparing the distributions of $\widehat{Y}_i$ and Y_i can also help to confirm error metrics.

Detrending a field helps the network train by eliminating the need to learn the variance due to an underlying trend. As previously stated, detrending the potential field highlights anomalous regions in which the potential deviates from the underlying linear trend. These residuals are approximately normally distributed about a mean close to zero. Here we track the mean of $\widehat{Y}_i$, which should approach the expected value of Y_i as training progresses.

2.4.4 D-2 Score

The result of both patch-based networks and multiscale networks are predictions made up of perceptible blocks. These are unphysical artifacts of the network algorithm due to the convolution operation. When analyzing these types of images for mass conservation or slice-wise flux continuity, the boundaries between blocks manifest as large jumps in otherwise continuous behavior. If unresolved, these can negatively impact calculations of properties, such as *F* or permeability. MS-Net appears to refine these blocks as training continues; however, they are still seen even after the model is severely overfit.

We quantify the smoothness of the image using a value derived from taking the Laplacian of the image — the D-2 score. First, the Laplacian of the image is taken,

$$\nabla \cdot \nabla \hat{Y} = \frac{\partial^2 \hat{Y}}{\partial x^2} + \frac{\partial^2 \hat{Y}}{\partial y^2} + \frac{\partial^2 \hat{Y}}{\partial z^2}. \tag{10}$$

The Laplacian of the image is calculated by convolving the predicted image with a 27-point Laplacian stencil [29]. We sum the Laplacian over all pore voxels to calculate the D-2 score. When compared to the D-2 score of the labeled training image, intermediate predictions with more pronounced boundary effects will have a larger D-2 score. The D-2 score

begins to decrease throughout the training progress as the artificial boundary effects are refined.

2.4.5 Formation Factor

Finally, we calculate the formation factor of $\hat{Y}$. The average current flux in the direction of flow is used for calculating F, so evaluating this property and its associated uncertainty in the predicted image is an indication of flux continuity through the image.

The uncertainty (standard deviation) in the current flux across every slice orthogonal to the flow direction is used as a convergence criterion in DRS simulations. Therefore, calculating the uncertainty in F in the prediction provides a direct comparison to DRS convergence.

Here, we convolve the prediction with a finite difference kernel to calculate the component of the current density parallel to the direction of flow. We then implement a masking scheme to respect the pore-grain boundary conditions outlined in the DRS algorithm. Like DRS, this component of current density is summed over each orthogonal-to-flow slice to find the current flux. Finally, the average and standard deviation of these fluxes are used to evaluate the macroscopic conductivity and formation factor.

3 Results and Discussion

MS-Net is trained using single 3D, binary images of 200^3 sphere packs and clean sandstones. The models' accuracy of predicting the potential field of the training samples is evaluated. The accuracy metrics are compared against the predicted formation factor. Finally, we investigate possible relationships between these metrics and accelerating simulation results by using the predicted field as input.

Training was performed using Nvidia 16 GB V100 GPUs on Longhorn — a Texas Advanced Computing Center (TACC) resource. Each model was trained for ~30,000 epochs, requiring ~16 hours of training time. This illustrates the cost balance between machine learning training and simulation that must be considered. Though models can require long training times (order of hours to days), making predictions are effectively instantaneous when compared to simulations. For example, in this study, testing is performed on a local desktop machine with 64 GB RAM and required only a few seconds to make a prediction. Using this framework with generalized models can lead to reaching accurate solutions far more quickly for a limitless number of unseen samples. This framework can also be extended to more complicated solvers that generally require longer run times for more immediate run-time reduction benefits.

3.1 Evaluation of Training Performance

The training progress of machine learning models are commonly tracked using the models' loss functions. Common examples of loss functions are MAE (L_1 loss) and the MSE (L_2 loss).

The L_1 norm (Fig. 4) of the detrended field monotonically decreases as training progresses, an indication that intermediate predictions are approaching the labeled data.

