Can Machine Learning Extract Useful Information about Energy Dissipation and Effective Hydraulic Conductivity from Gridded Conductivity Fields?

Moghaddam, M.A[1], Ty P.A. Ferre[1], Klakovich J [1], Gupta HV[1], Ehsani MR[1]

[1] Hydrology and Atmospheric Sciences, University of Arizona, Tucson, AZ, USA

Corresponding author: Ty Ferre (tyferre@gmail.com)

Abstract

We confirm that energy dissipation weighting provides the most accurate approach to determining the effective hydraulic conductivity (K_{eff}) of a binary K grid. Machine learning and deep learning algorithms of varying complexity (decision tree, vanilla CNN, UNET) can infer K_{eff} with extremely high accuracy ($R^2 > 0.99$), even given only the fraction of the grid occupied by the high K medium. Adding information derived from the energy dissipation distribution improved each algorithm. However, all methods failed to infer K_{eff} accurately for outlier cases, all of which were inferred accurately using energy dissipation weighting directly. The UNET architecture could be trained to infer the energy dissipation weighting pattern from an image of the K distribution with high fidelity, although it was less accurate for cases with highly localized structures that controlled flow. Furthermore, the UNET architecture learned to infer the energy dissipation weighting even if it was not trained on this information. However, the weights were represented within the UNET in a way that was not immediately interpretable by a human user. This reiterates the idea that even if ML/DL algorithms are trained to make some hydrologic predictions accurately, they must be designed and trained to provide each user-required output if their results are to be used to improve our understanding of hydrologic systems most effectively.

1- Introduction

Numerical modeling is fundamental to understanding hydrologic systems, and to predicting outcomes to be used for water resources management and groundwater contaminant remediation *[Ahuja et al., 2010; Chan & Elsheikh, 2017; Aliyari et al., 2019; Shamsudduha et al., 2019*]. Water

movement through the subsurface is controlled largely by the hydraulic conductivity of the region, which can vary over orders of magnitude across multiple scales [*Green et al., 2009*].

Recent advances in hydrogeophysics increasingly suggest that the spatial pattern of hydraulic conductivity can be mapped effectively *[Slater, 2007a; Hertrich, 2008; Dlubac et al., 2013*]. Coupled with carefully selected point measurements of hydraulic conductivity, these methods offer the promise of real improvements in our ability to accurately model water flow and associated solute transport in the subsurface. However, it is less clear whether "indirect methods" can be used to infer upscaled effective values for hydraulic conductivity in a heterogeneous medium; i.e., it is unclear whether such methods can be used to infer the same effective conductivity value that would be obtained via detailed modeling using the highly resolved conductivity field. One major challenge to achieving this goal is the current lack of understanding of how spatial structure (patterns in the conductivity field) affect the overall hydraulic conductivity of the medium. In this study, we examine whether machine learning tools can provide insight into the problem of hydraulic conductivity upscaling.

There is a rich body of literature on the upscaling of hydraulic conductivity. *Wen & Gómez-Hernández [1996]* categorized upscaling techniques as being either local or non-local. Local techniques, which include simple averaging, power averaging, renormalization, and percolation theory, are based on the assumption that effective upscaled conductivity depends only on the statistical distribution of media of different conductivities contained within the medium. Non-local techniques, which include inverse modeling and energy dissipation, also consider how boundary conditions affect flow.

Local methods based on simple or power averaging [*Journel et al., 1986; Matheron, 1965; Desbarats & Srivastava, 1991; Zhu & Mohanty, 2002; Masihi et al., 2016*] typically represent the

domain in terms of fractions, each having a single conductivity, and exponentially weight the conductivity of each fraction by the percent area or volume that it occupies. The extreme cases of arithmetic weighting (exponent of 1, conceptually representing flows in parallel) and harmonic weighting (exponent of -1, conceptually representing flows in series,) bound these approaches [*Cardwell & Parsons, 1945*]. In general, local approaches work well provided that the medium is approximately homogeneous; i.e., the spatial distributions of the fractions are not organized into patterns, giving rise to structure [*Durlofsky, 1992*]. For any specific case, the value of the exponent can be estimated by running a flow model *[Wen & Gómez-Hernández, 1996; Colecchio et al., 2020*], but this requires the extra step of running the flow model to determine the effective conductivity, which is often counter to the intended purpose of the upscaling effort.

The renormalization method to compute block conductivity (K_{eff}) is based on upscaling by a recursive calculation whereby the extent of each grid unit is doubled along each direction at each step *[King, 1989; King & Neuwelier, 2002*]. This approach essentially allows for the use of arithmetic and harmonic averaging at the local scale, thereby simplifying the computation of effective conductivity. However, while the method is very fast and efficient, severe errors can occur in the final estimates at the scale of the largest blocks due to unrealistic boundary representations during the recursive upscaling process [*Malick, 1995*]. Further, as with the exponential approach, the renormalization method is only applicable to statistically isotropic, lognormal conductivity fields having no clear structure *[Sánchez‐Vila et al., 1995; Wen & Gómez-Hernández, 1996*].

A significant advancement in the upscaling of K for binary media was achieved by the introduction of percolation theory, proposed by *Vinay Ambegaokar [1971]* to model electron hopping in semiconductors. The percolation concept was applied to hydrogeology by *Katz & Thompson*

[1985] to compute the K_{eff} of a medium characterized by a strong contrast between low and high conductivities, with the assumption that the upscaled value of conductivity is primarily a consequence of flows through connected high permeability pathways when they exist [*Slater, 2007b; Ambegaokar et al., 1971*]. Subsequent studies in which percolation theory was used to assess K_{eff} *[Berkowitz & Balberg, 1993; Hunt, Allen, Robert Ewing, 2014; Hunt & Sahimi, 2017*] have generally found that percolation theory is appropriate when the proportion of the high conductivity medium is close to the percolation transition threshold [*Colecchio et al., 2020*].

Non-local methods can be used to infer effective values for system parameters via inverse modeling, wherein the parameter field is constrained to be homogenous and the corresponding best-fit equivalent upscaled parameter value is determined; several recent studies [*Hassanzadegan et al., 2016; Kotlar et al., 2019; Cheng et al., 2019; Coutinho de Oliveira et al., 2020*] have used this technique for vadose zone parameter estimation. However, this approach requires solving the flow problem, including the boundary conditions, which requires that many observations are available to properly constrain the parameter estimation problem. This can be very computationally demanding [*Vrugt et al., 2008*], further, *Lai & Ren [2016]* have shown that this approach can provide imperfect results; e.g., they showed that three different inverse approaches applied to a one-dimensional situation resulted in models that were unable to reproduce the average soil water content profile.

The most direct approach to determining how spatially variable averaging of hydraulic conductivities occurs during flow is through energy dissipation analysis. This approach is largely limited to steady-state problems, and also requires solving the flow problem to determine the effective, upscaled parameter value. In essence, the energy dissipation approach defines the energy per unit time required to force the fluid through each block of the porous medium; this

value is normalized for the shape of the domain and the boundary conditions, and then can be used to define the spatial distribution of weights to be applied to the local conductivity values when upscaling to determine K_{eff}. In this regard, *Knight [1992]* and *Indelman & Dagan [1993]* suggested that K_{eff} can be determined from a grid of cells by assuming that dissipated energy must be preserved during the equivalent block conductivity computation.

Although the energy dissipation approach is computationally demanding and requires that the flow problem be solved for both the homogeneous and heterogeneous case, it has been found to be the most accurate and mathematically rigorous way to upscale conductivity for steady state problems [*Colecchio et al., 2020*]. Further, it can provide significant insight into the specific locations that contribute most to the upscaled value of K_{eff}. Borrowing on the approach to defining the sample area of time domain reflectometry probes using this approach (*Ferre et al., 1998*), it is possible to identify relatively small areas of the domain that contribute disproportionately to the value of K_{eff}, thereby identifying key structures in the subsurface that may be controlling flow.

In recent years, due to advances in storage, computation, and graphic processing power, machine learning (ML) and deep learning (DL) have gained popularity in different research areas, including in natural language processing *[Allison Marier et al., 2016; Yadav & Bethard, 2019; Zhao & Bethard, 2020*], computer vision *[O'Mahony et al., 2020; Ronneberger et al., 2015; Voulodimos et al., 2018; Liu et al., 2020*], economy [*Luo et al., 2017; Mai et al., 2019*], and hydrology *[Assem et al., 2017; Nearing et al., n.d.; Kratzert et al., 2019; Demiray et al., 2020*]. In hydrogeology, several attempts have been made to use deep learning to infer the forms of the equations governing flow through porous media *[Afzaal et al., 2019; Mo et al., 2020; Tartakovsky et al., 2020; Wang et al., 2020*].

In particular, the architecture underlying convolutional neural networks (CNNs) allows for the preservation of spatial structure and correlation information, and we might therefore expect that the CNN approach is particularly suitable for problems involving gridded inputs, such as hydraulic conductivity fields *[Chan & Elsheikh, 2017; Canchumuni et al., 2018; Mo et al., 2020; Zhou et al., 2020*]. For example, *Zhou et al. [2020]* used a CNN to map conductivity fields to macro-dispersivity, *Wu et al. [2018]* combined images of porous media with integral quantities of porosity and specific surface area to estimate pore-scale permeability, and *Mo et al. [2020]* parameterized a non-Gaussian conductivity field using a convolutional adversarial autoencoder as well as proposing a deep residual dense CNN to map spatially distributed conductivity to head and solute concentration for 2D and 3D media.

Reviews of several studies (e.g., *Tartakovsky et al., 2020; Mo et al., 2020*) indicate that data-driven approaches are efficient, and can even outperform stochastic modeling or local (i.e., structure-based) techniques. In particular, in the context of estimating effective parameter values, the accuracy of CNN-based approaches can be attributed to the fact that, unlike classic stochastic approaches that only consider the first and second statistical moments of a highly spatially variant media, the machine learning approaches can account for spatial patterns that are not explicitly characterized by those statistical moments [*Zhou et al., 2020*] or by classical structure-based models.

Despite their impressive predictive power, ML-based models can suffer from a lack of interpretability [*Chakraborty et al., 2018; Apley & Zhu, 2016*]. Most studies *[Srisutthiyakorn, 2016; Mosser et al., 2017; Wu et al., 2018; Zhou et al., 2020*] have mapped from measured inputs to outputs without due consideration of the underlying physical processes involved. Consequently, several studies *[Raissi et al., 2019; Tartakovsky et al., 2020; Wang et al., 2020*], have attempted

144 to incorporate physical constraints into DL algorithms. For example, *Wang et al. [2020]* used a
145 knowledge-based neural network to estimate head distribution by taking into consideration the
146 residuals of the governing equations, boundary conditions, and expert knowledge when
147 formulating the loss function used to train the model. *Tartakovsky et al. [2020]* incorporated
148 governing flow partial differential equation constraints (the Darcy and Richards equations) along
149 with training data into a DL algorithm to infer the hydraulic conductivity map based on sparse
150 observations of head and conductivity during saturated flow through a heterogeneous medium and
151 to infer the constitutive pressure-conductivity relationship from observations of capillary pressures
152 during unsaturated flow.

153 The aforementioned studies represent clear advances in the use of ML/DL for the upscaling of K_{eff}.
154 However, to date, little attention has been paid to the design of the underlying ML/DL architecture.
155 Further, we found no publications addressing the problem of how the ML/DL approach extracts
156 and uses information from the heterogeneous field in the process of inferring K_{eff}. Here, we make
157 use of recently developed approaches that facilitate comparing the activation patterns of different
158 DL models [*Kornblith et al., 2019a*] to examine how these ML tools extract and use the knowledge
159 that is relevant to the process of upscaling (i.e. energy dissipation weighting).

160 This study has three primary objectives. The first is to examine the potential for using ML/DL to
161 infer effective the hydraulic conductivities of two-dimensional binary conductivity fields; these
162 represent the simplest fields that display different levels of importance of K field structure (spatial
163 organization) on effective conductivity value K_{eff}. Conceptually, these binary fields can be viewed
164 as simplifications of bimodal K fields that can result from coastal depositional processes and
165 fracturing in low permeability media [*Knudby et al., 2006*]. The second is to evaluate the
166 performance of a specific type of CNN, an image to image translation algorithm known as UNET,

to infer the structure of energy dissipation weighting directly from binary K grids. The third is to compare the ability of a UNET to infer K_{eff} from a binary K grid when trained on the energy dissipation weighting to that when trained only on the K grid. In this regard, we examine how information is processed by the UNET, to examine whether it is accounting for energy dissipation 'naturally', even when it is not explicitly trained using such information.

