



Novel physics informed-neural networks for estimation of hydraulic conductivity of green infrastructure as a performance metric by solving Richards–Richardson PDE

Mahmoud Elkhadrawi¹ · Carla Ng² · Daniel J. Bain³ · Emelia E. Sargent³ · Emma V. Stearsman³ · Kimberly A. Gray⁴ · Murat Akcakaya¹

Received: 24 July 2023 / Accepted: 7 December 2023 / Published online: 10 January 2024
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2024

Abstract

Green infrastructure (GI) is an ecologically informed approach to stormwater management that is potentially sustainable and effective. Infiltration-based GI systems, including rain gardens, permeable pavements, green roofs infiltrate surface water and stormwater run-off to recharge ground water systems. However, these systems are susceptible to clogging and deterioration of their function, and we have limited understanding of the evolution of their function due to the lack of long-term monitoring. The ability of these systems to infiltrate water depends on the unsaturated hydraulic conductivity function K of the soil. We introduce a novel approach based on physics informed neural networks (PINNs) to estimate K of a homogeneous column of soil using data from volumetric water content sensors and by solving the Richards–Richardson partial differential equation (RRE). We introduce and compare two different deep neural network architectures to solve RRE and estimate K . To generate the ground truth, we simulate three types of soil water dynamics using HYDRUS-1D and compare the results of these two neural network architectures in terms of the estimation of K . We investigate the effect of inter-sensor placement on the estimation of K . Both architectures show satisfactory performance on homogeneous soil with three volumetric water content sensors with different advantages. PINN-based estimation of K can be used fundamental tool for assessment of the evolution of the performance of GI over time, while requiring as input only the data from simple soil moisture sensors that are easily installed at the time of GI construction or even retrofitted.

Keywords Green infrastructure · HYDRUS-1D · Physics informed neural networks · Richards equation · Machine learning

1 Introduction

In the past two decades, green infrastructure (GI) has developed into a sustainable and effective method for maintaining and improving urban quality of life [1, 2]. GI manages urban storm water by utilizing vegetation and

permeable materials to route water through slower hydrologic pathways often improving water quality and providing habitat. However, in urban systems, implementation of GI is limited by available space and the need to re-establish connections between surface and subsurface flowpaths. Further, the tendency to focus monitoring in periods immediately following installation limits our ability to understand the evolution of system function and may obscure issues that develop and worsen over time [3]. In particular, infiltration-based GI systems, including rain gardens, permeable pavements, green roofs, and bioswales, are susceptible to clogging and consequent deterioration in system function [4].

Infiltration-based GI routes surface water and storm water run-off to recharge groundwater systems [5]. Their infiltration rates vary with the hydraulic conductivity (K) of the soil, and decreases in K will limit the recharge [6]. In most cases, infiltration is measured as a falling head with

✉ Mahmoud Elkhadrawi
mae116@pitt.edu

¹ Department of Electrical and Computer Engineering,
University of Pittsburgh, Pittsburgh, PA, USA

² Department of Civil and Environmental Engineering,
University of Pittsburgh, Pittsburgh, PA, USA

³ Department of Geology and Environmental Science,
University of Pittsburgh, Pittsburgh, PA, USA

⁴ Department of Civil and Environmental Engineering,
Northwestern University, Evanston, IL, USA

pressure sensors or via time difference in soil moisture dynamics between moisture sensor depths. However, neither approach resolves the hydraulic properties of the porous media (see data description section) and therefore our ability to evaluate GI infiltration mechanistically is limited.

Laboratory methods used to identify the hydraulic conductivity are unsatisfactory, as sampling introduces artifacts in the measured hydraulic properties of the soil, thus, giving us results that might not be representative of field conditions [7]. That said, movement of water in unsaturated porous media is complicated by hysteresis in the relationship between soil moisture content and hydraulic conductivity. Inference of K generally relies on numerical solutions of the Richards–Richardson equation (RRE) [8, 9]. Numerical approaches involve implicit time discretization schemes and finite volume or finite element discretization in space [10]. These methods could become computationally expensive, as they require repeated evaluation of the forward problem [11]. Also, the solution of the RRE may deteriorate upon broader application, when certain conditions are not met [12].

Typically the solution of the RRE and its inverse is found using empirical parametric models, defining the hydraulic conductivity as a function of matric potential, Ψ , or volumetric water content, θ (soil moisture), as well as defining the relationship between Ψ and θ [8]. The Mualem–Van Genuchten [13] and the Brooks and Corey's [14] models are the most commonly used [15]. These models depend on the knowledge of some soil properties such as pore-size distribution and the saturated water content, which need to be estimated.

Another challenge in the identification of the inverse solution is that most methods require knowledge of initial and boundary conditions [7, 16], which are needed to evaluate the forward problem. However, they are difficult to identify under field conditions. An approach to overcome this problem is to model using Gaussian process (GP) regression [17–19]. But this method comes with its own set of drawbacks, mainly that it does not deal well with nonlinearities (in our case the nonlinear parameters of the RRE), and local linearization might be required to circumvent this [20]. Rai et al. [17] proposed to deal with the nonlinearity of the parameters by using the Van Genuchten model and by trying to estimate its parameters as part of the GP regression model; their approach is limited to using a predefined model for the hydraulic parameters. Moreover, GP modeling is limited by the Gaussian prior assumption.

In this paper, we propose a physics-informed neural network (PINN) framework to estimate the hydraulic conductivity. Usage of PINNs and their advantages and limitations for different geoscientific applications such as

geothermic and hydrological applications have been studied in multiple works [20–24]. In this work, we contrast our framework to estimate the hydraulic conductivity (architecture 1) with an existing framework proposed by Bandai et al. [26] which is itself a PINN model as described by Raissi et al. [20] (architecture 2). Both architectures use multi-depth time series data of volumetric water content, and both use the residuals of the RRE as a loss function to represent the physical constraints describing soil–water dynamics.

Neither of the above-mentioned architecture 1 nor architecture 2 requires knowledge of initial or boundary conditions, which are hardly available under field conditions such as in GI; they don't require any predefined shape for the hydraulic conductivity function nor prior approximations for it; they only use data that are easily available from a simple array of soil moisture sensors and they don't require matric potential measurements to find the inverse solution.

Tools that utilize common monitoring data streams to assess hydraulic conductivity and infer changes in K fill an important gap in our ability to manage and enhance green infrastructure systems. For example, fine sediment particles, contributed by erosion of surrounding soil, decrease K , causing water backups and degrading GI system function. In addition, decay of biological components in green infrastructure media (e.g., mulch) can also diminish K , therefore changes in hydraulic conductivity can signal the need for particular maintenance tasks [27].

As we are interested in the practical application of these methods in managing GI and to allow the estimation of the hydraulic conductivity function evolution as a GI performance metric, we investigate the usage of a simple and convenient moisture sensor array set-up that uses three moisture sensors to estimate K by finding the inverse solution of the RRE. Three sensors are the absolute minimum for application of the methods documented here (in order to estimate the second derivative of the volumetric water content w.r.t. space to solve the RRE), these sensors are: sensor 0 (S_0) at depth z_0 , sensor 1 (S_1) at depth z_1 and sensor 2 (S_2) at depth z_2 , as shown in Fig. 1.

We model the effect of placing the sensors at different depths and with different inter-sensor distances on the estimation of the hydraulic conductivity. We consider homogeneous and non-homogeneous soil architecture for our simulations. Through simulations we also learn the ground truth for the hydraulic conductivity to evaluate the performance of the two architectures described above.

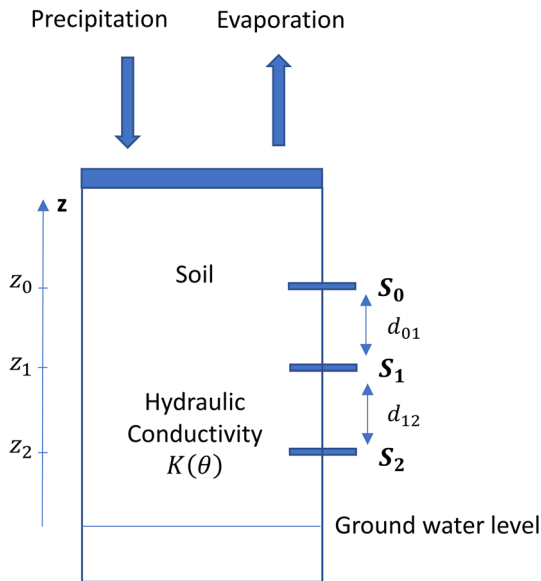


Fig. 1 Modeled GI soil profile with an array of 3 volumetric water content sensors placed at depths z_0 , z_1 and z_2 . d_{01} and d_{12} are the inner distances between sensors S_0 and S_1 and S_1 and S_2 respectively

2 Methods

2.1 The Richards–Richardson equation (RRE)

The law governing the unsaturated flow of water in porous media is Darcy's law [28]:

$$q = -K(\theta)\nabla h \quad (1)$$

$$h = \psi(\theta) + Z \quad (2)$$

where q is the water flow velocity $[\text{length}, \text{time}]^{-1}$, K is the hydraulic conductivity $[\text{length}, \text{time}]^{-1}$, and h is the hydraulic head (total potential) $[\text{length}]$, ψ is the matric potential $[\text{length}]$ and Z the elevation head $[\text{length}]$ along the vertical dimension z (positive upwards). Notice that we don't account for hysteresis, by assuming the relationship between θ and ψ , and between K and θ (or K and ψ) to be unique, as per the Mualem–Van Genuchten model [13]. The RRE is derived from Darcy's law and the continuity requirement (incompressible fluid, and not accounting for water vapor) and is defined as:

$$\frac{\partial \theta}{\partial t} = -\nabla \cdot q \quad (3)$$

$$\frac{\partial \theta}{\partial t} = \nabla \cdot (K\nabla \psi) + \frac{\partial K}{\partial z} \quad (4)$$

This form is called the mixed formulation, which includes θ and Ψ as variables. This formulation is used in the second PINN method (second architecture).