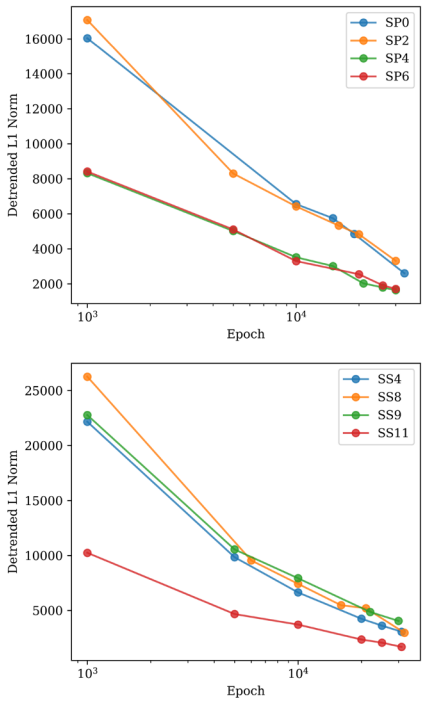

Fig. 4. L_1 distance to ground truth of various models saved through training. The L_1 distance, or MAE, is a common training performance metric. For both the sphere pack samples (left) and sandstone samples (right), we see that the MAE monotonically decreases as training progresses.

The L_2 loss (Fig. 5) shows similar behavior. The L_1 and L_2 losses follow the same general curve for a particular sample. SP2 and SS4 maintain the largest error throughout training among their respective sphere packs and sandstone

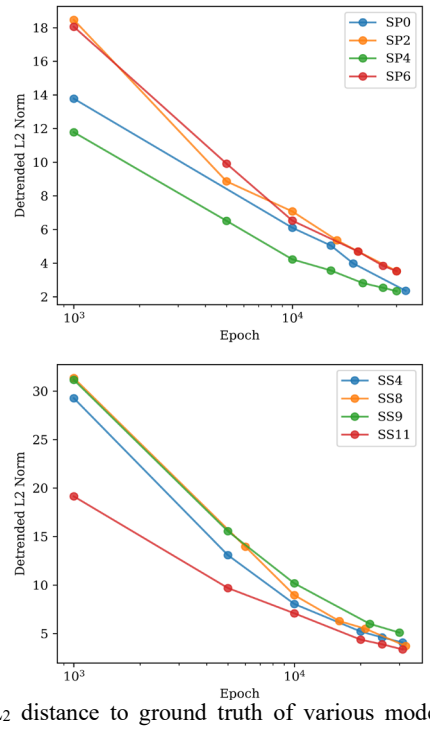

Fig. 5. L_2 distance to ground truth of various models saved through training. Similar to the MAE (Fig. 4), the MSE monotonically decreases as training progresses for both the sphere pack samples (left) and sandstone samples (right).

samples. SP6 and SS11 tend to have the smallest error during training, though this could be an artifact of their small porosity values.

As previously discussed, the MS-Net architecture trains convolutional neural networks on multiple resolution scales. Making predictions using this architecture involves a refinement operation, which results in blocky predictions. These blocks are artificial discontinuities within the pore space and will negatively impact calculations of macroscopic properties, such as F.

The Laplacian of the image (Fig. 6) highlights regions where rapid changes in intensity occur. Naturally, this occurs near the pore-grain boundary, where boundary conditions must be enforced. The potential within grains is assigned to be zero while the potential values in the pore space change gradually and with an approximately linear underlying trend. Therefore, the Laplacian should be close to zero within the pore and the grain space.