2- Methodology

We examined the effect of the structure of a binary medium on the effective hydraulic conductivity, K_{eff}, using the MODFLOW numerical 2-D groundwater model to produce the steady-state head distribution over a square grid with a 1-D applied gradient. We computed K_{eff} from the geometry of the grid, the applied Type I boundary conditions, and the steady-state flow through the system for different random distributions of two media with different K values. We also computed the energy dissipation in every cell to examine whether this information can provide insight into the spatial weighting of the K values used to determine K_{eff} [*Indelman & Dagan, 1993*]. Further, we examined a range of approaches to infer K_{eff} from the K grid, with and without information about the energy dissipation distribution, including simple tree-based and more complex, CNN-based,

machine learning algorithms. Finally, we used central kernel alignment similarity *Kornblith et al., [2019b]* to infer the hidden layer representation for K_{eff} estimation in an attempt to understand how and whether the deep learning algorithm considers energy dissipation during K_{eff} estimation.

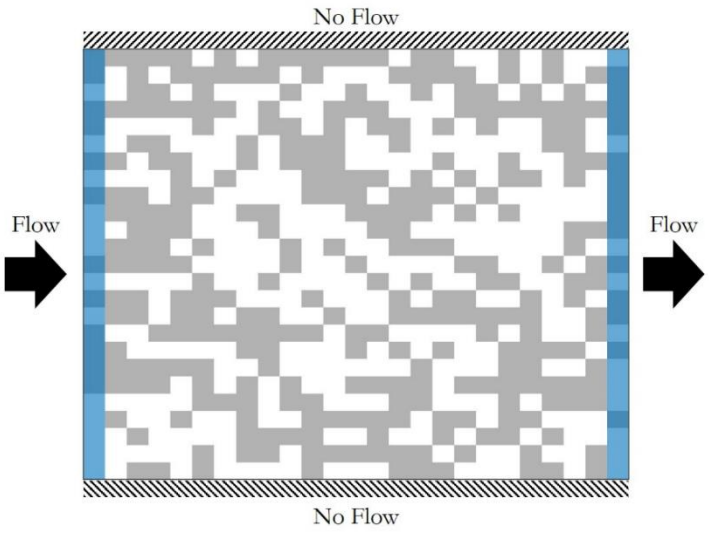


Figure 1. Sample 25x25 cell grid with 50% high K (white) and 50% low K (grey) cells, constant head boundaries (blue), and no flow boundaries (diagonal lines). The left boundary has a constant head of 2 and right boundary has a constant head of 1, with flow occurring from left to right.

2-1- Flow through Heterogeneous Binary Grids (Dataset Generation)

We defined 25 by 25 grid domains with no flow boundaries at the top and bottom and constant head boundaries of 2 m and 1 m on the left and right boundaries, respectively. Each cell has a length of 1 m on a side. Two media populated the grid, with K values of 1 and 0.001 cm/s. Different percentages of the prevalence of the high K material were considered, ranging from 1% to 99%. For each high K percent, 3000 random distributions of the media were modeled. Figure 1 shows one example of a grid with 50% high K material.

For each grid, the effective hydraulic conductivity was computed based on Darcy's Law, the global gradient applied over the domain, and the steady-state flow through the system. The convergence criterion on the head used in MODFLOW was 0.01 m. To account for small errors that persisted when the convergence criterion was met, the value of K_{eff} was calculated based on the flow into the left boundary and the flow out of the right boundary. The resulting K_{eff} values calculated at both boundaries agreed within 1%, and the average value was used for all analyses.

205

206 ## 2-1-1- Energy Dissipation Weighting Method

207 Conceptually, energy dissipation is defined as the energy per unit time necessary to force the fluid
208 through the porous medium [*Indelman & Dagan, 1993*]. The value of K_{eff} can be thought of as a
209 weighted average of the spatially distributed values of K. Energy dissipation can be used to define
210 the spatial distribution of weighting factors based on the square of the gradient of the potential at
211 each location normalized by the sum of the square of the gradient of the potential for the same
212 boundary conditions for a domain filled with a homogeneous medium [*Knight, 1992*]. The
213 weighting factor at a point at (x,y) can be expressed as:

214

215 $$W\ (x,y) = \frac{[\nabla \emptyset(x,y)]^2}{\iint [\nabla \emptyset_0(x,y)]^2 dxdy} \qquad \text{Eq. 1}$$

216

217 where w(x,y) is the weighting factor at point (x,y), $\emptyset$(x,y) is the potential at each location, and $\emptyset_0$
218 is the potential distribution for the equivalent homogenous field. *Knight [1992]* showed that
219 spatially variable properties (e.g. for K) can be weighted to determine an upscaled property (here
220 K_{eff}) as the sum of the local K weighted by the energy dissipation weighting factor over the domain,
221 as:

222 $$K_{eff} = \iint W(x,y)K(x,y)dxdy \qquad \text{Eq. 2}$$

223

224 In this study, the steady-state head values were used to compute the energy dissipation distribution.
225 Because MODFLOW determines head values at the nodes and the K values are defined over the

cells, the gradient and K values are not aligned. There are two approaches to compute K_{eff} with the energy dissipation approach for these conditions. First, the gradient can be computed at each cell edge and the value of K_{eff} at the edge can be determined based on the K value in the two neighboring cells. Second, the head values can be interpolated to the edges, allowing for gradients to be computed at the nodes, matching the locations of the K grid. Both of these approaches were tested and were found to agree within 1%; accordingly, the average of these two estimates of K_{eff} was used for each grid for further analyses. Hereafter, the energy dissipation weights are referred to as ED weights, or simply as weights.

2-1-2- Estimating K_{eff} with a Regression Tree with and without ED Weights

For a given percent of high K material, the energy dissipation distribution depends on the structure and arrangement of high K and low K cells in the domain. As a result, the K distribution and the ED weight distribution are related (but not identical) sources of information for inferring K_{eff}. Throughout this study, we examine whether added knowledge of the ED weight distribution improves the estimation of K_{eff} by machine learning and deep learning algorithms. First, we compare the performance of simple regression tree (RT) models with and without the inclusion of energy distribution information. We use a regression tree as a very simple ML to provide a baseline of comparison for more advanced machine learning algorithms. For completeness, a gradient boosting algorithm was also applied, but its performance was not significantly different than the RT, so we opted for the simpler version of the tree-based ML.

Because RT models are not well-suited to considering spatial relationships among inputs, rather than providing the RT models with the K grid and spatially distributed weights, we provided only

248 the following summary statistics: the percent of high K grids within the domain, and the fraction
249 of the cells that contain high spatial weights. To label the high ED weight cells in a grid, we
250 followed the procedure suggested by [*Ferré et al., 1998*]. We first computed the energy
251 distribution (Eqns. 1 & 2) after solving the steady-state head distribution with MODFLOW. The
252 weights were sorted in descending order, and the cells that contributed 95% of the total weight
253 were identified as high energy cells, thereby defining the smallest area contributing 95% of the
254 total weight in determining K_{eff}.

255 To apply the RT analysis, we considered paired values of targets (K_{eff}) and features (percent of
256 high K material with or without the fraction of high energy cells). These were divided randomly
257 into 65% training, 15% validation, and 20% testing. The models were trained using the training
258 set and tuned using the validation set. We chose to use a binary split at each node, such that the
259 RT sequentially divided the training set of K_{eff} samples at each node into two subsets. The split
260 was based on the choice of a feature and a threshold value such that every sample was identified
261 as being above or below that threshold. At each point of division, the tree maximized the reduction
262 in overall variance such that the sum of the population-weighted variability of the K_{eff} values in
263 the two subsets was less than that of the parent set. Because each feature and threshold
264 identification is made without regard to any future or past selections, RT is known as a "greedy"
265 algorithm and is not guaranteed to be optimally efficient. Furthermore, RT results in binned
266 outputs, with a single value applied to all samples that fall into the final node along each branch.

267 The performance of the RT was based on its ability to predict K_{eff} for the testing set. This
268 procedure was then repeated with only the percent high K material provided to the RT (i.e., without
269 also providing the fraction of high energy cells). A comparison of the performance of these two

RT models provides an estimate of the value of the information contained in the fraction of high weight cells for inferring K_{eff} with RT.

2-1-3- Estimating K_{eff} with a Vanilla CNN with and without ED Weights

The estimation of K_{eff} can be viewed as an image mapping problem: i.e., our goal is to map a grid of 25x25 K values onto a single value of K_{eff}. From this perspective, together with the expectation that the spatial patterns of K within the domain are important to predicting K_{eff}, a convolutional neural network (CNN) is a natural choice for a deep learning method to apply to this problem.

CNN is a class of supervised learning algorithms that is suitable for processing image-based datasets. It consists of two main types of components, a convolutional kernel, and a pooling layer. A kernel is a sliding window of weights used to extract "features" from the inputs by convolution of the inputs using the kernels, and propagation of the result to the next layer of the network. Application of an activation function provides nonlinear elementwise transformation. Pooling layers are optional blocks of a network that reduce the size of the hidden layers.

Herein, a 'vanilla' CNN refers to a common architecture comprising several convolutional layers that are fully-connected to a dense output layer to perform predictions (see structure presented in Table A1-A of the appendices). The number of layers and filters were selected via hyperparameter tuning using the validation dataset. Dropout regularization was not included. For the cost function, we used the mean squared error between the observed and predicted K_{eff} in the training samples. The Adam algorithm [*Kingma & Ba, 2015*] was used to optimize the model parameters and weights.

291 As for the RT analyses, the CNN was trained both with and without providing the ED weights as
292 input information. As shown previously by *Knight [1992]*, knowledge of the K field and the ED
293 weights provides a near-exact definition of K_{eff}. Therefore, we provided a degraded metric related
294 to the ED weights; specifically, a 25x25 0/1 masking matrix with 1 indicating high energy cells
295 was used as a surrogate for the ED weights.

296

297 2-1-4- Estimating K_{eff} with and without ED Weights

298 Recent advances in the application of deep learning to image processing have led to the
299 development of powerful machine learning architectures. In particular, given that the knowledge
300 of energy dissipation has been shown to provide valuable information regarding the weighting
301 required to define K_{eff}, the problem of estimating K_{eff} from a grid of K values can be seen as a
302 problem that has two stages. The first is to estimate the energy dissipation weighting at each cell,
303 and the second is to use the estimates of the spatially distributed ED weights to estimate K_{eff}. The
304 UNET architecture [*Ronneberger et al., 2015*] was developed to address problems that require
305 consideration of multiple scales by including skip connections, which recombine information from
306 earlier hidden layers with that of later hidden layers. Here, we propose a modified UNET
307 architecture that estimates the spatial weight distribution and then combines this estimate with the
308 K grid to estimate K_{eff} (Figure 2).

309 We applied the UNET in two different ways to understand if and how ED weighting is used in the
310 estimation of K_{eff}. In the first implementation, referred to as 'informed', the model is trained using
311 the freeze-training technique [*Zoph et al., 2016; Brock et al., 2017*], in which the lower branch of
312 the model, Figure (2), up to the point that the K grid information is reintroduced, is first trained to

estimate the spatially distributed ED weights. This is achieved by providing the ED weights during training. Once trained, the informed UNET is then used to predict K_{eff} ***without*** being provided ED weights. This is possible because UNET models are a variation of encoder-decoder algorithms, which include a contracting path (like the vanilla CNN) followed by an expanding path. The contracting path (i.e., encoder) is responsible for capturing the context while the expanding path (i.e., decoder) enables localization. Through the use of encoder-decoder paths, the UNET can provide an output that has the same dimensions as the input. In our application, this property is necessary to obtain ED weights on a grid having the same size as the K grid. Making use of this structure, we trained UNET to infer the ED weights and then used those inferred weights to predict K_{eff}. In other words, for the informed UNET, the weights of the lower branch were frozen after training, and training was then continued by feeding only the K grid into the UNET. The algorithm then provided estimates of the ED weights, which were concatenated with the K grid and fed into the final fully-connected layer. This model was trained to estimate K_{eff}.

The second implementation of UNET is referred to as 'uninformed'. The model structure was identical to the informed UNET, but was only provided K grid information; it was not trained using any information about the actual ED weights. Rather, all weights in the model were fitted simultaneously during training to fit K_{eff}.

The details of our UNET structure are provided in the Appendices (Table A1-B). Briefly, the contracting path is comprised of repeated blocks of two consecutive 3x3 convolutional kernels with rectified linear activation functions (Relu) followed by a 2 x 2 max-pooling layer with a stride of 2 to reduce the number of parameters and diminish the next layer input size. On the contracting path, multilevel decomposition is applied to each layer, doubling the number of feature maps (i.e., filters) at each step. The expanding path consists of repeated blocks of transposed convolution

layers with a kernel size of 2X2 and a stride of 2. In each block, the output of the transposed convolution layer is concatenated with the cropped feature map of the corresponding step from the encoding procedure (a skip connection). The concatenated values are subjected to two consecutive 3x3 convolutional kernels with Relu activation functions. The skip connections help to recover information that may be lost by down-sampling during decoding. The cropping procedure in the concatenation ensures that the tensor extracted from the encoder will have the same size as the corresponding layer in the decoder. During decoding, the convolutional layer halves the number of channels. A final convolution layer with a kernel size of 1X1 and linear activation maps the current number of channels to a single layer. A skip connection was introduced to recover information of the original grid, like the percent of high K, that may be lost by when inferring the ED weights. Specifically, the inferred ED weights were concatenated with the K grid and fed through a convolutional layer and a dense, fully-connected layer to estimate K_{eff}. It should be noted that as part of preprocessing, we padded the input image to 32×32 to make the final output of the UNET the same as the original image.