In this study, we only consider one-dimensional flow along the z -dimension, so the equation becomes:

$$\frac{\partial \theta}{\partial t} = \frac{\partial K}{\partial z} \frac{\partial \Psi}{\partial z} + K \frac{\partial^2 \Psi}{\partial z^2} + \frac{\partial K}{\partial z} \quad (5)$$

In order to get a formulation that only depends on θ , we define the moisture diffusivity D $[\text{length}]^2[\text{time}]^{-1}$ as:

$$D(\theta) = K(\theta) \frac{d\Psi}{d\theta} \quad (6)$$

From (5) and (6) we get the moisture formulation, which will be used in our proposed method (first architecture):

$$\frac{\partial \theta}{\partial t} = \nabla \cdot (D\nabla \theta) + \frac{\partial K}{\partial z} \quad (7)$$

Considering only the z -dimension we get:

$$\frac{\partial \theta}{\partial t} = \frac{\partial}{\partial z} \left(D \frac{\partial \theta}{\partial z} \right) + \frac{dK}{d\theta} \cdot \frac{\partial \theta}{\partial z} \quad (8)$$

$$\frac{\partial \theta}{\partial t} = D \frac{\partial^2 \theta}{\partial z^2} + \frac{\partial D}{\partial \theta} \frac{\partial \theta}{\partial z} \frac{\partial \theta}{\partial z} + \frac{dK}{d\theta} \cdot \frac{\partial \theta}{\partial z} \quad (9)$$

$$\frac{\partial \theta}{\partial t} = D \frac{\partial^2 \theta}{\partial z^2} + \frac{\partial D}{\partial \theta} \left(\frac{\partial \theta}{\partial z} \right)^2 + \frac{dK}{d\theta} \cdot \frac{\partial \theta}{\partial z} \quad (10)$$

2.2 First deep learning architecture

The goal of our network is to estimate the nonlinear coefficients on the right-hand side of Eq. (10): $\hat{D}(\theta)$, $\frac{d\hat{D}(\theta)}{d\theta}$ and $\frac{d\hat{K}(\theta)}{d\theta}$. Then, an estimate of the hydraulic conductivity function $\hat{K}$ is obtained by integrating the estimated $\frac{d\hat{K}(\theta)}{d\theta}$ over all observed values of θ . These estimated coefficients are then multiplied by their respective terms, $\frac{\partial^2 \theta}{\partial z^2}$, $\left(\frac{\partial \theta}{\partial z} \right)^2$ and $\frac{\partial \theta}{\partial z}$. Finally, they are summed to predict the value of the derivative of the volumetric water content w.r.t. time $\frac{\partial \theta}{\partial t}$, i.e., the left-hand-side of the RRE as shown in Fig. 2a.

The estimation of these coefficients is achieved using a set of volumetric water content data collected at three fixed depths $\theta[t^{(i)}, z_q]_{i=1}^{i=N}$, where N is the number of measurement points in time and the depths $z_q \in \{z_0, z_1, z_2\}$. The network uses discrete time and finite difference approximations of the derivatives of θ with respect to z , evaluated at depth level z_1 (i.e., the middle sensor S_1). The desired neural network $f(\mathbf{X}; \mathbf{W})$ (Fig. 2) takes as input current and past estimates of (1) the first derivatives w.r.t. z , (2) the square of the first derivative w.r.t. z , (3) the second derivative w.r.t. z (4) and the value of the volumetric water content measured by the middle sensor S_1 as shown in Fig. 1. Then, the network outputs the estimates of $D[t^{(i)}, z_1]$, $\frac{\partial D[t^{(i)}, z_1]}{\partial t}$ and $\frac{\partial D[t^{(i)}, z_1]}{\partial t}$. $\mathbf{W}$ represents the network's parameters (weights and biases) and $\mathbf{X}$ represents its input.

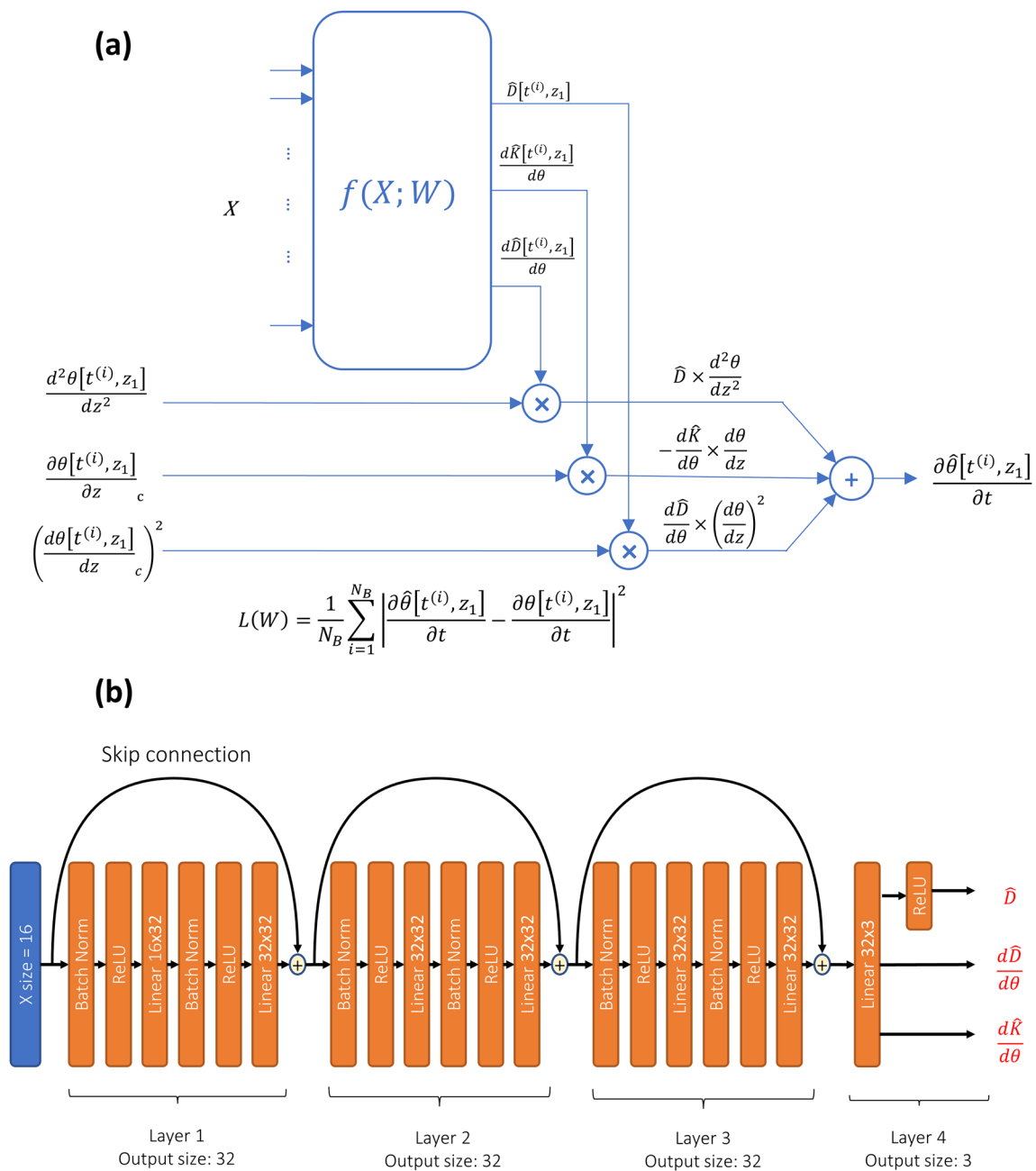


Fig. 2 **a** First network architecture. Figure **b** is the detailed architecture of the network $f(X; W)$

The first and second derivative of the volumetric water content w.r.t to space are evaluated at sensor S_1 (at depth z_1) using first-order finite difference approximation. We use the forward and backward difference of the first derivative w.r.t space as inputs. We use p past estimates as well:

$$\frac{\partial \theta[t^{(i-j)}, z_1]}{\partial z}_f, \frac{\partial \theta[t^{(i-j)}, z_1]}{\partial z}_b, j = 0, 1, \dots, p \quad (11)$$

The forward difference is defined as:

$$\frac{\partial \theta[t^{(i)}, z_1]}{\partial z}_f = \frac{\theta[t^{(i)}, z_2] - \theta[t^{(i)}, z_1]}{d_{12}} \quad (12)$$

while the backward difference is defined as:

$$\frac{\partial \theta[t^{(i)}, z_1]}{\partial z}_b = \frac{\theta[t^{(i)}, z_1] - \theta[t^{(i)}, z_0]}{d_{01}} \quad (13)$$

d_{12} and d_{01} are inner distances between the sensors as defined in Fig. 1.