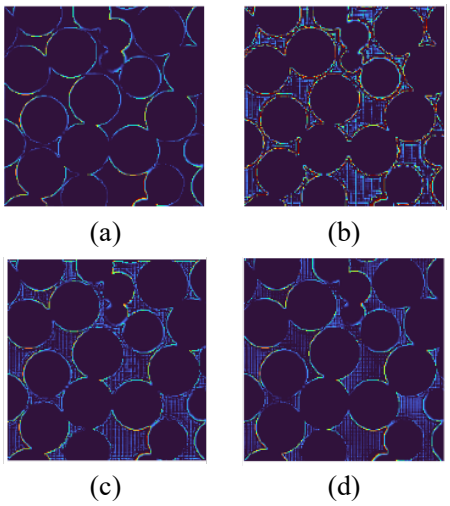

Fig. 6. Example slice (200 x 200) of the Laplacian transform of a) DRS output, Laplacian of the model predictions after b) 1000, c) 15000, d) 30000 training epochs. The Laplacian highlights sharp contrasts in voxel values (grain-pore boundaries and block artifacts). As training progresses, the boundary effects gradually disappear, which, when summed, results in decreasing D-2 scores approaching that of the simulation output.

Evaluating the Laplacian of the prediction captures the large change in image intensity near the pore-grain boundary and along the artificial discontinuities. As training progresses, MS-Net tends to refine these blocks. Nonetheless, they are still prevalent in the model predictions, even in highly overfit models. This highlights the need for more intelligent masking operations to refine the resolution of the prediction from the coarsest scale. One could also incorporate smoothing operations, such as median filtering, into model training.

3.2 Evaluation of Predicted Formation Factor

To compute the current density, we convolve the predicted electric potential field with a finite difference kernel and sum the result over each slice. The average of the slice-wise sum is used to calculate the predicted formation factor ($\hat{F}$).

$\hat{F}$ of the final model matches closely with the simulation results for both the sphere packs and the sandstone samples (Fig. 7). In general, the model underpredicts the value of F, particularly at smaller values of porosity. Likewise, the uncertainty in $\hat{F}$, calculated from the standard deviation in the sum of the slice-wise current flux, increases with decreasing porosity. These observations can be attributed to the potential value discontinuities at the boundaries of the block artifacts.

Fig. 7 also shows a clear relationship between the uncertainty of $\hat{F}$ and the simplicity of the pore system — the uncertainty in $\hat{F}$ of the sandstone samples is larger than that of the sphere packs. The uncertainty in $\hat{F}$ stems from local heterogeneities of the pore space. Structural details, such as tortuosity and constriction factor, nonuniformly affect the current flux and are not directly accounted for when evaluating F. Therefore, the uncertainty in $\hat{F}$ is generally small for more homogeneous samples like sphere packs and in samples with large porosity.

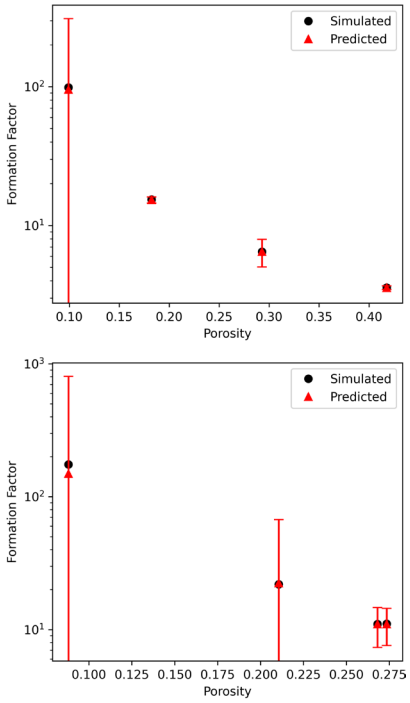

Fig. 7. Formation factor comparison between the simulated result (black circles) and the final model prediction (red triangles). The predicted values of formation factor are fairly accurate for both sphere packs (left) and sandstones (right) but tend to underpredict the ground truth. The uncertainty of the prediction is much larger than that of the simulation.