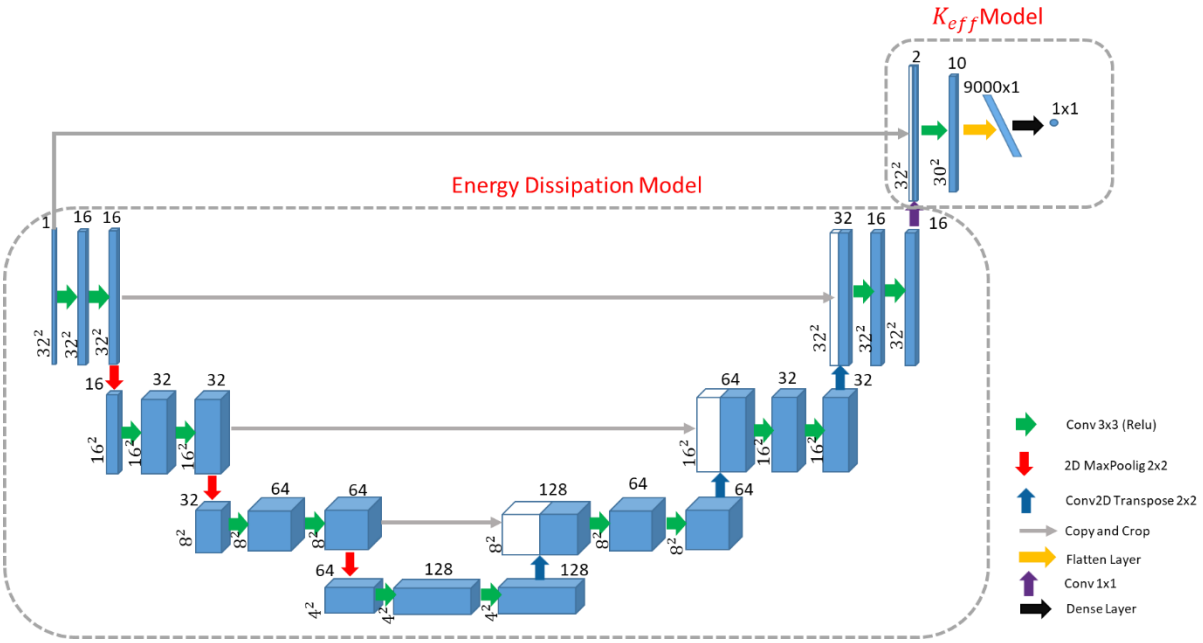


Figure2) Proposed U-net architecture. The architecture is composed of two submodel. Energy dissipation model has a UNET shape structure followed by a CNN model to map output of UNET to Keff. Blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent skipped connection. The arrows are operations performed on feature maps described in the legend.

2-2- Model Evaluation

Identical data were provided to all of the methods; specifically, the K grids (dimension 3000*99) MODFLOW-determined K_{eff} values, and (where applicable) ED weights. The inputs and targets were divided into training, validation, and testing subsets. A random selection of 65% of these inputs was used for training and 15 % were used as a validation dataset for hyperparameter tuning. The same training/validation/testing sets were used for all of the analyses reported herein (e.g. Regression Tree, vanilla CNN, etc.). Model performance is reported using the testing data set, comprising the remaining 20% of the data. Before training, the inputs were standardized by subtracting the mean value and dividing by the standard deviation. All hyper-parameters were tuned using a grid search approach. The root mean squared error (RMSE) between the observed

and model-calculated values (of K_{eff} or ED weight) is used to assess the prediction quality of each model. The R^2 value was also calculatedbut is only used to further illustrate the quality of the predictions.

2-3 Deep Learning Implementation

All deep learning architectures were implemented in Python 3.6.9 with Tensorflow V. 2.2.0 and CUDA version 10.1. Training and predictions were done on a P100 NVIDIA GPU. For both the "informed" and "uninformed" models, we used Adamax with a learning rate of 5e-4 as the optimizer. For the vanilla CNN, the Adam optimizer with a learning rate of 1e-4 was used. For all cases, training was stopped when performance on the validation dataset stopped improving within a patience value equals to 50.

2-4- CKA and Similarity Analysis

In addition to investigating whether machine learning algorithms can be trained to predict K_{eff} using gridded binary K information, we also wanted to determine whether these tools can infer the underlying pattern of energy-dissipation in the process of inferring K_{eff}. If it can be shown that the deep learning procedure naturally infers the spatial distribution of energy dissipation, then it would provide an example of how DL tools can "learn" underlying concepts. Further, because the distribution of energy dissipation indicates which parts of the medium are having the largest impact on steady-state flow, the ability to make inferences regarding these patterns would also enable an understanding of the relationship between K_{eff} and the structure of the K distribution. Such

knowledge would also be valuable for understanding soil property distributions that may impact dispersion, colloid trapping/mobilization, and erosion/piping.

To investigate the ability of deep learning tools to make inferences regarding the underlying pattern of energy dissipation, we applied the UNET methodology in both informed and uninformed modes, as described above. To compare how information flowed through the UNET in informed and uninformed modes, we examined the intermediate representations (i.e., hidden layer outputs) of each trained model. Specifically, the hidden layer outputs, known as hidden representations, characterize the "features" learned by a hidden layer of a neural network from an input (i.e, K grid), represented in a machine-readable format. Similarity measurements can be used to compare these intermediate representations between networks.

Kornblith et al. [2019] showed that for a similarity index to be suitable, it should be invariant to orthogonal transformation and isotropic scaling, and not be an invertible linear transformation. We use the Hilbert-Schmidt independence criterion (HSIC) [*Gretton et al., 2005*], which is a kernel-based statistical measure of the independence between two sets of variables:

$$HSIC(K,L) = \frac{1}{(n-1)^2} tr(KHLH) \qquad \text{Eq. 3}$$

where:

$$K, H, L \in R^{n \times n}$$

in which H is the centering matrix $H = I - \frac{1}{n}11^T$, and K=$k\left(X^{(i)}, X^{(j)}\right), L = l(Y^{(i)}, Y^{(j)})$ are positive semidefinite kernel functions. For linear kernels, K=$k(X, Y) = XY^T$. An HSIC value of 0 implies independence. Other researchers *[Cristianini et al., 2006; Cortes et al., 2012; Kornblith et al., 2019a]* showed that HSIC can be made to be invariant to isotropic scaling by normalization. This normalized HSIC index is known as centered kernel alignment (CKA):

$$CKA(K, L) = \frac{HSIC(K,L)}{\sqrt{HSIC(K,K)HSIC(L,L)}} \qquad \text{Eq. 4}$$

In this study, we used the Centered Kernel Alignment (CKA) metric proposed by *Kornblith et al., [2019b]* with linear kernels to evaluate the similarities of layer representations in our trained networks. Specifically, we calculated the CKA between corresponding intermediate representations of the informed and uninformed networks. To assess the similarity between corresponding intermediate representations of model 1 and model 2 at layer i and j, we flattened the representations and let $X \in R^{n \times m1}$ and $Y \in R^{n \times m2}$ be the matrix of intermediate representations of model 1 and model 2 with m1 and m2 neurons for n examples. Then, we constructed the linear kernel matrices: K= XX^T and L= YY^T. Finally, we used equation [4] to compute the CKA metric. We compared similarities for all paired combinations of layers to explore how information flowed through both networks.

3- Results

The main goal of this study was to investigate the impact that "structure" has on the effective value of hydraulic conductivity (K_{eff}) of a binary heterogeneous medium. We examined this for multiple realizations of random fields that contain different percentages of the higher K material.

A key insight regarding this was presented by *Knight, [1992]* and *Indelman & Dagan [1993],* who showed that the spatial distribution of energy dissipation during steady-state flow can be used to define spatially distributed weights on K that can be used to compute K_{eff}. We first confirm this finding for the set of binary grids examined. Then, we show that the performance of a regression tree, trained to predict K_{eff} based only on the percent high K material, is improved by providing (reduced) information about the ED weights. Finally, we examine whether deep learning algorithms can predict K_{eff} with and without information regarding the ED weights. By comparing DL algorithms trained with and without access to energy dissipation information, we seek to understand the mechanism by which K_{eff} is inferred by the DL.

3-1- Analysis of the Effective Hydraulic Conductivity (K_{eff}) and High K Percentage

The steady-state flow problem, Figure (1), was solved for 3000 random realizations of a binary flow field for high conductivity mixtures ranging from 1 to 99%. K_{eff} was computed from the overall gradient applied over the domain and the steady-state flow through the domain. Figure 3 indicates how K_{eff} varies as a function of the percent high K material present in the realization. The parallel and series arrangements for each percent high K realization were calculated analytically and are seen to place limits on the ranges that K_{eff} values can take. The mean value of K_{eff} for each high K percentage is shown in the figure.

The plot demonstrates the nonlinear dependence of K_{eff} on percent high conductivity. At low percentages of high conductivity, K_{eff} is only minimally affected by the addition of more high K material and remains approximately equal to the conductivity of the lower K material. A nonlinear transition zone is seen to occur at approximately 40 to 70% high K, and the relationship becomes approximately linear above 70%. For a given percentage of high K, the maximum variance of K_{eff} occurs in the central transition zone.

These results illustrate the two related but different challenges for inferring K_{eff} from a binary grid: predicting mean K_{eff} as a function of the percent high K material; and predicting K_{eff} for a specific grid given knowledge regarding the percentage of high K material present.

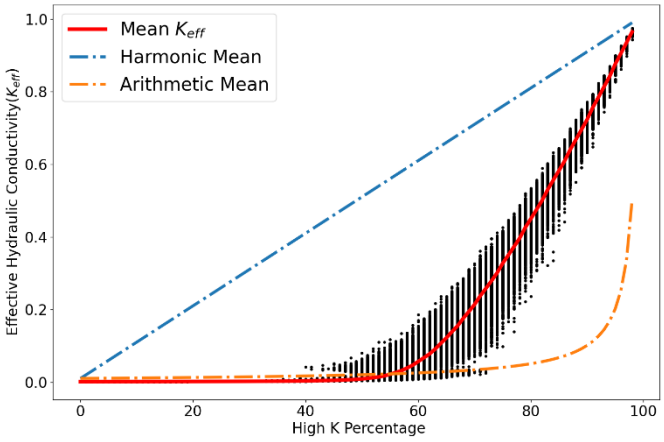


Figure 3 Keff distribution as a function of percent high K for medium K contrast condition

3-2- Analysis of the Energy-Dissipation Weighting Method to Explain the K_{eff}

Knight [1992] showed that the pattern of energy dissipation, calculated from the square of the gradient of the potential, can be used to determine an upscaled property like K_{eff}. This fact is

confirmed by our study (Figure 4). The energy dissipation approach can be thought of as computing a weighted average of the local K values on the grid that perfectly recovers the flow-based K_{eff}.

Despite the power of the energy dissipation approach, the weights are very difficult to identify visually. For example, the two grids are shown in Figures 5a and 5b both have 80% high conductivity material but have strikingly different K_{eff} values (0.53 and 0.24 respectively). The corresponding maps of the ED weights are shown in Figures 5c and 5d, illustrating that the grid with the lower K_{eff} has a much more localized pattern of ED weighing. While it might be tempting to attribute this localized weighting to the connected pattern of low K cells running vertically through Figure 5b, beyond this qualitative assessment it is essentially impossible to visually infer the values of the ED weights from the knowledge of the spatial organization of K. Of course, both the pattern of ED weights and their values can be computed readily by solving the steady-state flow problem, but then the value of K_{eff} can be determined directly and knowledge of the ED weights is superfluous. Accordingly, the ED weighting approach is best seen as a method for understanding spatial organization (e.g. *Ferre et al., 1998*), rather than a practical approach for inferring K_{eff} from a K grid.