We also add the squared values of the forward and backward derivatives as input as mentioned above.

$$\left(\frac{\partial\theta[t^{(i-j)}, z_1]}{\partial z}\right)_f, \left(\frac{\partial\theta[t^{(i-j)}, z_1]}{\partial z}\right)_b, j = 0, 1, \dots, p \quad (14)$$

The second derivative estimate at sensor S_1 is defined as:

$$\frac{\partial^2\theta[t^{(i)}, z_1]}{\partial z^2} \cong \frac{\frac{\partial\theta[t^{(i)}, z_1]}{\partial z}_f - \frac{\partial\theta[t^{(i)}, z_1]}{\partial z}_b}{\frac{(d_{12} + d_{01})}{2}} \quad (15)$$

We use current and $\mathbf{p}$ past estimates of the second derivative as part of the input.

$$\frac{\partial^2\theta[t^{(i-j)}, z_1]}{\partial z^2}, j = 0, 1, \dots, p \quad (16)$$

We also add current moisture value sensor S_1 to the input:

$$\theta[t^{(i)}, z_1] \quad (17)$$

The network then outputs (1) $\hat{D}[t^{(i)}, z_1]$ which gets multiplied by $\frac{\partial^2\theta[t^{(i)}, z_1]}{\partial z^2}$, (2) $\frac{d\hat{D}[t^{(i)}, z_1]}{d\theta}$ which gets multiplied by the central difference approximation of the first derivative w.r.t space defined as:

$$\frac{\partial\theta[t^{(i)}, z_1]}{\partial z}_c \cong \alpha_f \frac{\partial\theta[t^{(i)}, z_1]}{\partial z}_f + \alpha_b \frac{\partial\theta[t^{(i)}, z_1]}{\partial z}_b \quad (18)$$

$$\alpha_f = \frac{d_{12}}{d_{12} + d_{01}}, \alpha_b = \frac{d_{01}}{d_{12} + d_{01}} \quad (19)$$

The network is summarized as follows and is shown in Fig. 2:

$$\left[\hat{D}[t^{(i)}, z_1], \frac{d\hat{D}[t^{(i)}, z_1]}{d\theta}, \frac{d\hat{K}[t^{(i)}, z_1]}{d\theta}\right] = f(\mathbf{X}; \mathbf{W}) \quad (20)$$

$$\mathbf{X} = \begin{bmatrix} \frac{\partial\theta[t^{(i-j)}, z_1]}{\partial z}_f, \frac{\partial\theta[t^{(i-j)}, z_1]}{\partial z}_b \\ \left(\frac{\partial\theta[t^{(i-j)}, z_1]}{\partial z}_f\right)^2, \left(\frac{\partial\theta[t^{(i-j)}, z_1]}{\partial z}_b\right)^2 \\ \frac{\partial^2\theta[t^{(i-j)}, z_1]}{\partial z^2} \\ \theta[t^{(i)}, z_1] \end{bmatrix} j = 0, 1, \dots, p \quad (21)$$

$$\begin{aligned} \frac{\partial\hat{\theta}[t^{(i)}, z_1]}{\partial t} &= \hat{D}[t^{(i)}, z_1] \frac{\partial^2\theta[t^{(i)}, z_1]}{\partial z^2} \\ &+ \frac{d\hat{D}[t^{(i)}, z_1]}{d\theta} \left(\frac{\partial\theta[t^{(i)}, z_1]}{\partial z}_c\right)^2 + \frac{d\hat{K}[t^{(i)}, z_1]}{d\theta} \cdot \frac{\partial\theta[t^{(i)}, z_1]}{\partial z}_c \end{aligned} \quad (22)$$

The network parameters $\mathbf{W}$ are found by minimizing the loss function L that we define as the mean squared error (MSE) of the residuals $\mathbf{r}$ of the RRE:

$$L(\mathbf{W}) = \frac{1}{N_B} \sum_{n=1}^{N_B} \left| r[t^{(i)}, z_1] \right|^2 \quad (23)$$

$$r[t^{(i)}, z_1] = \frac{\partial\hat{\theta}[t^{(i)}, z_1]}{\partial t} - \frac{\partial\theta[t^{(i)}, z_1]}{\partial t} \quad (24)$$

where N_B represents the size of a training batch. The ground truth first derivative w.r.t. time is estimated using the following first order finite difference approximation:

$$\frac{\partial\theta[t^{(i)}, z_1]}{\partial t} \cong \frac{\theta[t^{(i)}, z_1] - \theta[t^{(i-1)}, z_1]}{\Delta t} \quad (25)$$

where Δt is the sampling period of the volumetric water content θ (see data description section). Here, we have set the number of past values $\mathbf{p}$ used in the input to 2, thus the number of features of the input $\mathbf{X}$ is 16. The network $\mathbf{f}$ is made of 4 layers (Fig. 2b). The first 3 layers are residual layers. Each residual layer is made of a batch normalization (batch norm) layer, a ReLU activation function, a fully connected layer (with biases), then another set of batch Norm, ReLU and fully connected layers. The output of the layer gets summed with its input hence the name residual layer [29]. However, for the first skip connection, we use a fully connected layer of size 16×32 in order to upscale the input features so that they have the same number of features as the output of the layer. For the other two residual layers, the input is directly summed to the output.

The last layer (4th layer) is made of a fully connected layer. A ReLU activation function is used only for the output that represents the estimate of the diffusivity $\hat{D}$. This was done to constrain its value to be positive. For the outputs $\frac{d\hat{K}[t^{(i)}, z_1]}{d\theta}$ and $\frac{d\hat{D}[t^{(i)}, z_1]}{d\theta}$, no activation functions were used.

Each estimate of $\frac{d\hat{K}[t^{(i)}, z_1]}{d\theta}$ is associated with a specific input value of volumetric water content to $\theta[t^{(i)}, z_1]$, which means that there will be multiple estimates of $\frac{d\hat{K}}{d\theta}$ for each unique θ value. So, for each unique value of θ , we compute the following value:

$$\frac{d\hat{K}(\theta)}{d\theta} = \text{median} \left(\frac{d\hat{K}[t^{(i)}, z_1]}{d\theta} \right) \quad (26)$$

Which we then integrate using the trapezoidal rule to get an estimate of the hydraulic conductivity function $\hat{K}(\theta)$.

The network was created on PyTorch (v 1.12.1, CUDA 11.6) using Python (v 3.8.2). We trained the network using ADAM optimizer with weight decay [30]. With a learning rate of 0.002, β_1 equal to 0.9, and β_2 equal to 0.999. The

number of epochs was 500 and the batch size N was set to 80.

2.3 Second deep learning architecture

The framework used for the second approach is the one proposed by Bandai et al. [26, 31] and it is based on PINNs as in the work of Raissi et al. [20]. PINNs are used to find the inverse of the RRE using the same set of volumetric water content data mentioned in the previous section $\theta[t^{(i)}, z^{(i)}]_{i=1}^{i=N}$. The PINNs used here are made of three networks as shown in Fig. 3.

The first network predicts the value of the matrix potential Ψ :

$$\hat{\psi}^{(i)} = f_{\psi}(t^{(i)}, z^{(i)}; W_{\psi}) \quad (27)$$

where W_{ψ} represents the network parameters. The network f_{ψ} is made of 8 fully connected layers and uses the hyperbolic tangent function, $\tanh$, as activation function after each layer. However, the last layer of this network uses the negative exponential function, $-\exp(x)$, as activation function, to force all the predicted values of matrix potential to be negative. The fully connected layers are made of 40 neurons. The predicted value of moisture potential $\hat{\psi}^{(i)}$ is converted to logarithmic scale:

$$\hat{\psi}_{\log}^{(i)} = -\log_e(-\hat{\psi}^{(i)}) \quad (28)$$

$\hat{\psi}_{\log}^{(i)}$ is then fed to two networks, which predict θ and K (Fig. 3a, b):

$$\hat{\theta}^{(i)} = f_{\theta}(\hat{\psi}_{\log}^{(i)}; W_{\theta}) \quad (29)$$

$$\hat{K}^{(i)} = f_K(\hat{\psi}_{\log}^{(i)}; W_K) \quad (30)$$

Note that this architecture assumes the relationship between θ and ψ , and between K and ψ to be unique as we use the Mualem-Van Genuchten model (see data description section). Network f_{θ} is made of 3 fully connected layers and network f_K is made of 1 fully connected layer. Both networks' fully connected layers are made of 40 neurons, and the hyperbolic tangent ($\tanh$) was used as activation function.