Although the MSE monotonically decreases with training epoch, the relative error in $\hat{F}$ does not follow the same trend (Fig. 8). Several samples show good early $\hat{F}$ and are only slightly improved by the final prediction, at the smallest MSE. Broadly speaking, this is indicative of the fact that F is a macroscopic quantity and does not account for the exact microscopic details of the pore space. The implications of this are that undertrained models are still serviceable as F estimators. Nevertheless, additional training generally still results in a decrease in the relative error of $\hat{F}$. This is apparent in the sandstone samples where later $\hat{F}$ are closer to the expected result than early $\hat{F}$.

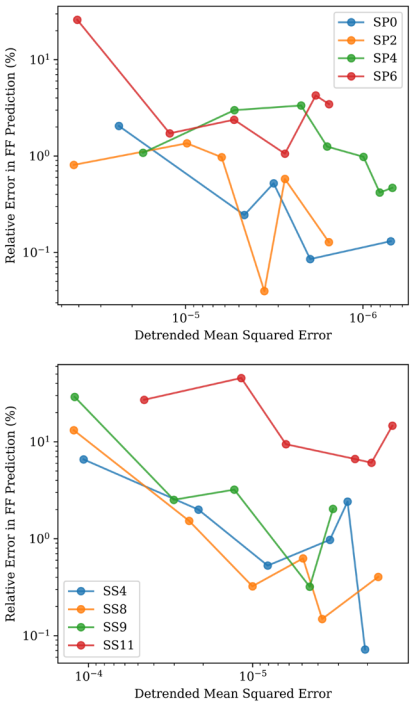

Fig. 8. Relative error in the formation factor prediction versus MSE (decreasing MSE to the right to reflect training progression) of the sphere pack samples (left) and the sandstone samples (right). Decreasing MSE does not necessarily indicate improved formation factor prediction.

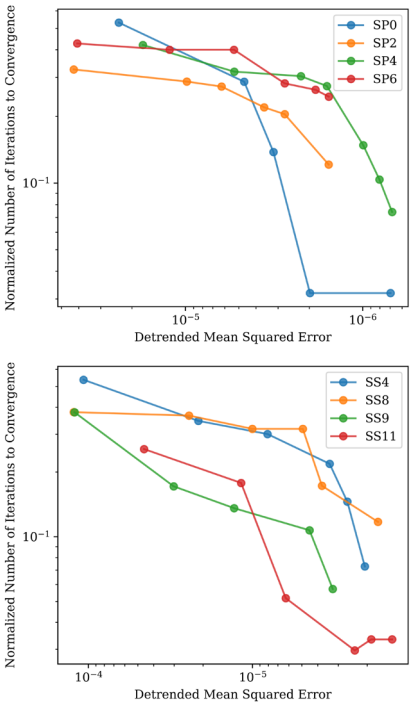

Fig. 9. Normalized number of iterations to convergence error versus MSE (decreasing MSE to the right). The framework improves the run time of both sphere packs (left) and sandstones (right) to less than 10% of the original.

3.3 Evaluation of Simulation Time Improvement

To assess the proposed framework, we save the re-trended predictions of intermediately trained models and feed them to DRS as initial conditions. We note the number of iterations that DRS requires to reach convergence and normalize it by the number of iterations required when assuming a linear potential gradient.

Typical simulations used for this study required between 800 – 2000 iterations to converge (more porous and well-connected samples generally required fewer iterations). The actual wall-clock run time for the simulations are largely hardware dependent. Because the images used in this study are relatively small, the total simulation run time only lasted

Here, we report the most illuminating metrics. We first investigate metrics associated with supervised learning — those that compare predictions with labeled data. We then look for relationships between unsupervised metrics and run time improvements to provide example diagnostics that can be included in future model training.

3.3.1 Supervised Metrics

Using $\hat{Y}$ as the initial condition for DRS results in a drastic improvement in the required run time to convergence (Fig. 9). In the best cases, the proposed framework saw over an order of magnitude improvement, reducing the number of iterations to convergence to less than 10% of that of the linear gradient assumption.