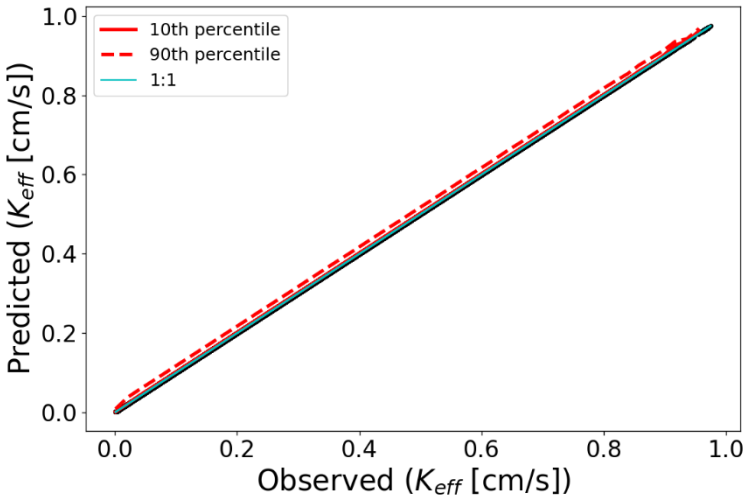
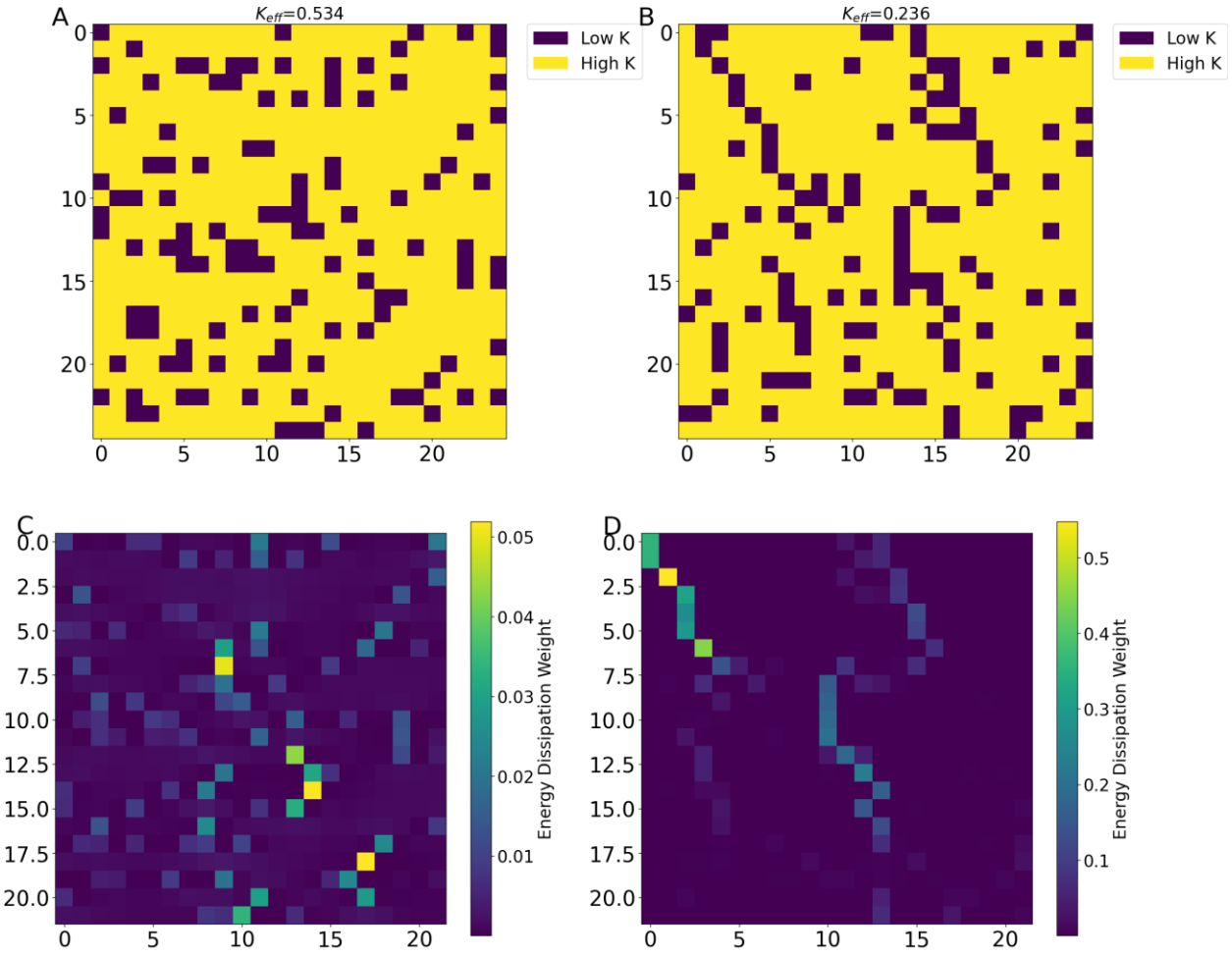


Figure 4) Keff estimation using energy dissipation method

By classifying the domain into high and low weight areas, we can see that the spatial structure of high-weight areas varies systematically with the percent high conductivity material. The paired images in Figure 6a show how the fractions of high energy cell relate to the corresponding ED maps for several K grids with different percentages of high K material. Figure 6b shows the expected fraction of high energy cells as a function of the threshold used to define high weight areas. There are two clear conclusions. First, the high energy area is restricted in a relatively small area for percent high conductivity conditions between approximately 50 and 80%. Second, the results are not highly sensitive to the choice of threshold. Finally, Figure 6c indicates a strong relationship between K_{eff} and the fraction of high energy dissipation cells (defined with a threshold of 95%), but with some interesting complications to that relationship in the range of 50 to 60% high conductivity material. These results suggest that information regarding the fraction of high energy cells may be informative for inferring K_{eff} for a given percent high conductivity material fraction, but that the relationship is likely to be complex.



503

Figure 5) Effects of structure on K_{eff} for the structures with the same percent high conductivity. A,B: Grid samples with percent high conductivity values of 80.

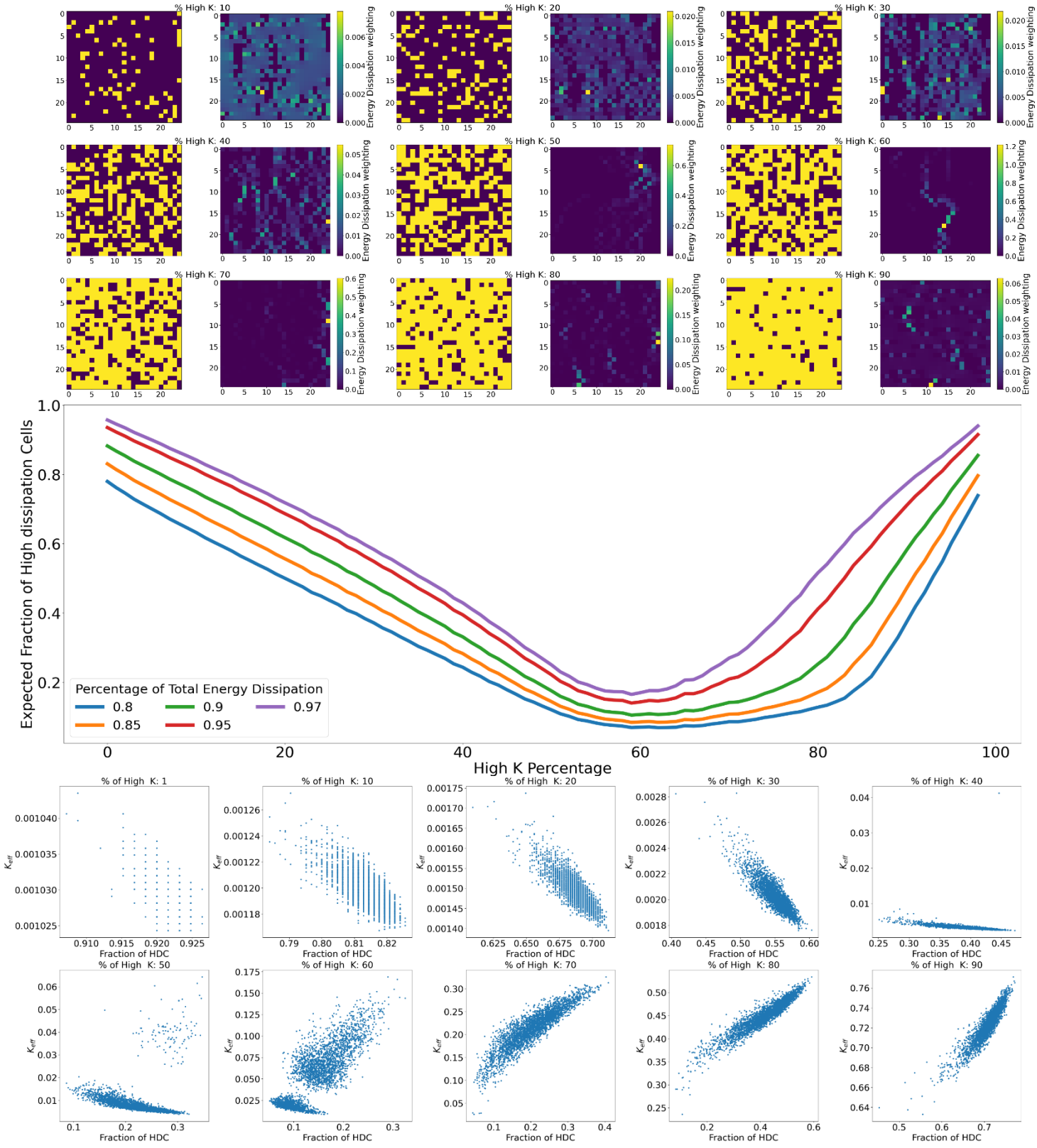


Figure 6) The energy dissipation pattern for different percent of high K materials. A: Grid samples and their corresponding energy dissipation weightings of high contributing cells as a function of percent of high K material. B: Average fraction of high energy dissipation cells as a function of the percent high K material, shown for definitions of "high energy dissipation". C: relationship between high energy cells and K_{eff} for different ranges of high k percentage.

505

3-3- Inferring K_{eff} with a Regression Tree Given Information on Energy Dissipation

Given only the gridded K values, it is a simple matter to determine the fraction of high conductivity cells. This property was used to infer K_{eff} by training a simple regression tree machine learning algorithm (Figure 7a), which achieved an RMSE of 0.0213 and R2 of 0.9942 when evaluated using the testing data. Note that this essentially finds only the *mean* value of K_{eff} at each percent high K [red line in Figure 3]. Further providing the number of high-weight cells (based on a threshold of 95%) to the regression tree algorithm improves the estimation of K_{eff}, achieving an RMSE of 0.0133 and an R2 of 0.9978 (Figure 7b). While the improvement in R2 may seem unimportant, there is a reduction in the error of Keff prediction for outliers, for which structure is more important (compare Figures 7a and 7b).

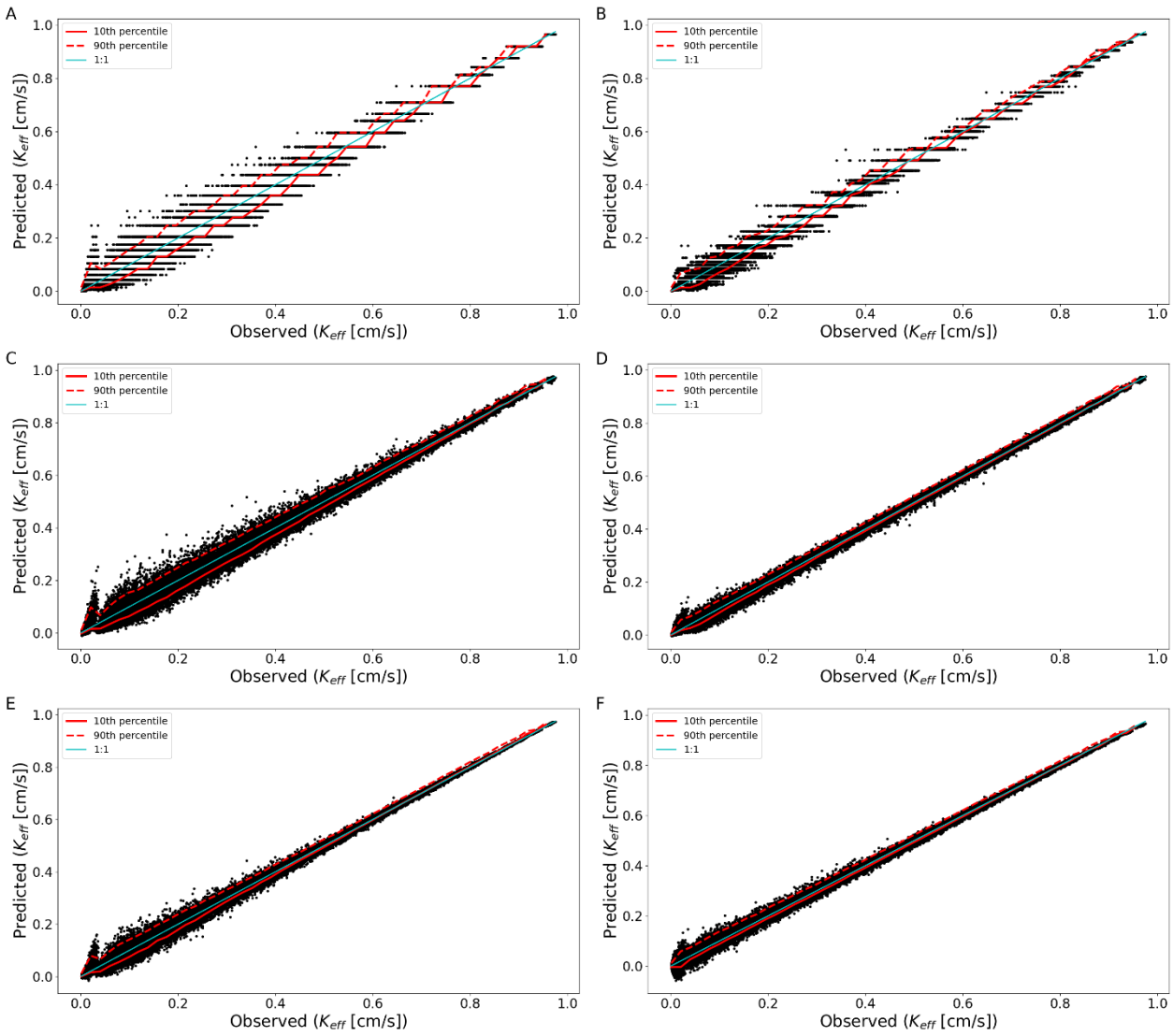


Figure 7) the testing performance of K_{eff} estimation using different methods. A: K_{eff} estimation using regression tree and only percent high K as input. B: K_{eff} estimation using regression tree and percent high K and fraction of high energy dissipation as inputs. C: K_{eff} estimation using vanilla CNN using only K grid as input. D: K_{eff} estimation using vanilla CNN using K grid and high energy dissipation cells mask as inputs. E: K_{eff} estimation using energy dissipation Uninformed UNET model. F: K_{eff} estimation using Informed UNET model with pre-training on energy dissipation.