The parameters of the networks represented by the set $W = \{W_{\Psi}, W_{\theta}, W_K\}$, are found by minimizing the following loss function:

$$L(W) = \sum_{i=1}^N (\hat{\theta}[t^{(i)}, z^{(i)}] - \theta[t^{(i)}, z^{(i)}])^2 + \sum_{i=1}^N (r[t^{(i)}, z^{(i)}])^2 \quad (31)$$

$$r[t^{(i)}, z^{(i)}] = \frac{\partial \hat{\theta}}{\partial t} - \frac{\partial \hat{K}}{\partial z} \frac{\partial \hat{\psi}}{\partial z} - \hat{K} \frac{\partial^2 \hat{\psi}}{\partial z^2} - \frac{\partial \hat{K}}{\partial z} \quad (32)$$

where N is the total number of measurements points. The first term of the loss function represents the error of the estimated volumetric water content and r is the residual of the RRE. In order to compute the residual, the derivatives

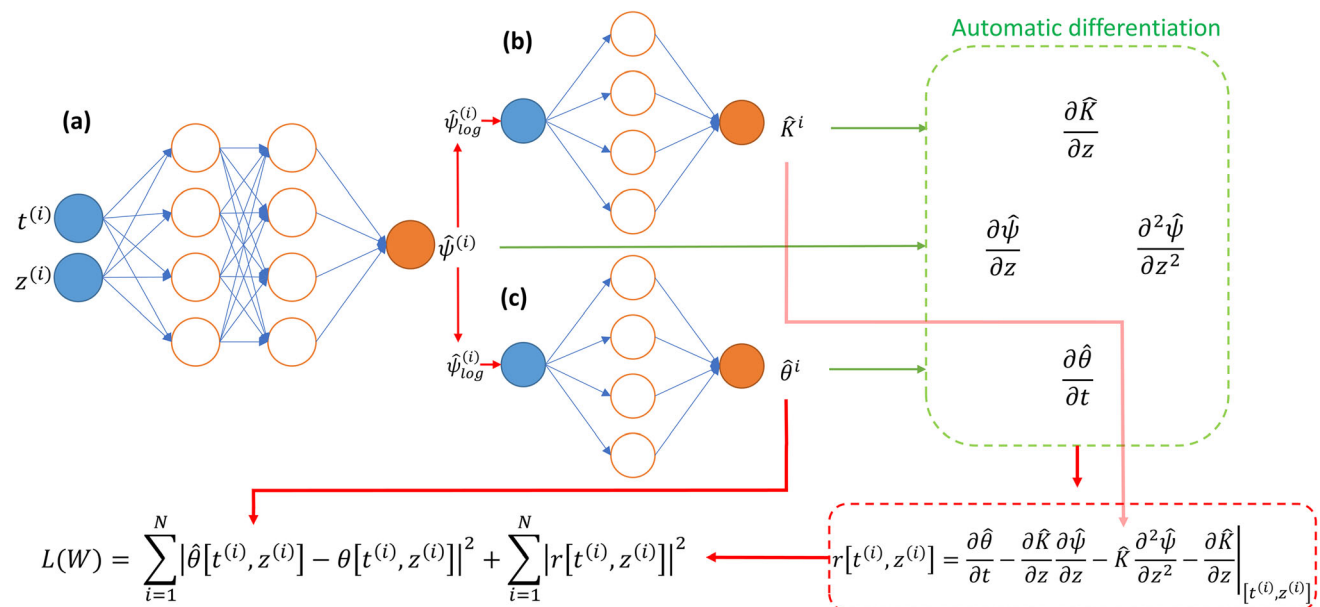


Fig. 3 The second network architecture physics informed neural networks (PINNs) as described in [6]. **a** is the network f_{ψ} that outputs the estimated matrix potential $\hat{\psi}$. **b** is the network f_K that outputs the estimated hydraulic conductivity $\hat{K}$. **c** is the network f_{θ} that outputs

the estimated volumetric water content $\hat{\theta}$. The number of layers and neurons in this figure is not the actual one but only representative

$\frac{\partial \theta}{\partial t}, \frac{\partial K}{\partial z}, \frac{\partial \psi}{\partial z}, \frac{\partial^2 \psi}{\partial z^2}$ are evaluated using automatic differentiation [32]. The residuals as well as the error of the volumetric water content are evaluated at all measurement points. Tanh is used as activation function (as well as $-\exp(x)$) because it is twice differentiable, which is necessary because we need to compute the second derivative $\frac{\partial^2 \psi}{\partial z^2}$.

The described network architecture is the same as the one used by the best performing model in [26]. An important caveat is that the best model was found on PINNs that used volumetric water content data from 10 measurement depths, while we use three measurement depths in this study. The weights W_ψ of the network f_ψ (Fig. 3a) are initialized using Xavier uniform initialization [33]. For the networks f_K (Fig. 3b) and f_θ (Fig. 3c), the weights were initialized by a squared Xavier uniform initialization. This forces the networks to start the training from a space where they are increasing monotonic functions of $\hat{\psi}$. Bandai et al. [26] describe the preference to make the networks increasing monotonic functions of $\hat{\psi}$ by the fact that it aligns with the physical nature of soil–water dynamics. The networks were trained using the Adam optimizer followed by the L-BFGS-B optimizer. The Adam optimizer used a learning rate equal to 0.001, β_1 equal to 0.9, and β_2 equal to 0.999. The number of epochs was 5000, and the batch size was equal to the size of all the measurements points (number of batches = 1). The L-BFGS-B optimizer parameters were set to the following: maxcor = 50, maxls = 50, maxiter = 50, 000, maxfun = 50, 000, ftol = $2.220446049250313 \times 10^{-16}$. The PINNs were created using TensorFlow v 1.14 as in the original work.

2.4 Data description

The volumetric moisture content measurement data was generated using HYDRUS-1D software [34]. A column of homogeneous soil of 100 cm was simulated and uniformly discretized at 0.1 cm intervals. Three types of soils were simulated: sandy loam, loam and silt loam. The Mualem–Van Genuchten model was used to parametrize the hydraulic conductivity function $K(\psi)$ and the soil–water retention curve $\theta(\psi)$ [13]:

$$\theta(\psi) = \theta_r + \frac{\theta_s - \theta_r}{(1 + (-\alpha\psi)^n)^m} \quad (33)$$

$$K(\theta(\psi)) = K_s S_e^l \left(1 - \left(1 - S_e^{\frac{1}{m}} \right)^m \right)^2 \quad (34)$$

$$S_e = \frac{\theta - \theta_r}{\theta_s - \theta_r} \quad (35)$$

$$m = 1 - \frac{1}{n} \quad (36)$$

The fitting parameters of these functions for the three types of simulated soils are shown in Table 1.

Notice that θ and K are increasing monotonic functions of ψ . The initial matric potential was set to -1000 cm for all depths. And, the Neumann boundary condition was used as the bottom boundary condition:

$$\frac{\partial \psi}{\partial z} = 0 \quad (37)$$

The upper boundary condition was set to atmospheric condition. To generate the time series data, three days of upper boundary water density flux were simulated (Table 2). The volumetric water content was sampled every 0.012 day). The scenarios are generated to provide the same number of training points for the estimation of the derivative of the volumetric water content w.r.t. z (finite difference estimation of the first and second derivative of the middle sensor S_1) for the first architecture. The second architecture uses only the data from scenario 1.

We used simulated measurement data at depths $z = -1, -3, -5, -7, -9, -11, -13, -15, -17, -19$ cm (z positive upward) to evaluate the effect of sensor placement configuration, both depths and inter-sensor distances. Thus, we will show the results for $\binom{10}{3} = 120$ configurations in this study as we are simulating using an array of 3 moisture sensors data. To compare between the estimated hydraulic conductivity functions, we calculate the following relative error:

$$\epsilon_k = \sum_{i=1}^{N_\theta} \frac{|\hat{K}(\theta^{(i)}) - K(\theta^{(i)})|^2}{|K(\theta^{(i)})|^2} \quad (38)$$

where N_θ is the number of point volumetric water content values for which $\hat{K}$ was estimated. This error is estimated at 500 evenly spaced values of θ for both architectures, where the beginning and the end of the interval correspond to the minimum and maximum value of θ the data from scenario 1 for each simulated soil. The hydraulic conductivity values were linearly interpolated for θ values that weren't part of the training data.

3 Results

3.1 Homogeneous soil columns

To compare the overall performances of the estimation of the hydraulic conductivity function between both deep learning architectures, we show the median of the estimated $\hat{K}(\theta)$ for the 120 configurations of sensor placement for three different types of soils as described in Sect. 2.4

Table 1 Fitting parameters for the Mualem-Van Genuchten model for all simulated soils

	$\theta_r [\text{cm}^3 \text{ cm}^{-3}]$	$\theta_s [\text{cm}^3 \text{ cm}^{-3}]$	$\alpha [\text{cm}^{-1}]$	n	$K_s [\text{cm day}^{-1}]$	l
Sandy loam	0.065	0.41	0.075	1.89	106.1	0.5
Loam	0.078	0.43	0.036	1.56	24.96	0.5
Silt loam	0.067	0.45	0.02	1.41	10.8	0.5

Table 2 The 3 scenarios of surface water flux density q [cm day^{-1}] (positive upward) generated using HYDRUS-1D

Time [day]	Scenario 1 q [cm/day^{-1}]	Scenario 2 q [cm/day^{-1}]	Scenario 3 q [cm/day^{-1}]
0.25	− 10	− 10	− 15
0.5	0	0	0
1	0.3	0.3	0.4
1.5	0	5	− 1
2	0.3	0.3	0.5
2.25	− 10	− 5	X
2.5	0	− 5	− 5
3	0.3	0.3	0.3

X no event

(data description) (Fig. 4). We also show an envelope that spans from the 1st to 3rd quartiles of the estimated hydraulic conductivity values, allowing comparison of the variance in the estimations between both methods.