MSE is a decent predictor of the run time improvement, where the number of iterations to convergence decreases with decreasing MSE. However, the plateau in iterations at large MSE indicates that the model needs to achieve a threshold accuracy before seeing a significant improvement in run time. Nevertheless, even under-trained models with large MSE still sees over a 50% improvement in run time, implying that models do not necessarily need to make entirely accurate predictions to seeing benefits in run time. This has important implications for the use of the proposed framework, particularly when applying a trained model to unseen data. We also expect to see more significant improvements as the size and complexity of the image increases. For example, the 900^3 spherepack samples used for code validation required simulation times upwards of 10 hours for the smallest saturation points. This could feasibly be reduced to a run time of only a few minutes if supplied an extremely accurate prediction result. If one assumes that the simulation run time approximately scales to the cube of input image side length [13], we hypothesize that one could feasibly start seeing significant run time benefits when predicting on $\sim 500^3$ images (2+ hours of simulation time reduced to several minutes).

Because PSNR is derived from MSE, it exhibits similar predictive capability of run time improvement as MSE (Fig. 10). The curves also show slow improvement in run time at small values of PSNR. At a PSNR value of ~ 40, we begin to see drastic improvement in run time. Except for SP0 and SS11, we do not see the run time curves level off in this PSNR range, implying that further run time improvement is possible.

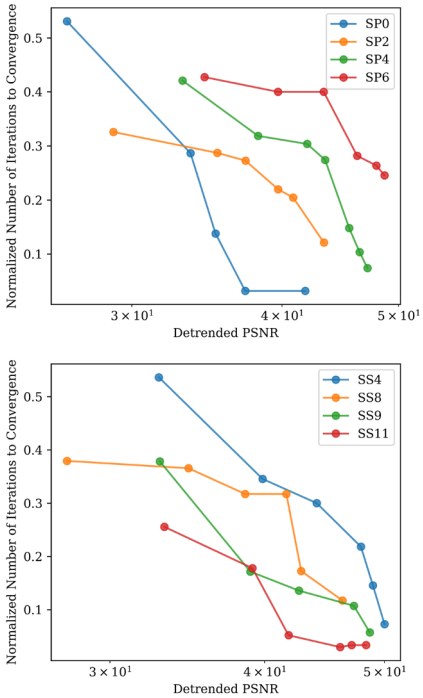

Fig. 10. Normalized number of iterations to convergence error versus PSNR. At PSNR of ~40, we begin to see large improvements in run times in both sphere packs (left) and sandstone (right) samples.

3.3.2 Unsupervised Metrics

Thus far, the discussed metrics for run time improvement have required labeled data against which the prediction is evaluated. When extending models to unseen samples, the test data does not yet have a labeled ground truth. Unsupervised learning is advantageous in this respect where the model learns from patterns in unlabeled data. Here, we discuss a few performance metrics that can be calculated on the prediction itself, without the need for labeled data. These metrics are suggestions for diagnostic tools that can be used to assess whether the solver will see a significant run time improvement by initializing using the model prediction.

We first explore the uncertainty in the slice-wise current flux — the solver's primary convergence criterion (Fig. 11). We compute the uncertainty using the method outlined for computing F, but stop short of calculating the macroscopic conductivity. The uncertainty in the current flux is a direct point of comparison to the convergence of DRS. We, therefore, expect a direct relationship between the decrease of the number of iterations to convergence and the decrease of uncertainty in current flux. Fig. 11 confirms this hypothesis, where we see three distinct regimes of run time improvement.

First, the improvement in run time is somewhat gradual at large current flux uncertainties. The run time speed-up in this uncertainty range can be substantial in relatively open geometries such as in SP0 and SS4, with run times accelerating by over 20%. In tighter or more heterogeneous samples, such as SP6 and SS8, the initial run time improvement is not as significant. This is because the solver needs to fix prediction errors that occupy a larger fraction of the computational domain.