3-4- Use of a Vanilla CNN to infer K_{eff} with and without ED Weights

For this part of the study, the binary K grid and a 0/1 map of high energy cells was provided as inputs to the CNN, and the target to be learned was K_{eff}. The vanilla CNN, provided with only the

K grid as input, performed as well (RMSE= 0.0171 and R2=0.9962) as the regression tree model that had been provided both the percent high conductivity material and the fraction of high energy cells (compare Figures 7c and 7b). Providing the binary high energy cell location map along with the K grid improves the performance (RMSE=0.0087 and R2=0.999); see Figure 7d.

Note that, unlike the RT, the CNN method provides estimates for K_{eff} on a continuum (not binned). Qualitatively, however, the CNN models provide relatively poor performance for low percent high conductivity material when given only the K grid as input (Figure 7c).

3-5- Inferring K_{eff} with UNET with and without ED weights

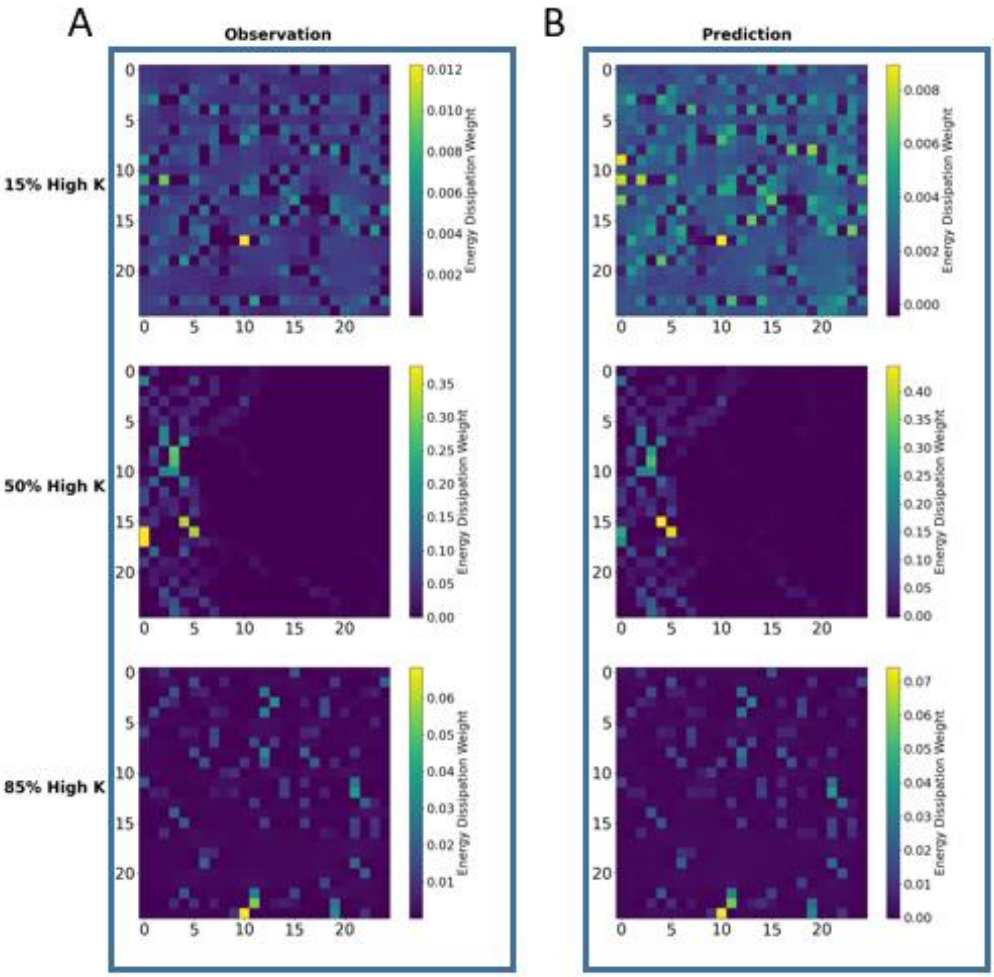
The uninformed UNET performs better (RMSE=0.0113 and R2 =0.9984) than the vanilla CNN when given only the K grid (compare Figures 7e and 7c). This indicates that the structure of the UNET enables it to learn something that allows it to achieve improved performance.

The results of the informed UNET (Figure 7f) are interesting. On the one hand, while it shows further improvement (RMSE=0.0106 and an R2=0.9986) over the uninformed UNET, it does not outperform the corresponding CNN where the high energy map was provided. So, the UNET structure seems to improve K_{eff} estimation while direct training on the ED weights only offers marginal performance improvement. On the other hand, although the informed UNET was provided information regarding the ED weights during training, its predictions of K_{eff} are made *based solely on the K grid*. In other words, training with knowledge of the ED distribution mainly affects the internal structure of the UNET. The result is that the "uninformed" and "informed" versions of UNET exhibit similar predictive performance (indicating equally informative

representations of the overall input-output mapping) while learning different internal representations of the mapping from gridded K to K_{eff}.

3-6- Inferring ED weights with UNET

The performance of the informed UNET for inferring EC is illustrated for some example grids in Figure 8. The correspondence between the ED weights predicted by the informed UNET and the value calculated directly from the flow model shows low RMSE (0.0069) and high R2 (0.9549) and the ability of the UNET to infer the fraction of high energy cells is likewise good (RMSE=0.04876 and R2=0.9832). However, there is still a considerable mismatch (Figure 9A). In particular, UNET consistently under-predicts the ED weights for cells that have very high actual weight, while consistent over-predicting the fraction of high energy cells for cases with intermediate percent high K (Figure 9B). From Figure 5b, these are the conditions that give rise to the most concentrated weighting. Taken together, these results suggest that the UNET has difficulty in inferring the ED weights when they are concentrated in highly localized areas (e.g. 60% high K material in Figure 5b).



558

559 Figure 8) Samples of energy dissipation weight distributions prediction for different ranges of percent of high K material. Panel A: Observation. Panel B: Predicted values.

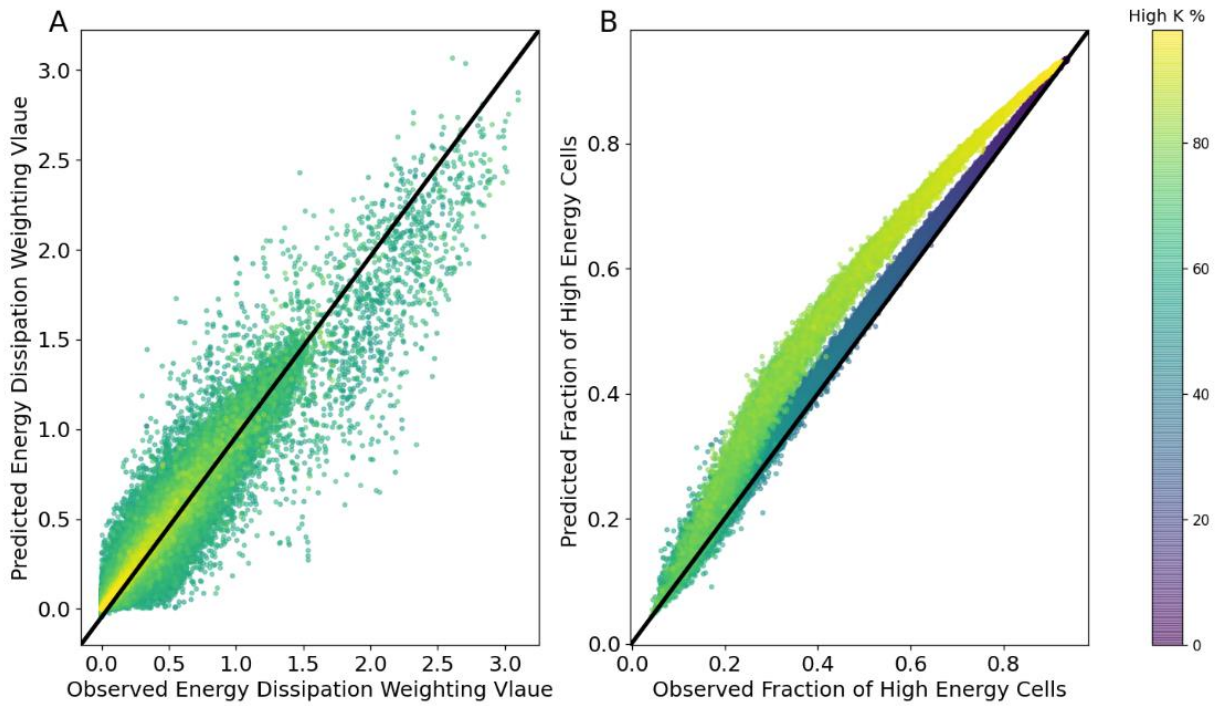


Figure 9) Performance of informed UNET model in energy dissipation estimation A: Energy dissipation weighting prediction for all grids. B: Fraction of high energy dissipation cells prediction performance as function of percent of high K material.

4- Discussion

Based on the results presented above, we discuss three issues. First, can ML/DL learn relationships that can predict both the trend and grid-specific variation of K_{eff} as a function of percent high K material? Second, does the use of increasingly more complex architectures lead to the improved inference of K_{eff}? Finally, can the DL algorithms make effective use of reduced information provided regarding ED weighting (here, the number of high energy cells), and can they use such information during training to infer patterns associated with ED weighting?

4-1- Dependence of the ED Weighting Distribution on the K Field

The K_{eff} associated with binary grids shows a highly nonlinear dependence on the percentage of high K material (Figure 3). Specifically, K_{eff} is closer to the arithmetic mean for materials with low to medium percentage of high K, while being approximately halfway between the arithmetic and harmonic means for materials with a higher percentage of high K. The variation in this trend is due to the influence of specific structural patterns in the spatial distribution of high and low K cells among grid realizations. The maximum degree of variability occurs for materials with intermediate percentages of high K values. In general, both the trend and the specific variations in K_{eff} are very well explained by ED-weighted averaging (Figure 4).

Given that the energy dissipation weights carry information regarding the impact of structure on the effective conductivity of a binary K field, we examined the nature of this weighting as a function of the percentage of high K material present in the medium. Specifically, we defined the minimum area that contains 95% of all of the ED weight, and classified the cells within this region as being 'high energy cells'.

At high and low percent high K conditions, the medium is nearly homogeneous, but the energy is distributed over ~75% of the domain (Figure 6b). The ED weighting is more highly constricted, residing in a smaller number of high energy cells, for 60% high K material grids. The restricted high K areas centered around 60% high K material tend to form localized regions within which most of the energy dissipation occurs, indicating the influence of structures that force the flow to occur through regions of relatively low K, leading to high energy loss. However, as the percentage of high K increases to 80%, the high weight areas become concentrated in a small number of unconnected regions, suggesting a different structural mechanism whereby flow is forced through a small number of low K cells, rather than being channeled through a continuous structure.

