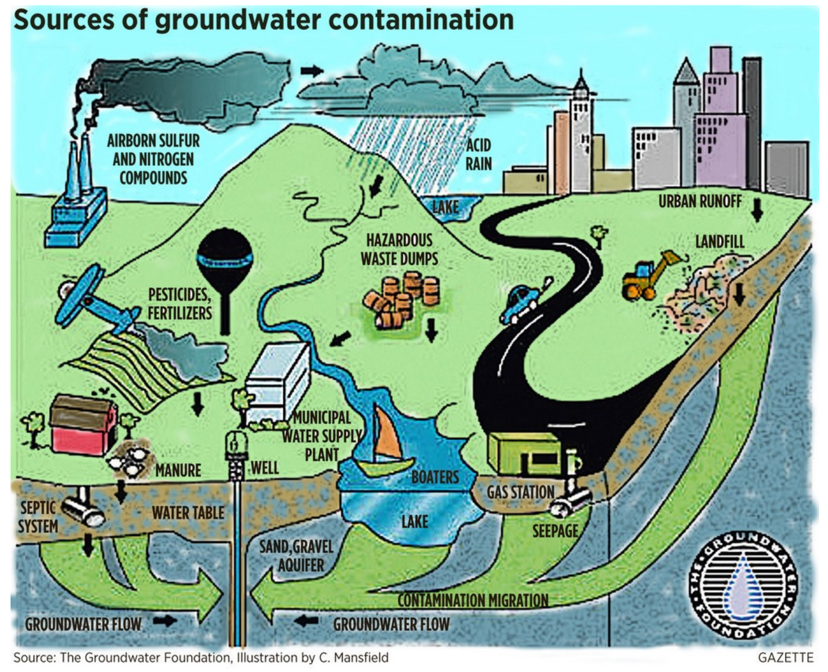
Physics-aware Neural Network Approach to Inverse Problems in Groundwater Contamination

Stanford
SCHOOL OF EARTH, ENERGY & ENVIRONMENTAL SCIENCES | Energy Resources Engineering

Hannah Lu and Ziyan Wang

1 Background and Motivation



Groundwater contamination occurs when man-made products such as gasoline, oil and chemicals get into the groundwater and make it unsafe and unfit for human use. For science-informed regulations and remedies, *it is crucial to estimate the highly heterogeneous aquifer properties and to identify the contaminant sources based on measurements of contaminate concentration.* This is defined as **inverse problems** in groundwater contamination.

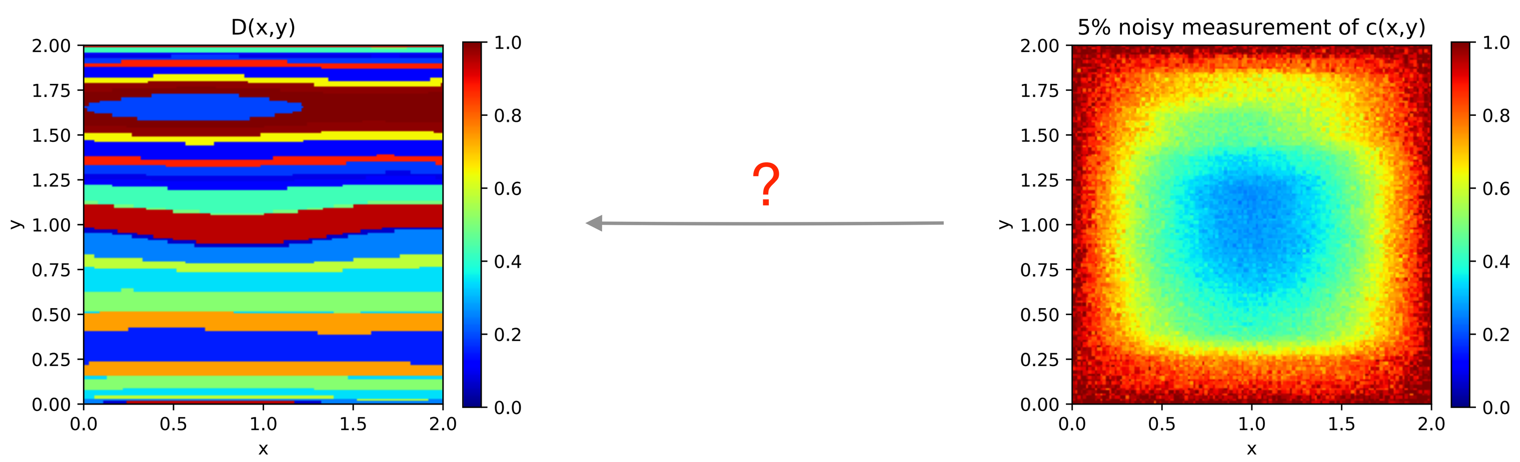
Inverse problems are highly challenging because the problem formulation is usually underdetermined (i.e. **ill-posed**). Difficulties of inferring aquifer properties from measured contaminate concentration include:

1. The measured data is usually **noisy**;
2. The aquifer is highly heterogeneous and there is **discontinuity** between layers;
3. The measured data is not sensitive to small-scale features of the hidden aquifer properties but not vice versa. Therefore, "butterfly effects" are commonly observed in inverse problems.

2 Problem Statement

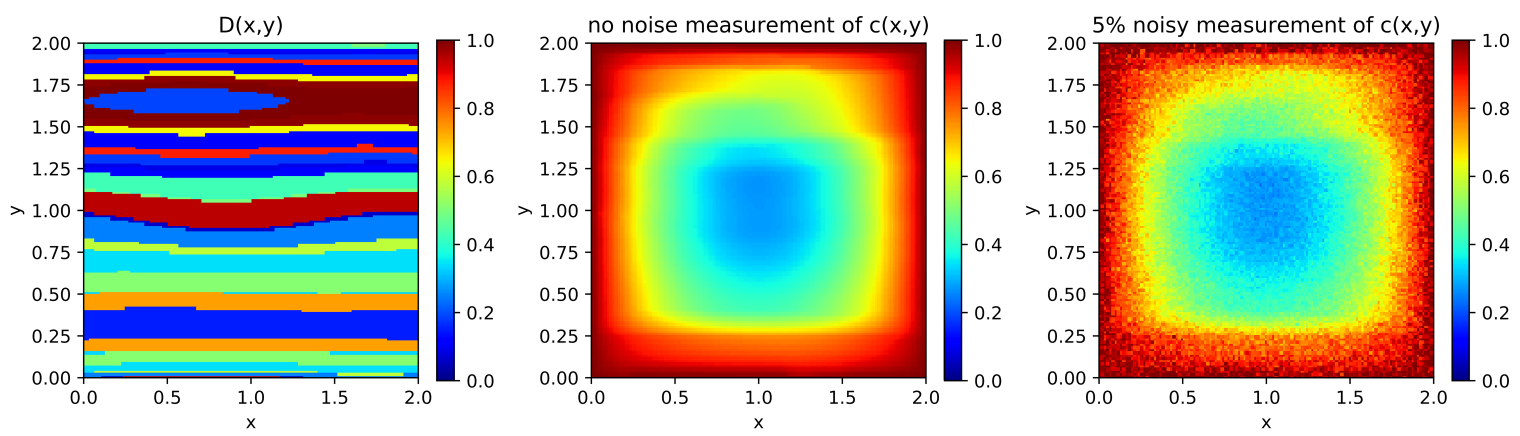
$$\begin{cases} \dfrac{\partial c}{\partial t} = \nabla\cdot(\mathbf{D}\nabla c) - \mathbf{v}\cdot\nabla c + f, \\ c(\mathbf{x}, t=0) = c_0(\mathbf{x}), \\ c(\mathbf{x}\in\partial\Omega, t) = c_b(t). \end{cases}$$

$c(\mathbf{x},t)$: contaminant concentration; $\mathbf{D}(\mathbf{x})$: dispersion coefficient; $\mathbf{v}(\mathbf{x})$: groundwater velocity; $f(\mathbf{x},t)$: contaminant source; Ω: the domain of consideration.



3 Data Generation

1. Select representative dispersion fields $\mathbf{D}(\mathbf{x})$ generated from the Stanford Geostatistical Modeling Software (SGeMS) with laterally correlated heterogeneity;
2. Use forward PDE solver to obtain contaminant concentration map $c(\mathbf{x},t)$;
3. Add noise in the concentration map to mimic the real measurement data $\bar{c}(\mathbf{x},t)$.