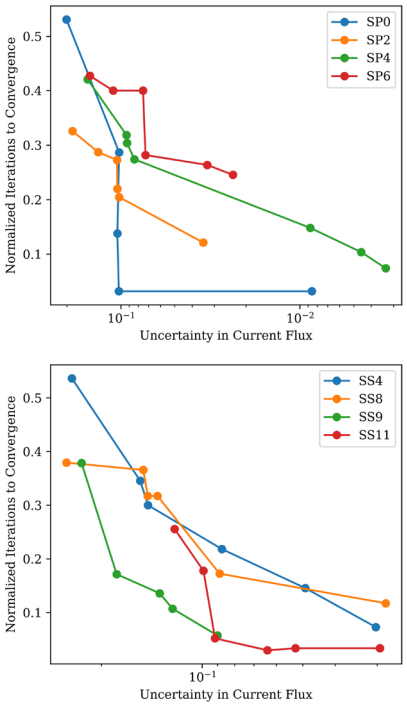

Fig. 11. Normalized number of iterations to convergence error versus uncertainty in current flux (decreasing to the right). The uncertainty in current flux provides a direct point of comparison to the solver. We see a large drop in run time at uncertainty ~10% for both sphere packs (left) and sandstones (right).

Next, when the uncertainty in current flux is reduced to 0.1-0.2, we see a sudden drop in the necessary iterations to convergence. In some cases, such as in SP0 and SS11, the number of required iterations drops to under 10% of that of the linear gradient assumption. The size of the drop is not necessarily dependent on the porosity or complexity of the sample. In the sphere pack samples, where the sphere locations remain constant, smaller porosity geometries do not tend to see the same level of run time improvement as their larger porosity counterparts. However, the amount by which the tightest sample improves is larger than some of the more open samples.

After the large drop, the run time improvement curve either levels off or continues a more gradual decrease. The curves appear to approach a theoretical run time limit, constrained by the solver itself and the level to which the model can resolve the blocks in the prediction. We cannot report further run time improvement in samples that have already reached this limit (SP0 and SS11), even with further training and error reduction. In the cases where the theoretical limit has not yet been reached, it appears the geometric complexity of the pore space dictates the rate by which the run time continues to improve. Using porosity as a proxy for complexity, we see that larger porosity samples tend to improve more rapidly after the initial drop than smaller porosity samples. This is consistent for both sphere pack and sandstone samples. We also observe that for similar porosity

samples (~ 0.2), the run time in this regime improves more slowly for more geometrically complex samples (sandstones) than for simpler samples (sphere packs).

Finally, we discuss the Laplacian of the image as an indicator tool of solver run time improvement. During training, we identified the refinement block artifacts as the largest source of errors. As training progresses, the block artifacts tend to become more refined and the D-2 score decreases. We mask the pore space immediately adjacent to grain-pore boundaries (Euclidean distance = 1) to mitigate the effects of these boundaries on the D-2 score. We also found that normalizing the D-2 score by the solid volume raised to the 2/3 power helped to overlap the curves for better comparison. Physically, the D-2 score accounts for phase discontinuities at the solid-pore boundary, so raising the solid volume to the 2/3 power provides a value dimensionally consistent with surface area.

We see behavior in the sphere pack samples like that of the uncertainty in current flux where a large drop in run time precedes more gradual improvement (Fig. 12). However, we also see non-unique values of iterations to convergence for a singular D2 score, indicating that it may not be as good of a predictor as the uncertainty in current flux.