4-2- Comparison of Performance

By considering the ML/DL algorithms in order of increasing architectural complexity (DT, vanilla CNN, UNET) we can assess the value of increased algorithmic complexity and the value of providing reduced information about the ED weights. In summary (Table 1), performance improves with architectural complexity and when ED information is provided. In terms of RMSE and R^2, all of the ML/DL algorithms, including a simple decision tree provided with only the percent high K material, performed extremely well. So, the differences in performance are mainly due to their abilities to make case-specific use of structural (pattern) information, which manifests as variations in K_{eff} at any given percentage of high K material (Figure 7).

606 Table 1) Training, validation, and testing performance of all models

	Energy Dissipation Weight	DT (Only High K%)	DT (Only High K% + Number of High Energy Dissipation Cells)	Vanilla CNN	Vanilla CNN with High Energy Disispaiton Zones	No Knowledge	Knowledge
K_{eff} RMSE (Train)	0	0.0213869	0.01332592	0.0171827	0.00864406	0.00626774	0.00964671
K_{eff} RMSE (Val)	0	0.0215940	0.01367984	0.0172679	0.00866852	0.01129849	0.01077667
K_{eff} RMSE (Test)	0	0.0213212	0.01334975	0.0171188	0.00873331	0.01129849	0.01064088
K_{eff} R (Train)	1	0.9970817	0.99886803	0.9981229	0.99952405	0.99975328	0.99941291
K_{eff} R (Val)	1	0.9970250	0.99880711	0.9981039	0.99952134	0.99920198	0.99926396
K_{eff} R (Test)	1	0.9971003	0.99886423	0.9981378	0.99951423	0.99918991	0.99928351
Energy Dissipation RMSE (Train)	NaN	NaN	NaN	NaN	NaN	0.02693278	0.00248980
Energy Dissipation RMSE (Val)	NaN	NaN	NaN	NaN	NaN	0.02703661	0.00548620
Energy Dissipation RMSE (Test)	NaN	NaN	NaN	NaN	NaN	0.03300000	0.00695936
Energy Dissipation R (Train)	NaN	NaN	NaN	NaN	NaN	-0.04757823	0.99531359
Energy Dissipation R (Val)	NaN	NaN	NaN	NaN	NaN	-0.04673303	0.97724645
Energy Dissipation R (Test)	NaN	NaN	NaN	NaN	NaN	-0.05657500	0.97722907

607

608 For all methods, the performance was poorest when K_{eff} values are low (Figure 7). The
609 performance was also relatively poor for intermediate percentage levels of high K (Figure 10).
610 That is, the methods had the most difficulty when localized structures act to impede flow, whether
611 those structures are organized as a continuous region (intermediate high K percentage) or as
612 isolated blocks of low K material (low K_{eff}).

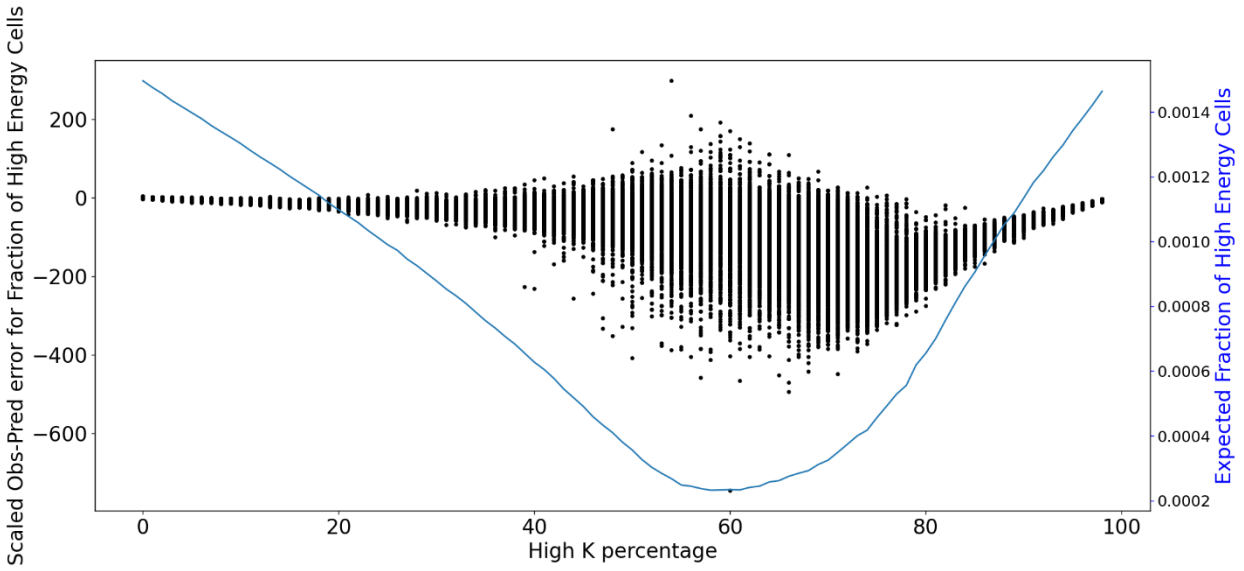


Figure10) Difference between inferred and actual fraction of high K cells for each grid. To compare the errors of grids at each high k percentage, the values of left y axis is scaled by average of actual number of high k cells at each k percentage. The fraction of high K cells for a 95% threshold is presented by blue line.

4-3- Hidden Layer Representation Analysis

The superior performance of the informed UNET is notable because it does not require that the flow problem be solved to make predictions for the testing set. Specifically, once trained with ED weight information (requiring solving the flow problem during testing and validation), the UNET algorithm uses the learned relationships to infer the values of the ED weights for the test samples and combines this with the K grid to infer K_{eff}.

The performance of the uninformed UNET, for which ED weight information was never presented, so the flow problem never had to be solved, is comparable to that of the trained UNET. Given that the ED weights are thought to represent a key mechanism linking the K grid to the value of K_{eff}, this raises the question of whether the uninformed UNET is somehow inferring information regarding the distribution of ED without being explicitly provided with such information during training.

For the informed UNET, the output layer of the lower branch, which is concatenated with the K grid before the final step of inferring K_{eff}, represents the ED weight distribution. Examining the corresponding layer of the uninformed UNET shows no correlation with the true ED weights. However, a more advanced analysis, based on computing the centered kernel alignment similarity (CKA) [*Kornblith et al., 2019*], provides a more complete picture of the information flows through the informed and uninformed UNETs. These results are visualized as a similarity matrix (Figure 11). The output of each layer of the informed model is compared to other layers of the uninformed model to examine the degree of similarity between them while accounting for the presence of invertible linear transformations. A similarity value of zero between two layers indicates that their representations are *not* invertible linear transformations of each other while a similarity value of 1 indicates that the two layers are equivalent up to a linear transformation.

We first compared the results for the informed UNET with that of an untrained network with random initial weights and the same architecture (Figure 11a). The values on the diagonal (representing the same layer in the two networks) have high CKA similarity for the first three layers; this makes sense given that both networks are being fed the same inputs. However, the similarity begins to diminish beyond that point; they show very strong dissimilarity at the output layer, where the informed UNET is constrained to predict values that correspond to the ED weights. They also differ strongly at the final dense layer because the untrained network did a poor job of inferring K_{eff}.

Comparing the informed and uninformed UNETs gave striking results (Figure 11b). Namely, layer similarity remains high for all layers *except* the output layer, where the informed UNET is required to predict values that correspond to the ED weights. Further, the final dense layer is also

highly similar, reflecting the near-identical skill in predicting K_{eff} achieved by both the informed and uninformed UNET.

In general, these results and patterns of similarity are consistent with the findings of *Kornblith et al., [2019b]* and *Thompson et al. [2019]*. They show that there can be many possible intermediate architectural solutions to achieve the same task, but that the representations learned for the layers closer to the inputs and the outputs tend to be similar. We interpret this to mean that the untrained UNET can "learn" some useful information that is related to the ED weights directly from the K grids. This information is not a direct map of actual ED weights. So, when required to produce such a map (training under-informed conditions), the UNET learns an intermediate relationship that can provide this map and to the user. It then uses the ED distribution to infer K_{eff}. However, when not required to produce an ED map (training under uninformed conditions), the UNET does not develop a layer to translate the information to a user-readable ED map. Rather, the latent information about the ED weights propagates through the UNET, with an associated change in the final dense layer to produce high-quality inferences of K_{eff}.

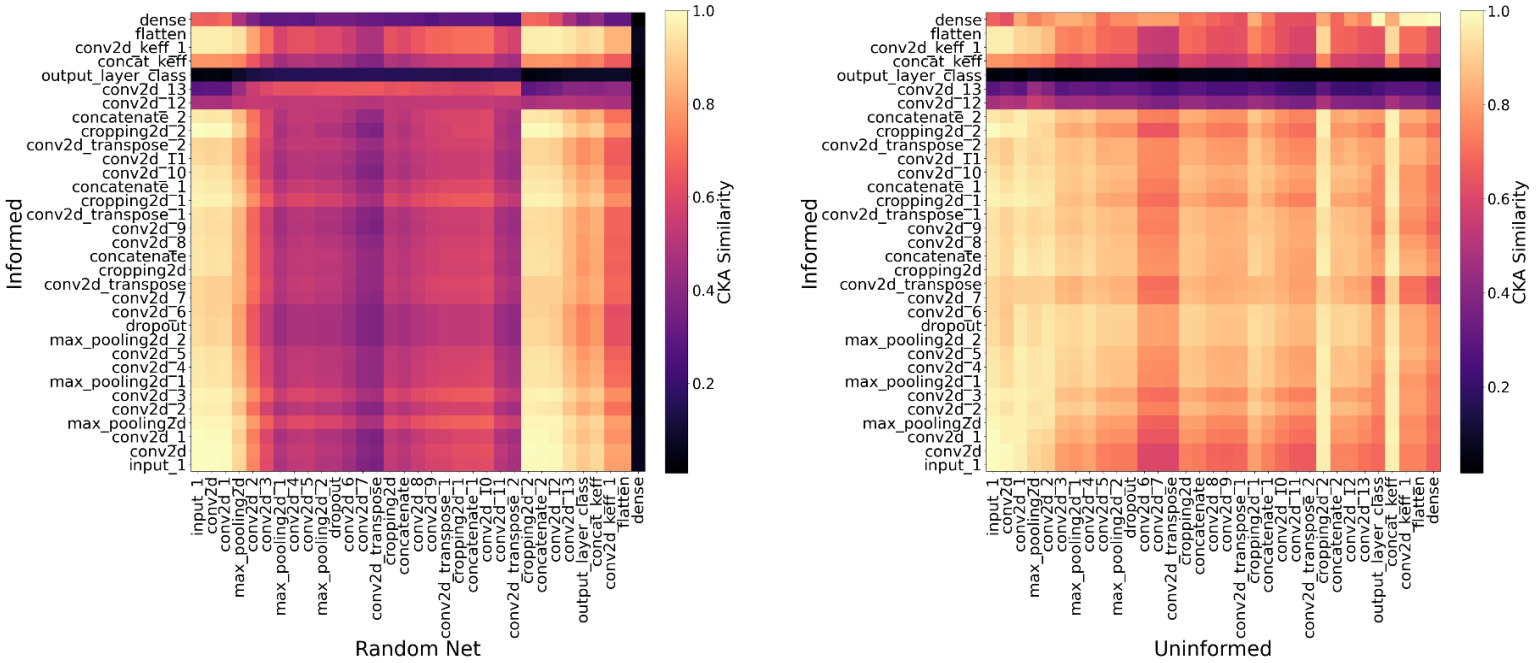


Figure 12) CKA similarity matrix between A) Informed Unet and untrained Unet b) Informed Unet and Uninformed Unet

The CKA analysis cannot uncover relationships between networks in the presence of invertible nonlinear transformations. To examine this, we sequentially swapped the weights of the uninformed UNET with those of the informed UNET. Specifically, at each step of this analysis (i.e., for each layer), we used the weights of the uninformed model for the preceding layers while maintaining the informed UNET weights for the succeeding layers. The results (Figure 13) are presented with the deepest layer at the top left, progressing along each row and then downward to the final layer at the bottom right. There are strong linear correlations between the observed K_{eff} and that predicted with the 'swapped' network until the substitutions reach the conv2d_12 layer. This is consistent with the high CKA representation similarity to this layer (Figure 12). There is a strongly nonlinear relationship for conv2d_13, which corresponds with a low CKA value at this layer. In the final layer (i.e, output layer), we see a strong negative linear correlation between the

output of the mixed structure model and that of the informed model. This pattern is consistent with the high CKA value observed in Figure 12 and suggests that an orthogonal transformation between the weights was necessary to overcome the changes applied in the deeper layers and recover the correct K_{eff} values. This analysis suggests that both the informed and uninformed UNET are implementing similar computational processes, ostensibly extracting information corresponding to the ED distribution from the K grid, but representing it differently in n-d dimensional space. Further, that the user-imposed requirement to produce a readable ED map results in a nonlinear transformation that must be compensated in later layers to produce accurate inferred K_{eff} values.