The errors of the median curves ε_K (Eq. 38) for both architecture for all soils are shown in Table 3. We can see that the errors of the median curves are smaller for architecture 2 for all three types of soils compared to architecture 1. Moreover, the 1st to 3rd quartile envelopes of architecture 2 are smaller than those of architecture 1. Thus, we can conclude that architecture 2 has better performance over all sensor placements trials. We note also that the estimation errors for both architectures are smaller with bigger hydraulic conductivity functions ($K_{\text{sandy loam}} > K_{\text{loam}} > K_{\text{silt loam}}$).

However, when comparing the results of the best configurations for each architecture, in Fig. 5 and Table 4, for soil of type sandy loam, we notice that the error is smaller for the first architecture compared to the second architecture, but the latter has smaller errors for the other two soils. We note also that the average depths of the sensor configurations for architecture 1 are smaller than those of architecture 2.

In order to document the effect of the inter-sensor distances on the network performance, we first compare the results of both architectures using the average inter-sensor distance, regardless of their depths. In Fig. 6, the x -axis is the average inter-sensor distance $x = \frac{d_{01} + d_{12}}{2}$ (d_{01} , and d_{12} are defined in Fig. 1) and the y axis represents the

corresponding average error of the hydraulic conductivity function calculated for all configurations.

We can see that the second architecture (Fig. 6b, d, f) is agnostic to the effect of having different inner distances. However, the first architecture's performance (Fig. 6a, c, e) worsens as the inner distances get bigger. Which aligns with the fact that we use finite difference approximations for this architecture. The results are consistent across all soil types. We also notice that the errors of the first architecture are one order of magnitude larger than those of the second architecture, specifically when the inter-sensor distances are large. Similar conclusions can be drawn by only considering the errors with respect to equal inner distances between the sensors, i.e., X axis = $d_{01} = d_{12}$ (results are shown in supplemental Fig. 1).

To analyze the effect of sensor depths on the performance, in Fig. 7, we show the average relative error of the hydraulic conductivity function ε_K depending on the average depth of the sensor placement configuration, which means that the x -axis is $x = \frac{z_0 + z_1 + z_2}{3}$. The error of the second architecture gets smaller as the depth of the average depth of the sensor array gets higher (Fig. 7b, d, f). Which shows that the second architecture achieves better performance at deeper depths.

The converse is true for the first architecture (Fig. 7a, c, e), the error is lower at shallower depths, which means that the first architecture achieves better results for shallow sensor placements, that have sharper wet fronts. Notice that the best sensor configurations for the first architecture in Fig. 5 are shallower than those of the second architecture, which aligns with the last conclusion. However, for the first architecture, we can see that the errors pass through a maximum at 10–11 cm, and then get lower again for deeper depths, although not as low as for the shallower depths 3–6 cm. Similar conclusions can be drawn by only considering the errors with respects to the middle sensor depth z_1 i.e., X axis = d_1 (results are shown in supplemental Fig. 2).

3.2 Non-homogeneous soil column

Although the proposed networks are intended to estimate the hydraulic conductivity for homogeneous soil, we tested their performance on a simulated non-homogeneous vertical soil column. This is done because we expect the soil of GI to be non-homogeneous in practice. The column is of

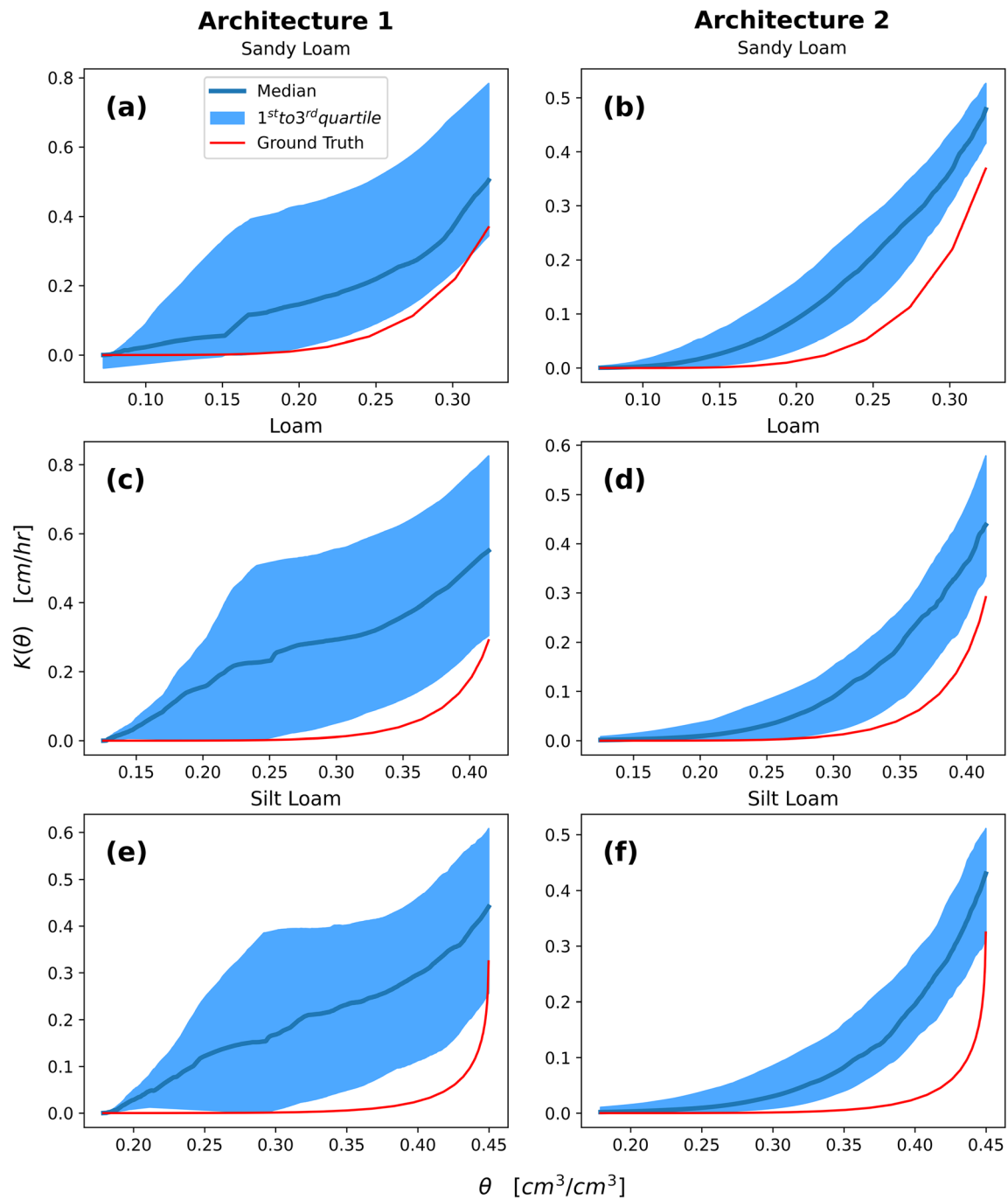


Fig. 4 First versus second architecture median hydraulic conductivity curve of the 120 sensor placements and 1st to 3rd quartile envelope. Figures **a**, **c** and **e** are the results of the first architecture. Figures **b**, **d** and **f** are the results of the second architecture

length 21 cm, made of three types of soil: sandy loam, loam and silt loam, placed in order from top to bottom, each of length 7 cm. We used the same 120 sensor placement configurations, and we show the median and the 1st to 3rd quartile envelope plots in Fig. 8. We compare the estimated hydraulic conductivity with the equivalent vertical hydraulic conductivity K_z of the three-layered soil, which is

Table 3 Errors of the median curves ε_K for architectures 1 and 2 for the three simulated soils

Soil type	ε_K Architecture 1	ε_K Architecture 2
Sandy loam	1.16	0.79
Loam	13.44	2.25
Silt loam	20.88	6.63