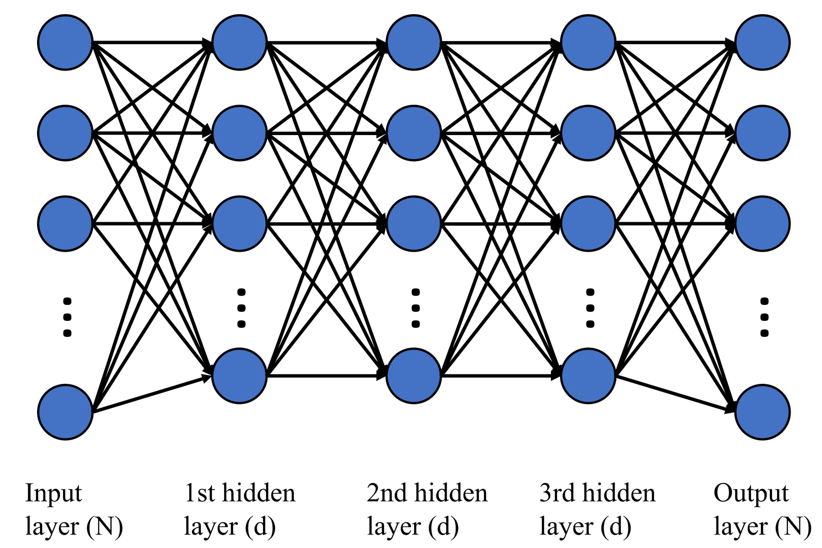


4 Methods

- **Gradient Descent Optimization**

$$\min_{D\in\mathbb{R}^{N_x\times N_y}} J(D) = \sum_{i=2}^{N_x-1}\sum_{j=2}^{N_y-1}\Bigg[\frac{\mathbf{c}^1_{ij}-\mathbf{c}^0_{ij}}{\Delta t} + u_{ij}\frac{\mathbf{c}^0_{i+1,j}-\mathbf{c}^0_{i-1,j}}{2h} + v_{ij}\frac{\mathbf{c}^0_{i,j+1}-\mathbf{c}^0_{i,j-1}}{2h} - \frac{1}{2}(f_i^1+f_i^0) - \frac{D_{i+1/2,j}(\mathbf{c}^0_{i+1,j}-\mathbf{c}^0_{ij}) - D_{i-1/2,j}(\mathbf{c}^0_{ij}-\mathbf{c}^0_{i-1,j})}{h^2} - \frac{D_{i,j+1/2}(\mathbf{c}^0_{i,j+1}-\mathbf{c}^0_{ij}) - D_{i,j-1/2}(\mathbf{c}^0_{ij}-\mathbf{c}^0_{i,j-1})}{h^2}\Bigg]^2.$$

- **Neural Network Optimization**



We adopt *tanh* as the activation function and optimize the neural network parameters θ by backpropagation. The network is constructed as:

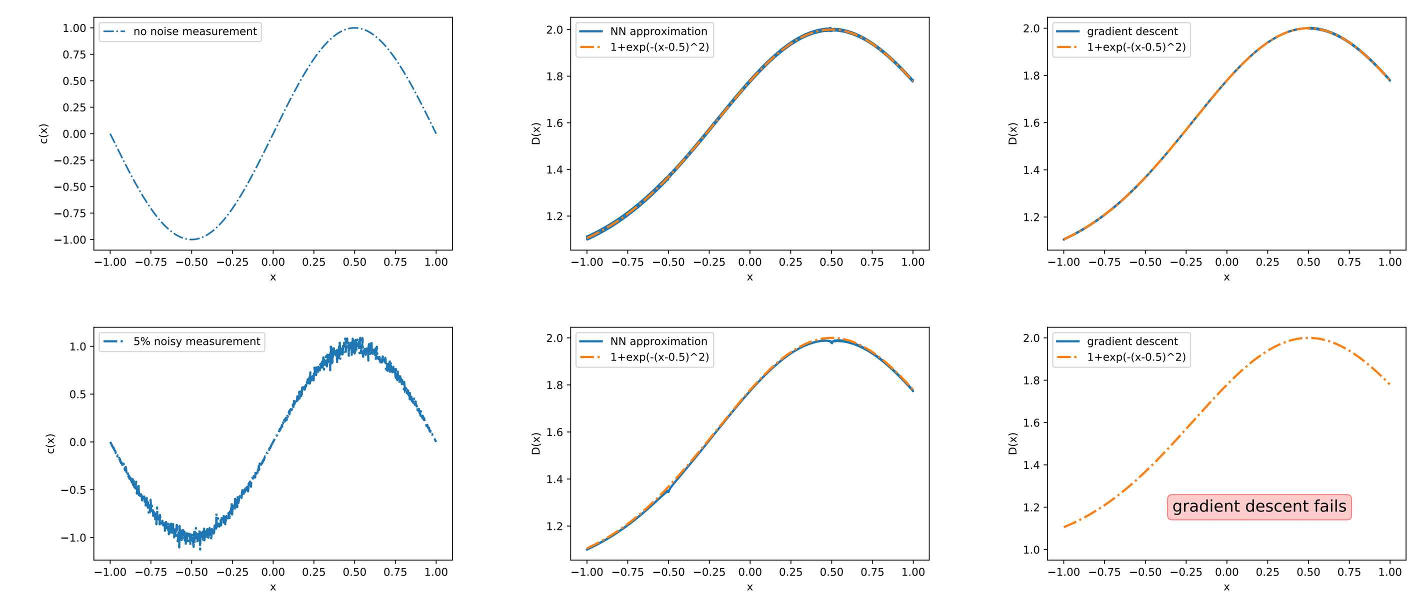
- Number of neurons in each hidden layer: $d = 40$ (20 for 1D);
- Number of hidden layers: $n_l = 3$;
- Size of input and output layer: $N = 128\times 128$ (1000 for 1D);

- **Step 1:** Randomly initialize NN parameters;
- **Step 2:** Compute the hypothesis $h_\theta(\mathbf{x})$;
- **Step 3:** Compute the loss function;
- **Step 4:** If the loss is larger than the tolerance, update the neural network parameters by gradient decent following the backpropagation rules;
- **Step 5:** Go back to step 2 until the loss is smaller than the tolerance.

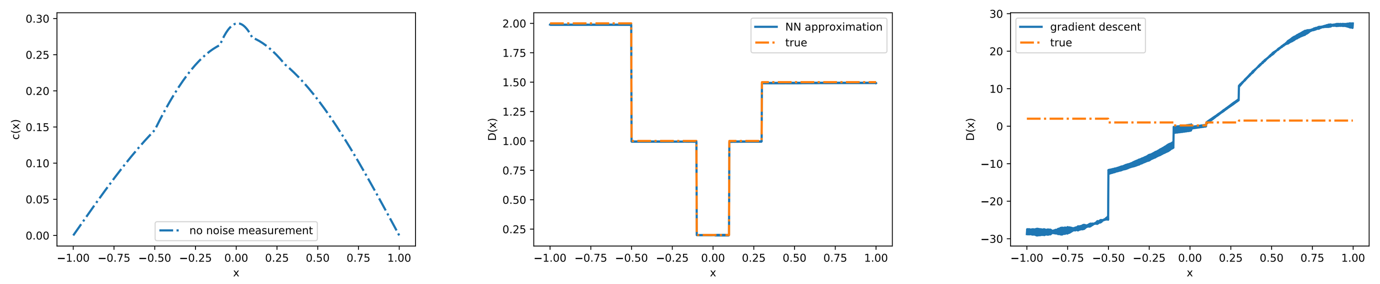
$$\min_\theta \mathbf{loss}(\theta) = \sum_{i=2}^{N_x-1}\sum_{j=2}^{N_y-1}\Bigg[\frac{\mathbf{c}^1_{ij}-\mathbf{c}^0_{ij}}{\Delta t} + u_{ij}\frac{\mathbf{c}^0_{i+1,j}-\mathbf{c}^0_{i-1,j}}{2h} + v_{ij}\frac{\mathbf{c}^0_{i,j+1}-\mathbf{c}^0_{i,j-1}}{2h} - \frac{1}{2}(f_i^1+f_i^0) - \frac{h_\theta(x_{i+1/2,j})(\mathbf{c}^0_{i+1,j}-\mathbf{c}^0_{ij}) - h_\theta(x_{i-1/2,j})(\mathbf{c}^0_{ij}-\mathbf{c}^0_{i-1,j})}{h^2} - \frac{h_\theta(x_{i,j+1/2})(\mathbf{c}^0_{i,j+1}-\mathbf{c}^0_{ij}) - h_\theta(x_{i,j-1/2})(\mathbf{c}^0_{i,j}-\mathbf{c}^0_{i,j-1})}{h^2}\Bigg]^2.$$