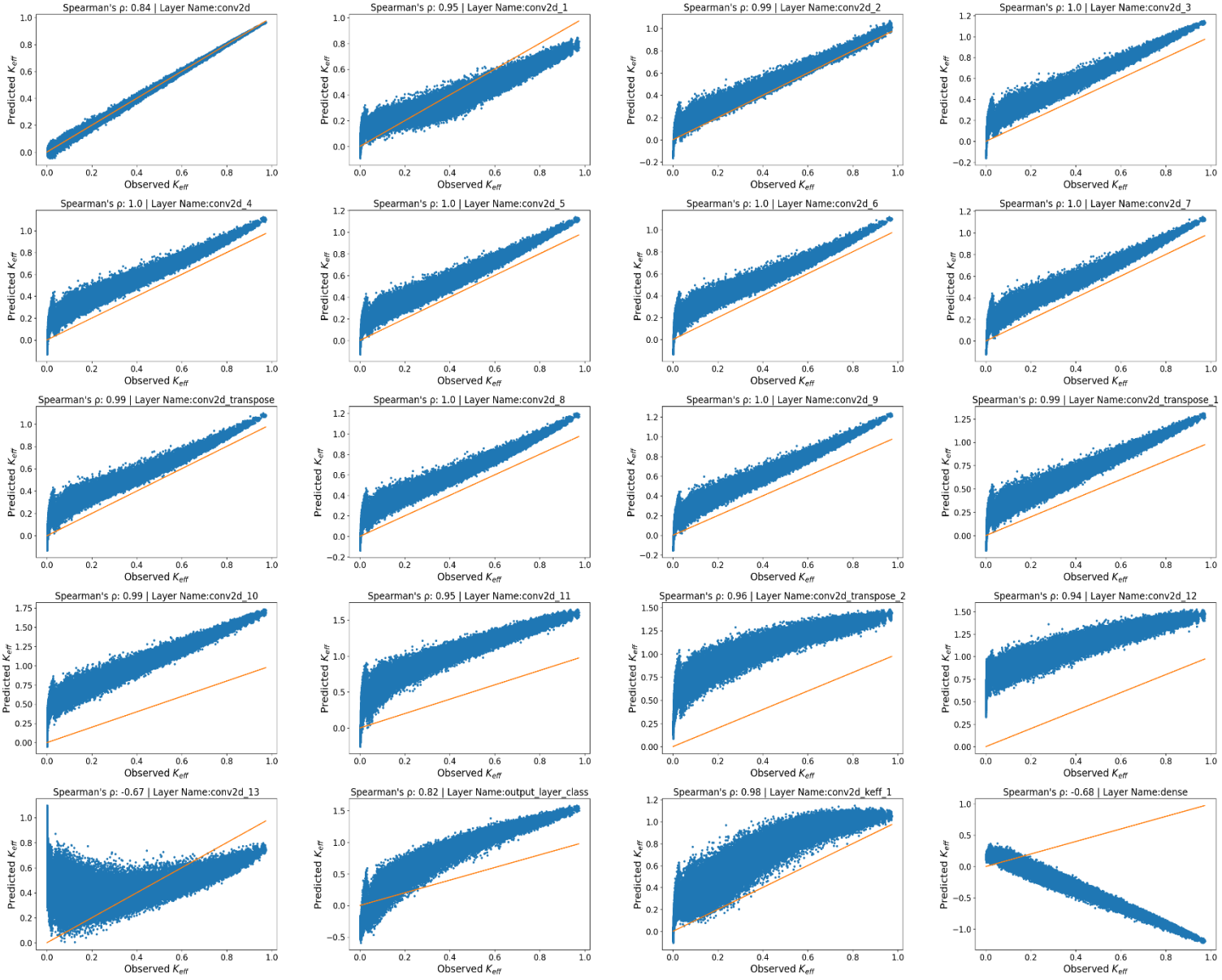


Figure 13) Correlation between True Keff and the output of Unet model built up by sequential substitution of Informed model weights with Uninformed Unet collectively.

5-Conclusions

We have investigated the ability of ML and DL algorithms to infer the effective hydraulic conductivity of binary K grids. All of the ML/DL methods were able to infer K_{eff} with extremely

high accuracy (R2 > 0.99) when provided with only the binary grid. But, there was some improvement in identifying the K_{eff} of outlier realizations, those most strongly affected by structure, with increasing algorithmic complexity, progressing from a decision tree, to a vanilla CNN, to a UNET.

Relying on previous work that showed the value of energy dissipation weighting for understanding and inferring K_{eff}, we examined whether providing such information improved the ML/DL performance. While adding information derived from the ED distribution improved the performance of each algorithm, the improvement was similar to that realized by increasing the algorithmic complexity.

The UNET architecture could be trained to infer the ED weighting from the K grid. This finding was supported by a similarity analysis of the hidden layers of UNETs with and without ED information provided. The accuracy of the inferred ED weights was lower when the energy dissipation weights were concentrated into small areas; i.e., the UNET was better able to infer the impacts of diffuse structures than highly localized structures. This finding may be due to the relatively small number of realizations that showed strong structural control in our sample set, suggesting that future work should examine this possibility.

While the UNET extracted the relevant ED weight information from the K grids, it only translated this information to a user-readable map if forced to do so. This may have other implications for the use of ML/DL techniques in subsurface hydrology. For example, ML/DL algorithms may be able to implicitly infer head distribution information 'naturally' if they are trained to predict streamflow; but the head distributions may not be available to the user unless the algorithms are specifically designed to produce them. This may be an important consideration if ML/DL

algorithms are applied to models with multiple calibration data types or if the models will be used for multi-objective decision support.

6-References

Afzaal, H., Farooque, A. A., Abbas, F., Acharya, B., & Esau, T. (2019). Groundwater Estimation from Major Physical Hydrology Components Using Artificial Neural Networks and Deep Learning. Water, 12(1), 5. https://doi.org/10.3390/w12010005

Ahuja, L. R., Ma, L., & Green, T. R. (2010). Effective Soil Properties of Heterogeneous Areas For Modeling Infiltration and Redistribution. Soil Science Society of America Journal, 74(5), 1469–1482. https://doi.org/10.2136/sssaj2010.0073

Aliyari, F., Bailey, R. T., Tasdighi, A., Dozier, A., Arabi, M., & Zeiler, K. (2019). Coupled SWAT-MODFLOW model for large-scale mixed agro-urban river basins. Environmental Modelling and Software, 115, 200–210. https://doi.org/10.1016/j.envsoft.2019.02.014

Allison Marier, Lauren E.W. Olsho, William Rhodes, & William D. Spector. (2016). Improving prediction of fall risk among nursing home residents using electronic medical records. Journal of the American Medical Informatics Association, 23(2), 8–9. https://doi.org/10.1093/JAMIA

Ambegaokar, V., Halperin, B. I., & Langer, J. S. (1971). Hopping conductivity in disordered systems. Physical Review B, 4(8), 2612–2620. https://doi.org/10.1103/PhysRevB.4.2612

Apley, D. W., & Zhu, J. (2016). Visualizing the Effects of Predictor Variables in Black Box Supervised Learning Models. http://arxiv.org/abs/1612.08468

Assem, H., Ghariba, S., Makrai, G., Johnston, P., Gill, L., & Pilla, F. (2017). Urban Water Flow and Water Level Prediction Based on Deep Learning. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 10536 LNAI, 317–329. https://doi.org/10.1007/978-3-319-71273-4_26

Berkowitz, B., & Balberg, I. (1993). Percolation theory and its application to groundwater hydrology. Water Resources Research, 29(4), 775–794. https://doi.org/10.1029/92WR02707

Brock, A., Lim, T., Ritchie, J. M., & Weston, N. (2017). FreezeOut: Accelerate Training by Progressively Freezing Layers. http://arxiv.org/abs/1706.04983

Canchumuni, S. W. A., Emerick, A. A., & Pacheco, M. A. C. (2018). Towards a Robust Parameterization for Conditioning Facies

737 Models Using Deep Variational Autoencoders and Ensemble Smoother. Computers and Geosciences, 128, 87–102.
738 https://doi.org/10.1016/j.cageo.2019.04.006

739 Cardwell, W., & Parsons, R. (1945). Average permeability of heterogeneous oil sands. Transactions of AIME, 160.

740 Chakraborty, S., Tomsett, R., Raghavendra, R., Harborne, D., Alzantot, M., Cerutti, F., Srivastava, M., Preece, A., Julier, S., Rao,
741 R. M., Kelley, T. D., Braines, D., Sensoy, M., Willis, C. J., & Gurram, P. (2018). Interpretability of deep learning models:
742 A survey of results. 2017 IEEE SmartWorld Ubiquitous Intelligence and Computing, Advanced and Trusted Computed,
743 Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People and Smart City Innovation,
744 SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI 2017 - , 1–6. https://doi.org/10.1109/UIC-ATC.2017.8397411

745 Chan, S., & Elsheikh, A. H. (2017). Parametrization and generation of geological models with generative adversarial networks.
746 http://arxiv.org/abs/1708.01810

747 Cheng, Q., Xu, Q., Cheng, X., Yu, S., Wang, Z., Sun, Y., Yan, X., & Jones, S. B. (2019). In-situ estimation of unsaturated hydraulic
748 conductivity in freezing soil using improved field data and inverse numerical modeling. Agricultural and Forest
749 Meteorology, 279, 107746. https://doi.org/10.1016/j.agrformet.2019.107746

750 Colecchio, I., Boschan, A., Otero, A. D., & Noetinger, B. (2020). On the multiscale characterization of effective hydraulic
751 conductivity in random heterogeneous media: A historical survey and some new perspectives. Advances in Water Resources,
752 140, 103594. https://doi.org/10.1016/j.advwatres.2020.103594

753 Cortes, C., Mohri, M., & Rostamizadeh, A. (2012). Algorithms for Learning Kernels Based on Centered Alignment. In Journal of
754 Machine Learning Research (Vol. 13). https://dl.acm.org/doi/abs/10.5555/2503308.2188413

755 Coutinho de Oliveira, L. F., Rezende de Souza, G., Vilela Corrêa, F., & Ribeiro Mota e Silva, J. (2020). Parameter estimation of
756 soil hydraulic characteristics by inverse modeling of the analytical equation for unsaturated subsurface water flow. Journal
757 of Hydroinformatics. https://doi.org/10.2166/hydro.2020.015

758 Cristianini, N., Kandola, J., Elisseeff, A., & Shawe-Taylor, J. (2006). On Kernel Target Alignment BT - Innovations in Machine
759 Learning: Theory and Applications (D. E. Holmes & L. C. Jain (eds.); pp. 205–256). Springer Berlin Heidelberg.
760 https://doi.org/10.1007/3-540-33486-6_8

761 Demiray, B. Z., Xiang, Z., Sermet, Y., & Demir, I. (2020). A Comprehensive Review of Deep Learning Applications in Hydrology
762 and Water Resources. June. https://doi.org/10.31223/osf.io/xs36g

763 Desbarats, A. J., & Srivastava, R. M. (1991). Geostatistical characterization of groundwater flow parameters in a simulated aquifer.
764 Water Resources Research, 27(5), 687–698. https://doi.org/10.1029/90WR02705

765 Dlubac, K., Knight, R., Song, Y. Q., Bachman, N., Grau, B., Cannia, J., & Williams, J. (2013). Use of NMR logging to obtain
766 estimates of hydraulic conductivity in the High Plains aquifer, Nebraska, USA. Water Resources Research, 49(4), 1871–
767 1886. https://doi.org/10.1002/wrcr.20151

768 Durlofsky, L. J. (1992). Representation of grid block permeability in coarse scale models of randomly heterogeneous porous media.
769 Water Resources Research, 28(7), 1791–1800. https://doi.org/10.1029/92WR00541

770 Ferré, P. A., Knight, J. H., Rudolph, D. L., & Kachanoski, R. G. (1998). The sample areas of conventional and alternative time
771 domain reflectometry probes. Water Resources Research, 34(11), 2971–2979. https://doi.org/10.1029/98WR02093

772 Green, T. R., Dunn, G. H., Erskine, R. H., Salas, J. D., & Ahuja, L. R. (2009). Fractal Analyses of Steady Infiltration and Terrain
773 on an Undulating Agricultural Field. Vadose Zone Journal, 8(2), 310–320. https://doi.org/10.2136/vzj2008.0021

774 Gretton, A., Bousquet, O., Smola, A., & Scḧlkopf, B. (2005). Measuring statistical dependence with Hilbert-Schmidt norms.
775 Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in
776 Bioinformatics), 3734 LNAI, 63–77. https://doi.org/10.1007/11564089_7

777 Hassanzadegan, A., Cacace, M., Sippel, J., & Scheck-Wenderoth, M. (2016). The application of inverse modeling in characterizing
778 hydraulic conductivity beneath the city of Berlin, Germany. Environmental Earth Sciences, 75(20), 1–17.
779 https://doi.org/10.1007/s12665-016-6107-2