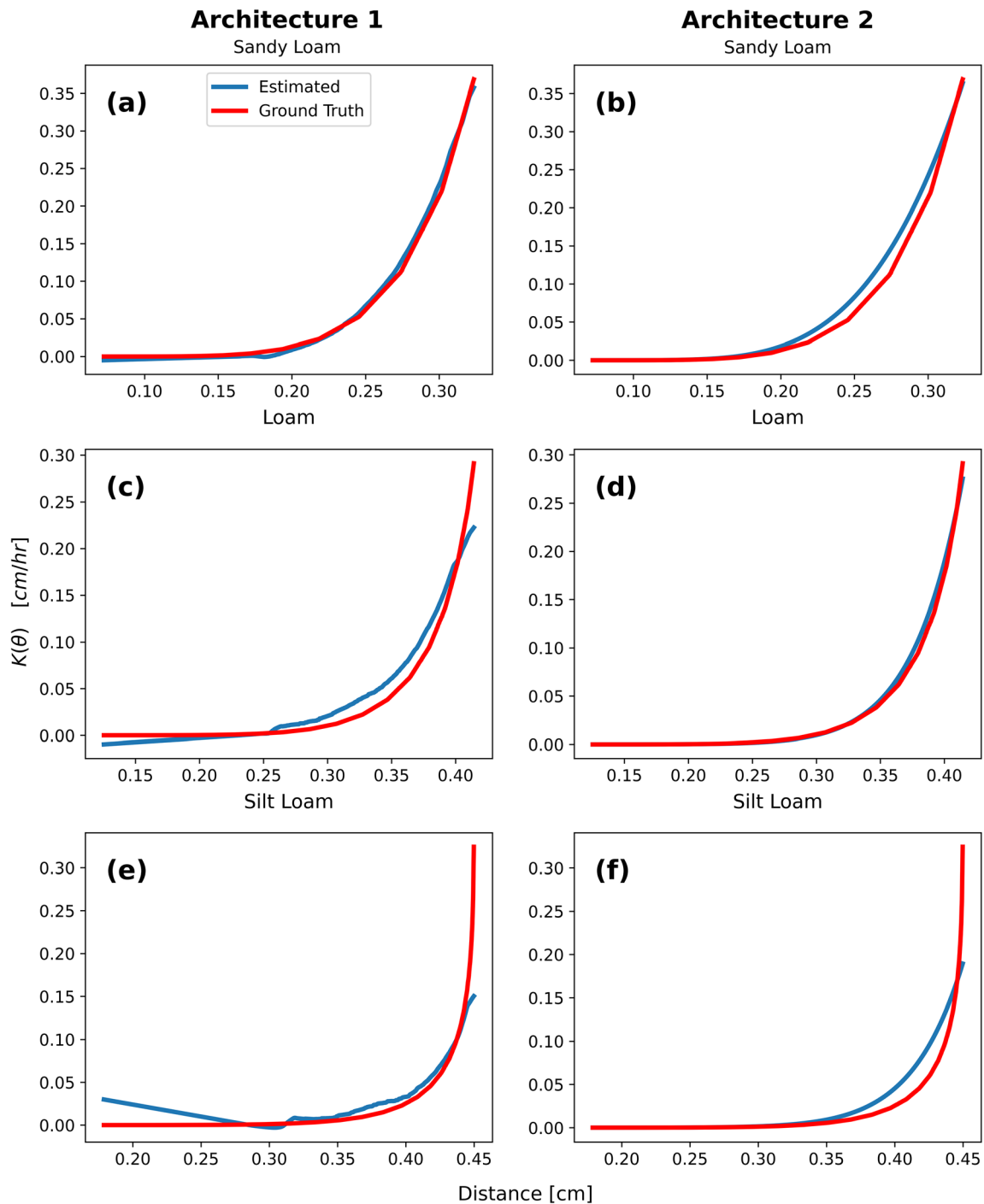


Fig. 5 First versus second architecture best sensor placement results. Figures **a**, **c** and **e** are the results of the first architecture. Figures **b**, **d** and **f** are the results of the second architecture

Table 4 Errors ε_K and sensor placements of the best estimated hydraulic conductivity curves for architectures 1 and 2 for the three simulated soils

Soil type	ε_K architecture 1	ε_K architecture 2
Sandy loam	0.00295 (− 3, − 5, − 9 cm)	0.01621 (− 7, − 13, − 19 cm)
Loam	0.04049 (− 1, − 3, − 15 cm)	0.00504 (− 9, − 11, − 19 cm)
Silt loam	0.15204 (− 1, − 3, − 15 cm)	0.12764 (− 13, − 15, − 17 cm)

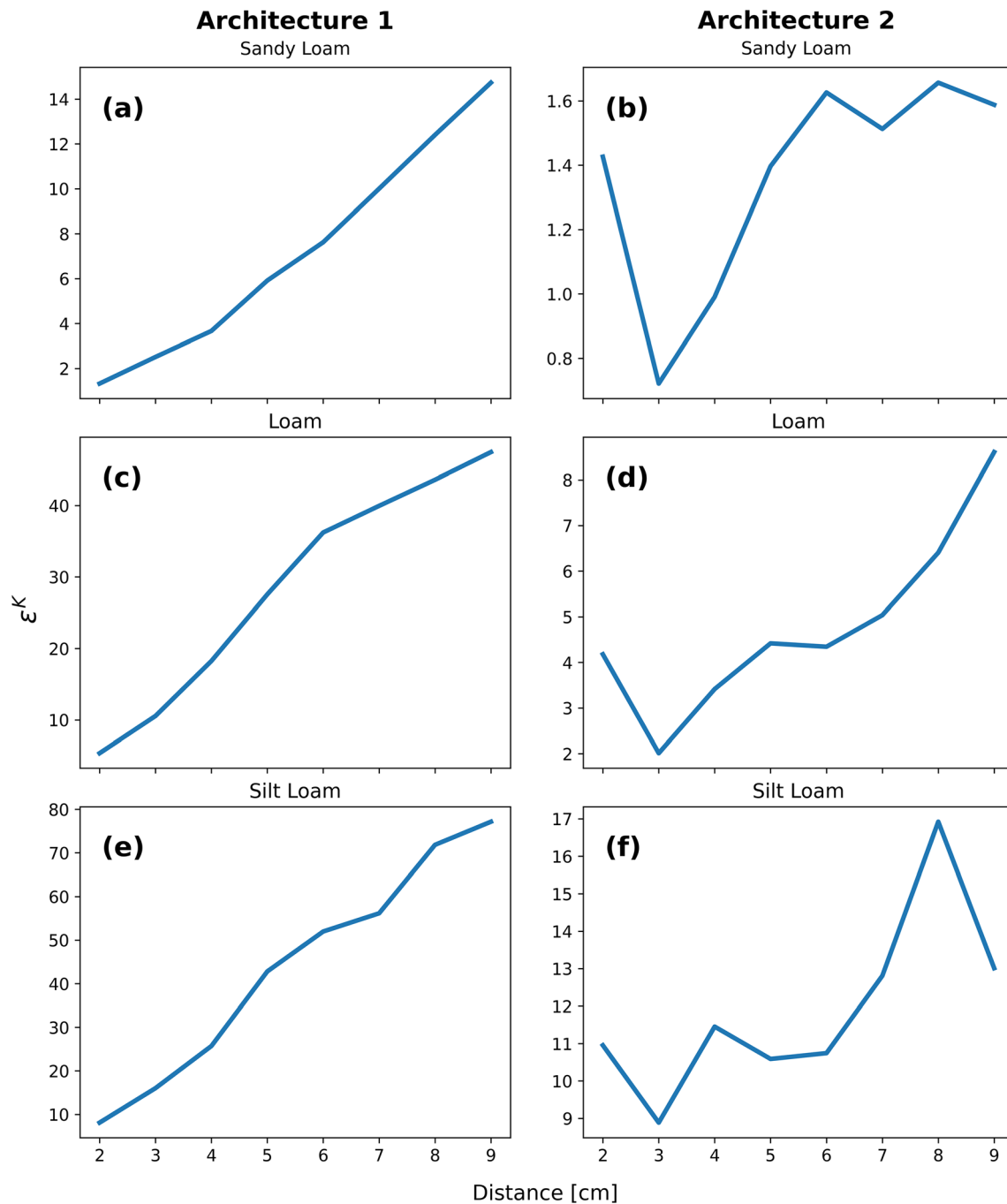


Fig. 6 Effect of average inter-sensor distance on the performance of the first vs second architecture best sensor placement results. X axis = $(d_{01} + d_{12})/2$. Figures **a**, **c** and **e** are the results of the first

architecture. Figures **b**, **d** and **f** are the results of the second architecture. Y axis represents the mean relative error

$$K_z = \frac{d}{\sum_{i=1}^n \frac{d_i}{K_i}} \quad (39)$$

where n is the number of layers of the soil, here 3, d is the length of the column, which is 21 cm, d_i is the depth of each layer, here 7 cm, and K_i is the layer's hydraulic conductivity function.

The errors of the median curves relative to the equivalent hydraulic conductivity K_z for architecture 1 and 2 are 0.984 and 0.591. Although smaller than those of the homogeneous soils, the median predicted K values are close to zero. Also, the variance of the estimation is high (specially for architecture 2).