5 Experiments and Results

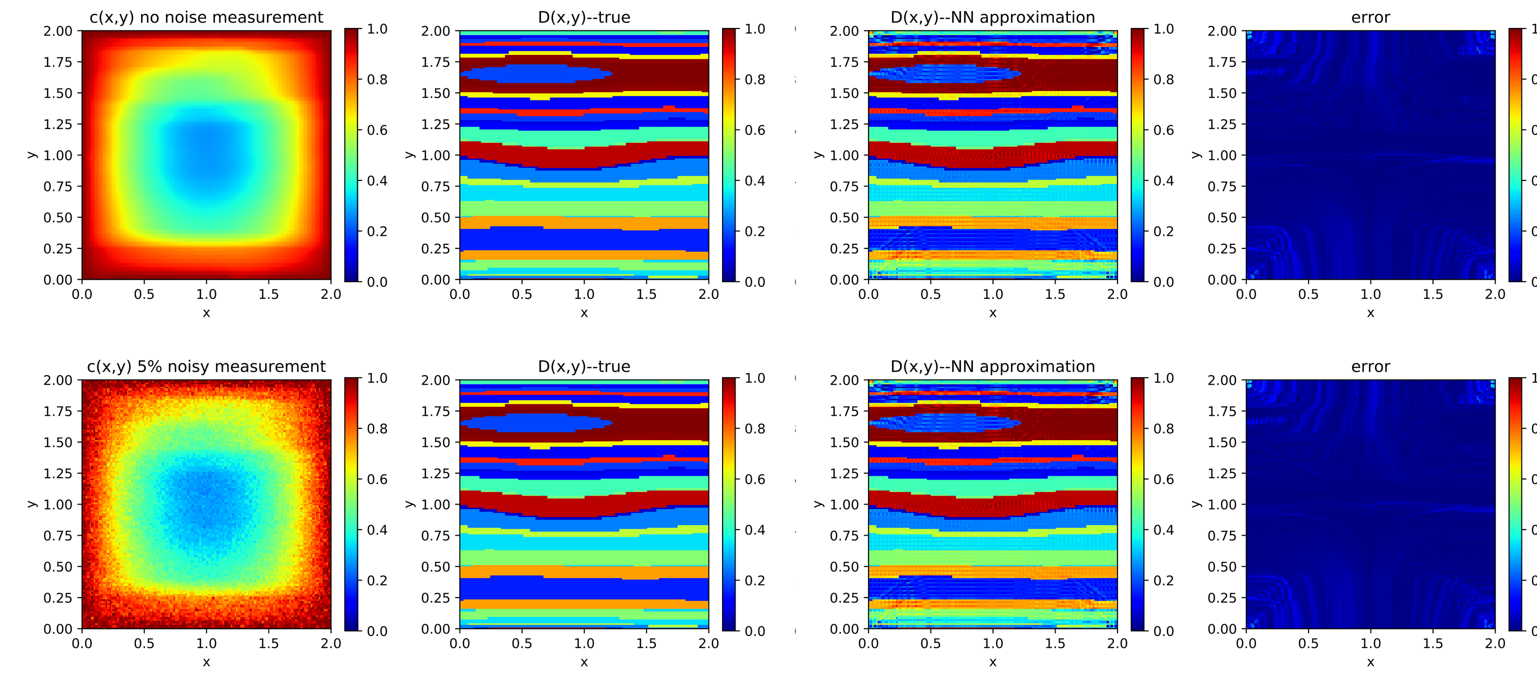
- **Case 1: one dimensional continuous $D(x)$**