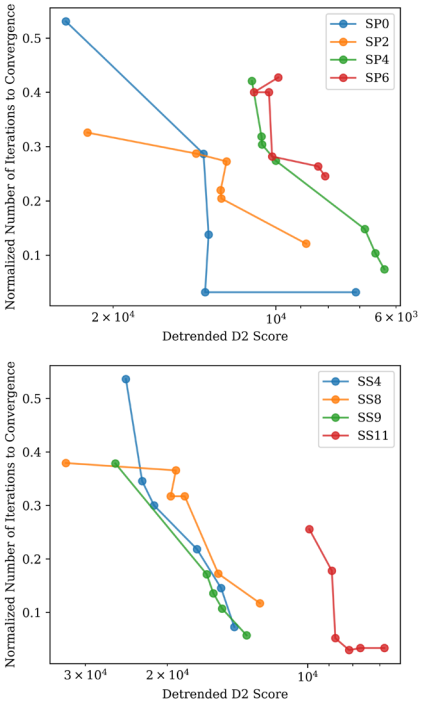

Fig. 12. Normalized number of iterations to convergence error versus normalized D2 score (decreasing to the right). The D2 score is normalized by $(1-\phi)^{\frac{2}{3}}$. While we see similar behavior as Fig. 11, several D2 scores exhibit non-unique iterations to convergence. However, it is still a good indicator of how blocky the prediction is.

In the sandstone samples, we see that the normalized curves agree with each other quite well, except for the tightest sample (SS11). In the other three cases, we observe a well-behaved decrease of run time in this normalized D-2 score range. This indicates that a non-trainable Laplacian filter or D-2 score could be implemented in a neural network as a performance metric and could help guide model training.

4 Conclusions

This study confirms that the proposed framework of initializing a solver with a machine learning prediction can drastically reduce the run time needed to achieve a deterministic result. Here, we limit the samples to a size of 200^3 due to GPU limitations and we observed over an order of magnitude run time improvement.

Throughout this study we also conclude the following:

- Commonly used error metrics (i.e., MAE and MSE) do not always provide adequate detail of the predictive performance of the model.
- An initial ML prediction does not need to be exact to result in an accurate formation factor prediction
- Supervised metrics, such as MSE and PSNR, are good indicators of run time improvement.
- Achieving a convergence metric of ~ 0.1 – 0.2 drastically decreases the number of iterations to convergence.
- The D2 score can be incorporated into the model loss to help smooth the predictions.

This framework has important implications for training predictive models. These metrics can serve as performance criteria when the model is extended to unseen data. It is clear that even overfit models struggle to make perfect predictions and that solvers are still needed to enforce the conservation laws the solution is required to obey.

The sizes of the image data used in this study are admittedly small because we predicted on the same set of images as was used to train the models. The intent was to assess the sensitivity of the run time improvement to the proposed error metrics. Because testing can be performed on CPUs, the prediction samples are not limited to the same sizes as the training data. We expect more significant run time improvements if applying the framework to larger image samples where any gains are a balance between the reduced accuracy when predicting on unseen images and a model's ability to make predictions effectively instantaneously.

Future work will emphasize the development of more generalizable models and assessing these metrics on larger images. Because the actual prediction space accounts for less than 30% of the entire image (only pore space), training on larger images could possibly be achieved by supplying the network with a sparse representation of the pore space during training. Incorporating findings from this study, we plan to implement the finite difference kernel used to find the current flux as a hard constraint, thereby limiting the prediction space to more physical solutions. We also plan to incorporate the Laplacian filter into the cost function to mitigate the block effects of the refinement process. More robust models can be developed by including morphological feature maps (e.g., Euclidean distance map, time of flight, maximum inscribed sphere, etc.) as extra training channels, instead of only using the binary image.

Acknowledgements

We gratefully acknowledge the Digital Porous Media (formerly Digital Rocks Petrophysics) Industrial Affiliates Program for their support and technical expertise (B.C., M.P).

J.S. gratefully acknowledges the support of the U.S. Department of Energy through the LANL/LDRD Program and the Center for Non-Linear Studies.

The Texas Advanced Computing Center provided high performance computing resources used to perform the simulations and training reported in this study.