780 Hertrich, M. (2008). Imaging of groundwater with nuclear magnetic resonance. In Progress in Nuclear Magnetic Resonance
781 Spectroscopy (Vol. 53, Issue 4, pp. 227–248). https://doi.org/10.1016/j.pnmrs.2008.01.002

782 Hunt, Allen, Robert Ewing, and B. G. (2014). Percolation Theory for Flow in Porous Media.
783 https://books.google.com/books?hl=en&lr=&id=KhW6BQAAQBAJ&oi=fnd&pg=PR3&ots=b559LK9bRY&sig=fhItT0U
784 AMBZQNDrW7NUuidkahR0#v=onepage&q&f=false

785 Hunt, A. G., & Sahimi, M. (2017). Flow, Transport, and Reaction in Porous Media: Percolation Scaling, Critical-Path Analysis,
786 and Effective Medium Approximation. In Reviews of Geophysics (Vol. 55, Issue 4, pp. 993–1078). Blackwell Publishing
787 Ltd. https://doi.org/10.1002/2017RG000558

788 Indelman, P., & Dagan, G. (1993). Upscaling of permeability of anisotropic heterogeneous formations: 2. General structure and
789 small perturbation analysis. Water Resources Research, 29(4), 925–933. https://doi.org/10.1029/92WR02447

790 Journel, A. G., Deutsch, C., & Desbarats, A. J. (1986). Power averaging for block effective permeability. Society of Petroleum
791 Engineers - SPE California Regional Meeting, CRM 1986, 329–334. https://doi.org/10.2118/15128-ms

792 Katz, A. J., & Thompson, A. H. (1985). Fractal sandstone pores: Implications for conductivity and pore formation. Physical Review

Letters, 54(12), 1325–1328. https://doi.org/10.1103/PhysRevLett.54.1325

King, P. R. (1989). The use of renormalization for calculating effective permeability. Transport in Porous Media, 4(1), 37–58. https://doi.org/10.1007/BF00134741

King, P. R., & Neuwelier, I. (2002). Probability upscaling. Computational Geosciences, 6(1), 101–114. https://doi.org/10.1023/A:1016533230647

Kingma, D. P., & Ba, J. L. (2015). Adam: A method for stochastic optimization. 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings. https://arxiv.org/abs/1412.6980

Knight, J. H. (1992). Sensitivity of time domain reflectometry measurements to lateral variations in soil water content. Water Resources Research, 28(9), 2345–2352. https://doi.org/10.1029/92WR00747

Knudby, C., Carrera, J., Bumgardner, J. D., & Fogg, G. E. (2006). Binary upscaling - The role of connectivity and a new formula. Advances in Water Resources, 29(4), 590–604. https://doi.org/10.1016/j.advwatres.2005.07.002

Kornblith, S., Norouzi, M., Lee, H., & Hinton, G. (2019a). Similarity of neural network representations revisited. 36th International Conference on Machine Learning, ICML 2019, 2019-June, 6156–6175.

Kornblith, S., Norouzi, M., Lee, H., & Hinton, G. (2019b). Similarity of neural network representations revisited. 36th International Conference on Machine Learning, ICML 2019, 2019-June, 6156–6175. http://arxiv.org/abs/1905.00414

Kotlar, A. M., Varvaris, I., Jong van Lier, Q., Wollesen de Jonge, L., Møldrup, P., & Iversen, B. V. (2019). Soil Hydraulic Properties Determined by Inverse Modeling of Drip Infiltrometer Experiments Extended with Pedotransfer Functions. Vadose Zone Journal, 18(1), 1–11. https://doi.org/10.2136/vzj2018.12.0215

Kratzert, F., Klotz, D., Brandstetter, J., Hoedt, P.-J., Nearing, G., & Hochreiter, S. (2019). Using LSTMs for climate change assessment studies on droughts and floods. http://arxiv.org/abs/1911.03941

Lai, J., & Ren, L. (2016). Estimation of effective hydraulic parameters in heterogeneous soils at field scale. Geoderma, 264, 28–41. https://doi.org/10.1016/j.geoderma.2015.09.013

Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X., & Pietikäinen, M. (2020). Deep Learning for Generic Object Detection: A Survey. International Journal of Computer Vision, 128(2), 261–318. https://doi.org/10.1007/s11263-019-01247-4

Luo, C., Wu, D., & Wu, D. (2017). A deep learning approach for credit scoring using credit default swaps. Engineering Applications of Artificial Intelligence, 65, 465–470. https://doi.org/10.1016/j.engappai.2016.12.002

Mai, F., Tian, S., Lee, C., & Ma, L. (2019). Deep learning models for bankruptcy prediction using textual disclosures. European

Journal of Operational Research, 274(2), 743–758. https://doi.org/10.1016/j.ejor.2018.10.024

Malick, K. . (1995). Boundary effects in the successive upscaling of absolute permeability. Stanford University, CA.

Masihi, M., Gago, P. A., & King, P. R. (2016). Estimation of the Effective Permeability of Heterogeneous Porous Media by Using Percolation Concepts. Transport in Porous Media, 114(1), 169–199. https://doi.org/10.1007/s11242-016-0732-9

Matheron, G. (1965). Les variables régionalisées et leur estimation. Une application de la théorie des fonctions aléatoires aux sciences de la nature. In Masson et Cie (Issue 1).

Mo, S., Zabaras, N., Shi, X., & Wu, J. (2020). Integration of Adversarial Autoencoders With Residual Dense Convolutional Networks for Estimation of Non-Gaussian Hydraulic Conductivities. Water Resources Research, 56(2), 1–24. https://doi.org/10.1029/2019WR026082

Mosser, L., Dubrule, O., & Blunt, M. J. (2017). Reconstruction of three-dimensional porous media using generative adversarial neural networks. Physical Review E, 96(4), 43309. https://doi.org/10.1103/PhysRevE.96.043309

Nearing, G., Sampson, A. K., Kratzert, F., & Frame, J. (n.d.). Post-Processing a Conceptual Rainfall-Runoff Model with an LSTM. https://doi.org/10.31223/OSF.IO/53TE4

O'Mahony, N., Campbell, S., Carvalho, A., Harapanahalli, S., Hernandez, G. V., Krpalkova, L., Riordan, D., & Walsh, J. (2020). Deep Learning vs. Traditional Computer Vision. Advances in Intelligent Systems and Computing, 943, 128–144. https://doi.org/10.1007/978-3-030-17795-9_10

Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. Journal of Computational Physics, 378, 686–707. https://doi.org/10.1016/j.jcp.2018.10.045

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 9351, 234–241. https://doi.org/10.1007/978-3-319-24574-4_28

Sánchez-Vila, X., Girardi, J. P., & Carrera, J. (1995). A Synthesis of Approaches to Upscaling of Hydraulic Conductivities. Water Resources Research, 31(4), 867–882. https://doi.org/10.1029/94WR02754

Shamsudduha, M., Zahid, A., & Burgess, W. G. (2019). Security of deep groundwater against arsenic contamination in the Bengal Aquifer System: a numerical modeling study in southeast Bangladesh. Sustainable Water Resources Management, 5(3), 1073–1087. https://doi.org/10.1007/s40899-018-0275-z

Slater, L. (2007a). Near surface electrical characterization of hydraulic conductivity: From petrophysical properties to aquifer

geometries - A review. Surveys in Geophysics, 28(2–3), 169–197. https://doi.org/10.1007/s10712-007-9022-y

Slater, L. (2007b). Near surface electrical characterization of hydraulic conductivity: From petrophysical properties to aquifer geometries - A review. Surveys in Geophysics, 28(2–3), 169–197. https://doi.org/10.1007/s10712-007-9022-y

Srisutthiyakorn, N. (2016). Deep learning methods for predicting permeability from 2-D/3-D binary segmented images. SEG Technical Program Expanded Abstracts, 35, 3042–3046. https://doi.org/10.1190/segam2016-13972613.1

Tartakovsky, A. M., Marrero, C. O., Perdikaris, P., Tartakovsky, G. D., & Barajas-Solano, D. (2020). Physics-Informed Deep Neural Networks for Learning Parameters and Constitutive Relationships in Subsurface Flow Problems. Water Resources Research, 56(5), 1–16. https://doi.org/10.1029/2019WR026731

Thompson, J., Bengio, Y., & Schoenwiesner, M. (2019, December 4). The effect of task and training on intermediate representations in convolutional neural networks revealed with modified RV similarity analysis. https://doi.org/10.32470/ccn.2019.1300-0

Vinay Ambegaokar, B. I. H. (1971). HOPPING CONDUC TIVITY IN DISORDE RE D S YS TE MS.

Voulodimos, A., Doulamis, N., Doulamis, A., & Protopapadakis, E. (2018). Deep Learning for Computer Vision: A Brief Review. https://doi.org/10.1155/2018/7068349

Vrugt, J. A., Stauffer, P. H., Wöhling, T., Robinson, B. A., & Vesselinov, V. V. (2008). Inverse Modeling of Subsurface Flow and Transport Properties: A Review with New Developments. Vadose Zone Journal, 7(2), 843–864. https://doi.org/10.2136/vzj2007.0078

Wang, N., Zhang, D., Chang, H., & Li, H. (2020). Deep learning of subsurface flow via theory-guided neural network. Journal of Hydrology, 584(January), 124700. https://doi.org/10.1016/j.jhydrol.2020.124700

Wen, X. H., & Gómez-Hernández, J. J. (1996). Upscaling hydraulic conductivities in heterogeneous media: An overview. In Journal of Hydrology (Vol. 183, Issues 1–2, pp. ix–xxxii). https://doi.org/10.1016/s0022-1694(96)80030-8

Wu, J., Yin, X., & Xiao, H. (2018). Seeing permeability from images: fast prediction with convolutional neural networks. Science Bulletin, 63(18), 1215–1222. https://doi.org/10.1016/j.scib.2018.08.006

Yadav, V., & Bethard, S. (2019). A Survey on Recent Advances in Named Entity Recognition from Deep Learning models. http://arxiv.org/abs/1910.11470

Zhao, Y., & Bethard, S. (2020). How does BERT's attention change when you fine-tune? An analysis methodology and a case study in negation scope. https://github.com/

Zhou, Z., Shi, L., & Zha, Y. (2020). Seeing macro-dispersivity from hydraulic conductivity field with convolutional neural network.

Advances in Water Resources, 138(January), 103545. https://doi.org/10.1016/j.advwatres.2020.103545

Zhu, J., & Mohanty, B. P. (2002). Upscaling of soil hydraulic properties for steady state evaporation and infiltration. Water Resources Research, 38(9), 17-1-17–13. https://doi.org/10.1029/2001wr000704

Zoph, B., Yuret, D., May, J., & Knight, K. (2016). Transfer Learning for Low-Resource Neural Machine Translation. EMNLP 2016 - Conference on Empirical Methods in Natural Language Processing, Proceedings, 1568–1575. http://arxiv.org/abs/1604.02201

7-Data Availability

The data that supports the findings of this study are openly available in the Univerity of Arizon research data repository at 10.25422/azu.data.13324796.

Appendix

Vanilla CNN
3*3 conv. 8-same padding-stride 1-Relu *2 2*2 Maxpooling stride 1
3*3 conv. 16-same padding-stride 1-Relu *2 2*2 Maxpooling stride 1
3*3 conv. 32-same padding-stride 1-Relu *2 2*2 Maxpooling stride 1
3*3 conv. 16-same padding-stride 1-Relu *2 2*2 Maxpooling stride 1
3*3 conv. 8-same padding-stride 1-Relu *2 2*2 Maxpooling stride 1
3*3 conv. 4-same padding-stride 1-Relu *2 2*2 Maxpooling stride 1
Flatten 1 Dense-Linear

Modified Unet Model
3*3 conv. 16-same padding-stride 1-Relu *2 2*2 Maxpooling stride 2
3*3 conv. 32-same padding-stride 1-Relu *2 2*2 Maxpooling stride 2
3*3 conv. 64-same padding-stride 1-Relu *2 2*2 Maxpooling stride 2 Dropout 0.64
3*3 conv. 128-same padding-stride 1-Relu *2
2*2 Conv2DTranspose. 64-same padding-stride 2-No activation *1 Cropping Concatenation 3*3 conv. 64-same padding-stride 1-Relu *2
2*2 Conv2DTranspose. 32-same padding-stride 2-No activation *1 Cropping Concatenation 3*3 conv. 32-same padding-stride 1-Relu *2
2*2 Conv2DTranspose. 16-same padding-stride 2-No activation *1 Cropping Concatenation 3*3 conv. 16-same padding-stride 1-Relu *2
1*1 conv. 1-same padding-stride 1-No activation *1
Concatenation 3*3 conv. 10-same padding-stride 1-Tanh *1 Flatten 1 Dense-Linear

Table A1 Deep learning structure parameters. A: vanilla CNN Structure.B: UNET model structure.