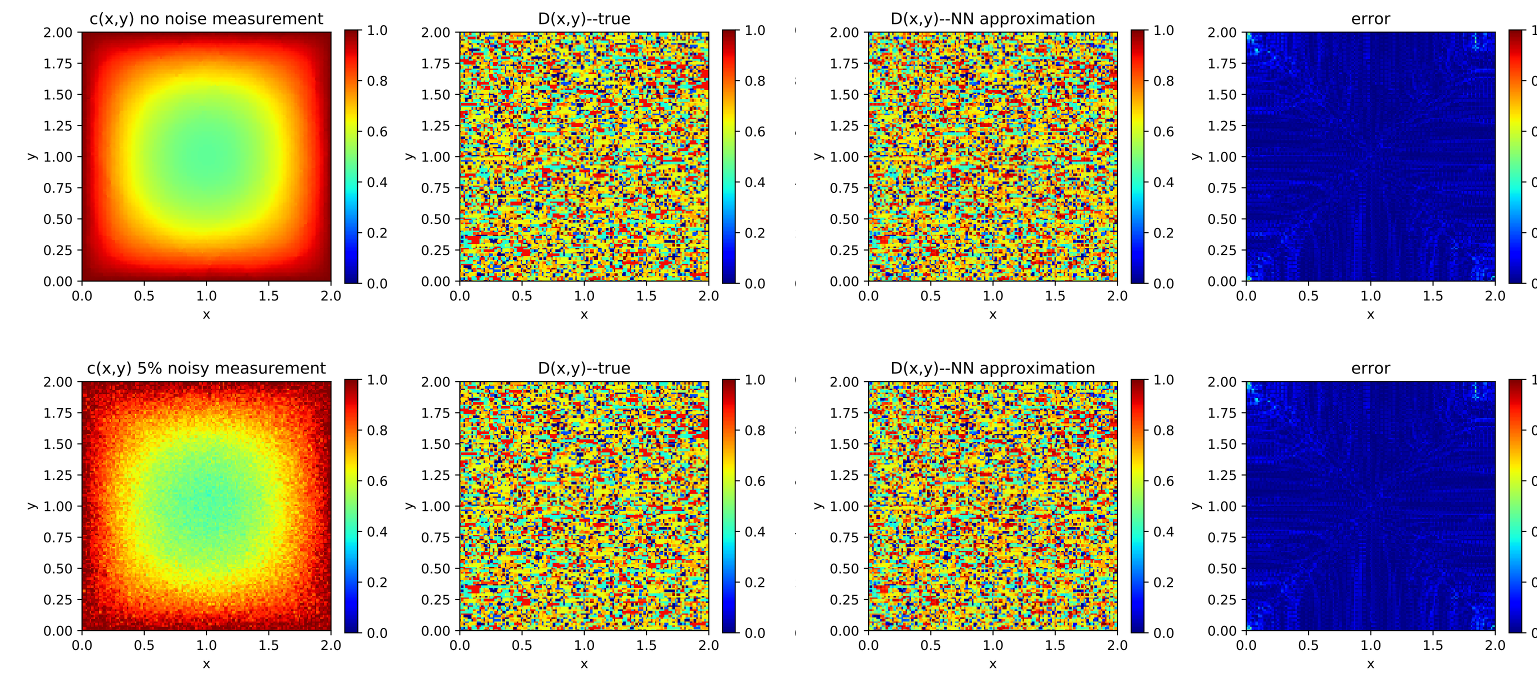
- **Case 2: one dimensional discontinuous $D(x)$**



- **Case 3: two dimensional layered $\mathbf{D}(\mathbf{x})$**



- **Case 4: two dimensional mosaic $\mathbf{D}(\mathbf{x})$**



6 Conclusion and Future Work

We employed a **physics-aware** framework for inverse problems based on neural networks and automatic differentiation. Compared to conventional optimization tools, this framework is more stable and powerful, especially for noisy measurements and heterogeneous targets. Our numerical examples show the robustness of this framework in heterogeneous aquifer property identification, even with noisy contaminate concentration measurements.

Future work: 1) We want to model the complete time-dependent advection-dispersion problem and use this framework to infer other aquifer properties as well; 2) Comparison between our NN approximation and the popular adjoint method from the geophysics community will be included; 3) Sensitivity analysis and error estimation will be conducted to quantify the uncertainty under such framework in practical problems.