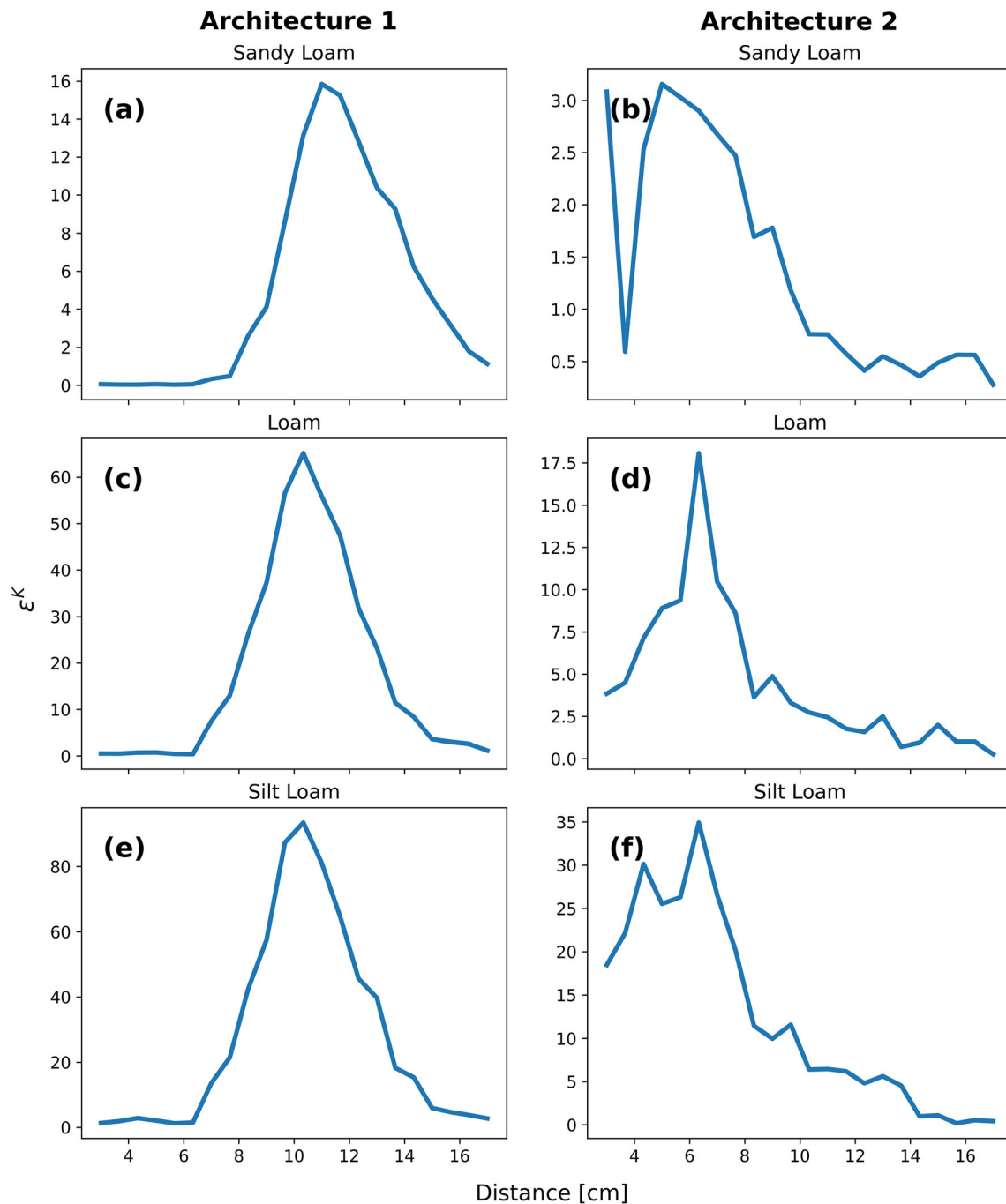


Fig. 7 Effect of the average depth of the sensors on the performance of the first vs second architecture. X axis = $(z_0 + z_1 + z_2)/3$. Figures **a**, **c** and **e** are the results of the first architecture. Figures **b**, **d** and **f** are the results of the second architecture. Y axis represents the mean relative error

The best estimated hydraulic conductivity results are shown in Fig. 9. The errors of architectures 1 and 2 are 0.413 and 0.178, and the corresponding sensor placements are $(-1, -5, -7 \text{ cm})$ and $(-9, -11, -15 \text{ cm})$, respectively. These errors are bigger than all the errors of homogeneous soil configurations.

3.3 Discussion and conclusions

In this work, we present two deep learning approaches to estimate the hydraulic conductivity function K using volumetric water content data from 3 moisture sensors. The first architecture shows the best estimate of K . Also, the first architecture has lower estimation error compared to the second architecture for sensor placements at shallower

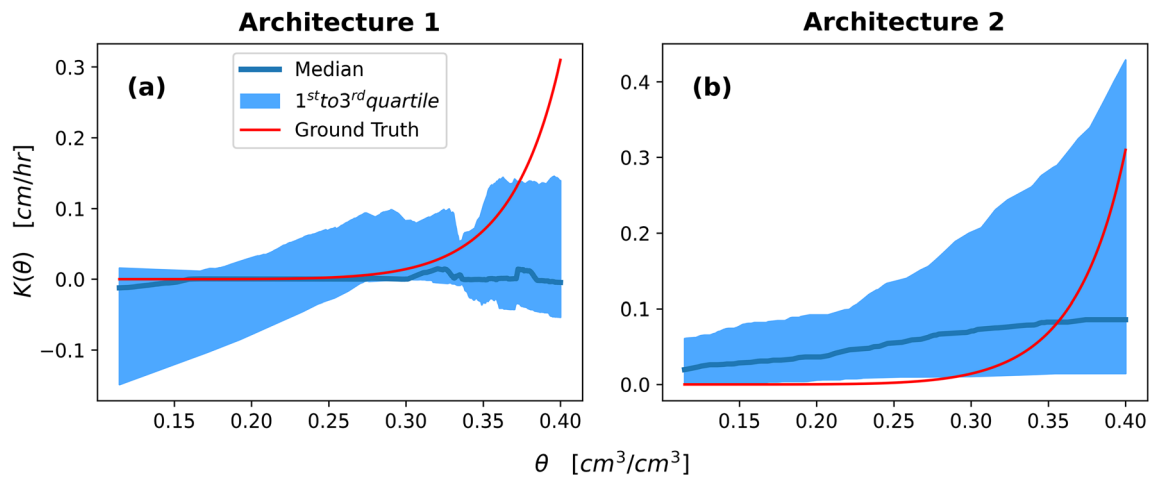


Fig. 8 First versus second architecture median hydraulic conductivity curve of the 120 sensor placements and 1st to 3rd quartile envelope. The ground truth is the equivalent vertical hydraulic conductivity K_z of the non-homogeneous soil column

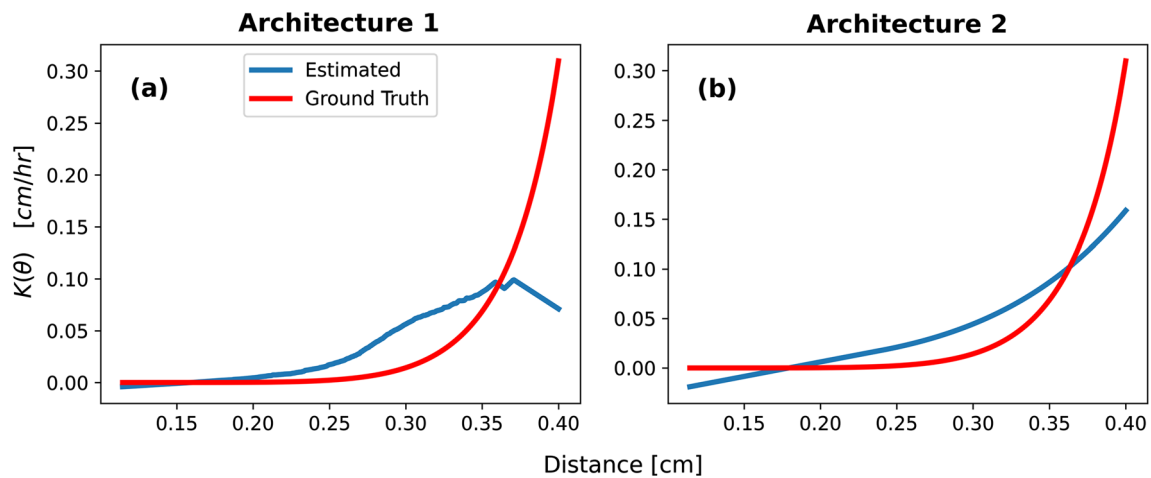


Fig. 9 First versus second architecture best sensor placement results. **a** Results of the first architecture **b** results of the second architecture. The ground truth is the equivalent vertical hydraulic conductivity K_z of the non-homogeneous soil column

depths. These two measures suggest the first architecture may be more applicable in shallow GI applications with sharp moisture gradients, such as green roofs which are shallow and are exposed directly to rain. (These formulations do not allow assessment of impacts of evapotranspiration at these shallow depths.) Beyond shallow systems, the second architecture had a better performance over all sensor placements and was insensitive to the inner distances between the sensors compared to the first architecture, which required smaller inner distances to give good estimates of K . Therefore, the second architecture is more flexible with sensor placement configurations.

The nebulous initial and boundary conditions lead to an ill posed problem. That is, traditional methods would fail to find the inverse solution of the RRE, as no unique solution can be guaranteed. However, the proposed PINN architectures, and many other similar examples [20, 24, 25],

have demonstrated it is possible to find the inverse solution of a differential equation using time-series data (or spatial data). For this work, data is simulated assuming outputs from an array of 3 moisture sensors.

However, the proposed methods are limited in important ways. Some physical processes were not accounted for in the formulation of the RRE and the architecture of the networks. In particular, the effect of hysteresis [35] was not accounted for, as we assumed the relationship between K and θ , θ and ψ to be unique. Also, the effect of coupled heat and water transport, and the effect of solute transport were ignored. Moreover, the performance of both architectures on non-homogeneous soil is unsatisfactory, therefore developing a model that accounts for multi-layered soils is needed for applications where changes in K at specific depths are important (e.g., evaluation of plant root impacts).