References

1. P. Glover. Treatise on Geophysics, vol. 11,477 89–137, 10.1016/B978-0-444-53802-4.00189-5 (2015).
2. A. Revil, M. Karaoulis, T. Johnson, and & A. Kemna. Hydrogeol. J. 20, 617–658, (2012).
3. P. Egbelehulu, M. Abu, N. Abdulsalam, and A. Taiwo. Pak. J. Geol. 4, (2020).
4. J.E. Carothers and C.R. Porter. The Log Analyst 12, no. 01 (January 1, 1971).
5. C.F. Berg. Physical Review. E, Statistical, Nonlinear, and Soft Matter Physics 86, no. 4 Pt 2 (October 2012): 046314.
6. J. H. Norbisrath, G.P. Eberli, B. Laurich, G. Desbois, R.J. Wegeer, and J. L. Urai. AAPG Bulletin 99, no. 11 (2015): 2077–98.
7. W.O. Winsauer, H. M. Shearin Jr., P. H. Masson, and M. Williams. AAPG Bulletin 36, **2,** 253–277 (1952).
8. G.E. Archie, Transactions of the AIME 146 **01**, 54–62 (1942)
9. Q.F. Niu and C. Zhang. Geophysical Research Letters 45, **4,** 1870–1877 (2018).
10. M. Prodanović, M. Esteva, J. McClure, B. C. Chang, J. E. Santos, A. Radhakrishnan, A. Singh, and H. Khan. Society of Core Analysts (2022).
11. A. Mehmani, R. Verma, and M. Prodanović. Marine and Petroleum Geology 114 (2020): 104141.
12. B. Bijeljic, A. Raeini, P. Mostaghimi, and M.J. Blunt. Physical Review E 87, no. 1 (January 10, 2013): 013011.
13. J. E. Santos, D. Xu, H. Jo, C.J. Landry, M. Prodanović, and M.J. Pyrcz. Advances in Water Resources 138 103539 (2020).
14. S. Mohaghegh. Journal of Petroleum Technology - J PETROL TECHNOL 52 (September 1, 2000): 64–73.
15. J. Pollock, Z. Stoecker-Sylvia, V. Veedu, N. Panchal, and H. Elshahawi. “Machine Learning for Improved Directional Drilling.” In Offshore Technology Conference, 2018.
16. L.P. Zhu, H.Q. Li, Z.G. Yang, C.Y. Li, and Y.L. Ao. Petrophysics-The SPWLA Journal of Formation Evaluation and Reservoir Description 59, no. 06 (2018): 799–810.
17. Sudakov, E. Burnaev, and D. Koroteev. Computers & Geosciences 127, 91–98 (2019).
18. J.E. Santos, Y. Yin, H. Jo, W. Pan, Q. Kang, H.S. Viswanathan, M. Prodanović, M.J. Pyrcz, and N. Lubbers. Transport in Porous Media 140, **1,** 241–272 (2021).
19. Y.D. Wang, T. Chung, R.T. Armstrong, and P. Mostaghimi, arXiv, (2020).
20. J.E. Santos, B.C. Chang, Q. Kang, H.S. Viswanathan, N. Lubbers, A. Gigliotti, and M. Prodanović. Digital Rocks Portal, (2021).
21. J.E. Santos, B.C. Chang, A. Gigliotti, Y. Yin, W. Song, M. Prodanović, Q. Kang, N. Lubbers, and H.S. Viswanathan, Scientific Data, (Under Review).
22. J. Finney, Digital Rocks Portal, (2016).
23. R. Neumann, M. Andreeta, and E. Lucas-Oliveira. Digital Rocks Portal, (2020).
24. R.A. Victor, Pore Scale Modeling of Rock Transport Properties (2014).
25. J. Mcclure, James, Digital Rocks Portal, (2016).
26. R. Bohn and E. Garboczi. User Manual for Finite Element and Finite Difference Programs: A Parallel Version of NISTIR 6269 and NISTIR 6997 (2003).
27. M. Han, S. Youssef, E. Rosenberg, M. Fleury, and P. Levitz. Physical Review E 79, no. 3 (March 30, 2009): 031127.
28. K. Kawaguchi, L.P. Kaelbling, and Y. Bengio, arXiv, (2020).
29. H. O’Reilly and J. M. Beck. International Journal For Numerical Methods in Engineering 0, 1-16 (2006).