Nevertheless, the presented methods can be used as a fundamental tool for assessment of the evolution in the hydraulic conductivity of GI soil over time, while requiring as input only the data from simple soil moisture sensors that are easily installed at the time of GI construction or even retrofitted. The insight provided by these estimates of changing K can clarify processes of both clogging and expansion (e.g., creation of flow paths by roots), both of which are often invoked, but poorly documented in field conditions. In particular, these networks can be trained on data subsets from distinct periods of time. For example, we can compare winter K estimates with summer K estimates to evaluate the role of vegetation and other biota on water flow in the GI. Alternatively, we can evaluate differences in K estimates from one year to K estimates in subsequent years and evaluate both the relative importance and interactions between clogging and root mechanics. This method requires a minimal investment in sensors, allowing a simple and inexpensive approach for continuous monitoring of complicated GI functions.

Supplementary Information The online version contains supplementary material available at <https://doi.org/10.1007/s00521-023-09378-z>.

Acknowledgements The publicly available data used for this study (scenario 1 & 2) as well as the code for the second PINN architecture (based on Dr. Maziar Raissi PINN code) and the code used to transform “Nod_inf.out” files from Hydrus 1D to csv files created by Dr. Toshiyuki Bandai and Dr. Teamrat A. Ghezzehei were helpful for this study. This study was funded by the National Science Foundation, grant 1854827.

Funding This study was funded by the National Science Foundation, Grant 1854827.

Data availability The dataset used in this study is available through Zenodo <https://doi.org/https://doi.org/10.5281/zenodo.8173194>.

Declarations

Conflict of interest The authors have no competing interests to declare that are relevant to the content of this article.

References

- Badiu DL, Nita A, Iojă CI, Niță MR (2019) Disentangling the connections: a network analysis of approaches to urban green infrastructure. *Urban For Urban Green* 41:211–220. <https://doi.org/10.1016/j.ufug.2019.04.013>
- Hanna E, Comin FA (2021) Urban green infrastructure and sustainable development: a review. *Sustainability*. <https://doi.org/10.3390/su132011498>
- Clary J, Leisenring M, Jones J, Hobson P, Strecker E (2020) International stormwater BMP database: 2020 summary statistics. *Water Res Found* 4968:1–118
- William R, Gardoni P, Stillwell AS (2019) Reliability-based approach to investigating long-term clogging in green stormwater infrastructure. *J Sustain Water Built Environ*. <https://doi.org/10.1061/jswbay.0000875>
- Zhang K, Chui TFM (2019) A review on implementing infiltration-based green infrastructure in shallow groundwater environments: challenges, approaches, and progress. *J Hydrol*. <https://doi.org/10.1016/j.jhydrol.2019.124089>
- Lewellyn C, Lyons CE, Traver RG, Wadzuk BM (2016) Evaluation of seasonal and large storm runoff volume capture of an infiltration green infrastructure system. *J Hydrol Eng*. [https://doi.org/10.1061/\(asce\)jhe.1943-5584.0001257](https://doi.org/10.1061/(asce)jhe.1943-5584.0001257)
- Hopmans JW, Šimůnek J, Romano N, Durner W (2018) Inverse methods. *Methods Soil Anal Part 4 Phys Methods*. <https://doi.org/10.2136/sssabookser5.4.c40>
- Farthing MW, Ogden FL (2017) Numerical solution of richards' equation: a review of advances and challenges. *Soil Sci Soc Am J* 81(6):1257–1269. <https://doi.org/10.2136/sssaj2017.02.0058>
- Tracy FT (2006) Clean two- and three-dimensional analytical solutions of Richards' equation for testing numerical solvers. *Water Resour Res*. <https://doi.org/10.1029/2005WR004638>
- Cockett R, Heagy LJ, Haber E (2018) Efficient 3D inversions using the Richards equation. *Comput Geosci* 116:91–102. <https://doi.org/10.1016/j.cageo.2018.04.006>
- Bilionis I, Zabarar N (2014) Solution of inverse problems with limited forward solver evaluations: a Bayesian perspective. *Inverse Probl*. <https://doi.org/10.1088/0266-5611/30/1/015004>
- Zha Y, Yang J, Zeng J, Tso CHM, Zeng W, Shi L (2019) Review of numerical solution of Richardson–Richards equation for variably saturated flow in soils. *Wiley Interdiscip Rev Water*. <https://doi.org/10.1002/wat2.1364>
- van Genuchten MT (1980) A closed-form equation for predicting the hydraulic conductivity of unsaturated soils. *Soil Sci Soc Am J* 44:892–898
- Brooks R, Corey A (1964) Hydraulic properties of porous media. *Hydrol Pap Color State Univ* 3(3):37
- Barari A, Omidvar M, Ghotbi AR, Ganji DD, Barari A (2009) Numerical analysis of Richards' problem for water penetration in unsaturated soils. *Hydrol Earth Syst Sci Discuss* 6(5):6359–6385
- Tubini N (2021) Theoretical and numerical tools for studying the critical zone from plot to catchments. Doctoral dissertation, University of Trento [Online]. https://iris.unitn.it/retrieve/handle/11572/319821/498093/Tubini_2021_Theoretical_and_numerical_tools_for_studying_the_Critical_Zone_from_plot_to_catchments.pdf
- Rai PK, Tripathi S (2019) Gaussian process for estimating parameters of partial differential equations and its application to the Richards equation. *Stoch Environ Res Risk Assess* 33(8–9):1629–1649. <https://doi.org/10.1007/s00477-019-01709-8>
- Raissi M, Perdikaris P, Karniadakis GE (2018) Numerical Gaussian processes for time-dependent and nonlinear partial differential equations. *SIAM J Sci Comput* 40(1):A172–A198. <https://doi.org/10.1137/17M1120762>
- Raissi M, Perdikaris P, Karniadakis GE (2017) Machine learning of linear differential equations using Gaussian processes. *J Comput Phys* 348:683–693. <https://doi.org/10.1016/j.jcp.2017.07.050>
- Raissi M, Perdikaris P, Karniadakis GE (2019) Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J Comput Phys* 378:686–707. <https://doi.org/10.1016/j.jcp.2018.10.045>
- Degen D et al (2023) Perspectives of physics-based machine learning for geoscientific applications governed by partial differential equations. *Geosci Model Dev*. <https://doi.org/10.5194/gmd-16-7375-2023>
- Cuomo S, Di Cola VS, Giampaolo F, Rozza G, Raissi M, Piccialli F (2022) Scientific machine learning through physics-informed

- neural networks: where we are and what's next. *J Sci Comput*. <https://doi.org/10.1007/s10915-022-01939-z>
23. Cai S, Mao Z, Wang Z, Yin M, Karniadakis GE (2021) Physics-informed neural networks (PINNs) for fluid mechanics: a review. *Acta Mech Sin Xuebao* 37(12):1727–1738. <https://doi.org/10.1007/s10409-021-01148-1>
 24. Mishra S, Molinaro R (2022) Estimates on the generalization error of physics-informed neural networks for approximating a class of inverse problems for PDEs. *IMA J Numer Anal* 42(2):981–1022. <https://doi.org/10.1093/imanum/drab032>
 25. Yu J, Lu L, Meng X, Karniadakis GE (2022) Gradient-enhanced physics-informed neural networks for forward and inverse PDE problems. *Comput Methods Appl Mech Eng*. <https://doi.org/10.1016/j.cma.2022.114823>
 26. Bandai T, Ghezzehei TA (2021) Physics-informed neural networks with monotonicity constraints for richardson-richards equation: estimation of constitutive relationships and soil water flux density from volumetric water content measurements. *Water Resour Res*. <https://doi.org/10.1029/2020WR027642>
 27. EPA (2016) Operation and maintenance of green infrastructure receiving runoff from roads and parking lots. https://www.epa.gov/sites/production/files/2016-11/documents/final_gi_main_tenance_508.pdf
 28. Philip JR (1969) Theory of infiltration. *Adv Hydrosol*. <https://doi.org/10.1016/b978-1-4831-9936-8.50010-6>
 29. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. *Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit* 2016:770–778. <https://doi.org/10.1109/CVPR.2016.90>
 30. Loshchilov I, Hutter F (2019) Decoupled weight decay regularization. In: 7th International conference on learning representations ICLR 2019.
 31. Bandai T, Ghezzehei TA (2023) Applications of physics informed neural networks for modeling soil water dynamics. In: Abstract from 13th annual meeting interpore 2021. https://events.interpore.org/event/25/attachments/580/1179/2021_book-of-abstracts.pdf. Accessed 27 Nov 2023
 32. Güneş Baydin A, Pearlmutter BA, Andreyevich Radul A, Mark Siskind J (2018) Automatic differentiation in machine learning: a survey. *J Mach Learn Res* 18:1–43
 33. Glorot X, Bengio Y (2010) Understanding the difficulty of training deep feedforward neural networks. *J Mach Learn Res* 9:249–256
 34. Simunek J, Sejna M, Saito H, van Genuchten MT (2009) The HYDRUS-1D software package for simulating the one-dimensional movement of water, heat, and multiple solutes in variably-saturated media. *Environ Sci* 3:1–240
 35. Holtzman R, Dentz M, Planet R, Ortin J (2023) Hysteresis of multiphase flow in porous and fractured media. In: Abstract from 11th annual meeting interpore 2019 Valencia, Valencia, Spain. https://orbit.dtu.dk/files/187391656/book_of_abstracts_interpore.pdf. Accessed 27 Nov 2023

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