Neural Inverse Operators for Solving PDE Inverse Problems

Yunan Yang

June 20, 2023

Advanced Fellow, Institute for Theoretical Studies, ETH Zürich
(Starting July 1st) Assistant Professor, Department of Mathematics, Cornell University

BIRS Scientific Machine Learning Workshop, June 18-23, 2023

The Collaborators

Roberto Molinaro

Björn Engquist

Siddhartha Mishra

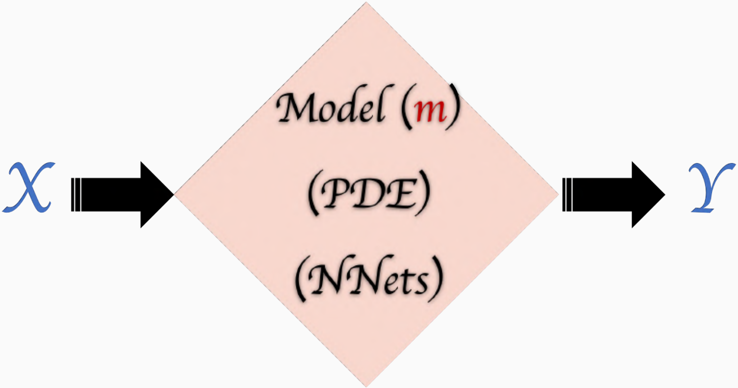
The paper `https://arxiv.org/pdf/2301.11167.pdf` is to appear at ICML 2023.

The modeling step (the forward problem)

Well-Known Inverse Problems:

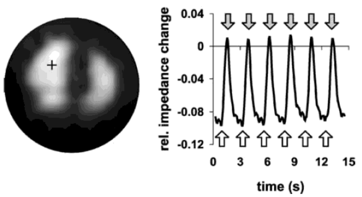
Locate Earthquake Source, Image the Black Hole, X-ray/CT/Ultrasound

General “Inverse Problems”



Inverse data matching problems aim at finding m such that the predicted outputs $(X, F(m))$ match given measured data (X, Y).

Calderón's Problem (Electrical Impedance Tomography, EIT)



— Wikipedia

$$\begin{cases} \nabla \cdot (a(x)\nabla u) = 0, & x \in \Omega \\ u(x) = \psi, & x \in \partial\Omega \end{cases}$$

Given "Dirichlet-to-Neumann" map

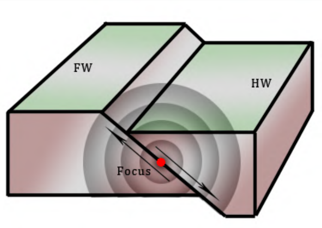
$$\Lambda_a : \mathcal{H}^{1/2}(\partial\Omega) \longrightarrow \mathcal{H}^{-1/2}(\partial\Omega)$$
$$\Lambda_a : \quad \psi \quad \longrightarrow a\nabla u_\psi \cdot \mathbf{n},$$

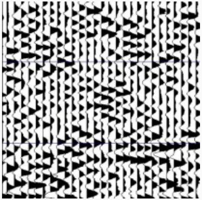
the goal is to find

$$a(x), \quad x \in \Omega.$$

Kohn, R. V., & Vogelius, M. (1987). Relaxation of a variational method for impedance computed tomography. CPAM.

Waveform Inversion (FWI)





— Wikipedia

$$\begin{cases} \color{red}{m(\mathbf{x})}\dfrac{\partial^2 u(\mathbf{x},t)}{\partial t^2} - \triangle u(\mathbf{x},t) = s(\mathbf{x},t) \\ \text{Zero i.c. in half-space } \Omega \\ \text{Neumann b.c. on } \partial\Omega \end{cases}$$

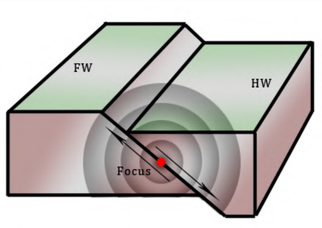
$\color{red}{m(\mathbf{x}) = \dfrac{1}{c(\mathbf{x})^2}}$, $c(\mathbf{x})$ is the wave velocity

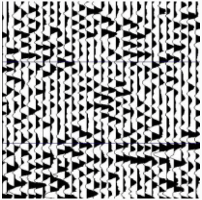
Given $u(x_r, y_r, z = 0, t)$ the goal is to find

$$\color{red}{m(x)}, \quad x \in \Omega.$$

Tarantola, A. (2005). Inverse problem theory and methods for model parameter estimation. SIAM.

Helmholtz Equation Based Inversion





— Wikipedia

$$\begin{cases} \triangle u + \omega^2 m(x) u = s(x, \omega) \\ \text{Neumann b.c. on } \partial\Omega \end{cases}$$

$m(\mathbf{x}) = \dfrac{1}{c(\mathbf{x})^2}$, $c(\mathbf{x})$ is the wave velocity

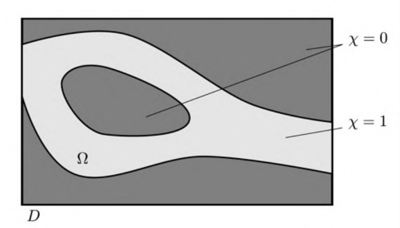
Given $u(x_r, y_r, z = 0; \omega)$ the goal is to find

$$m(x), \quad x \in \Omega.$$

Colton, David L., Rainer Kress. Inverse acoustic and electromagnetic scattering theory. Vol. 93. Berlin: Springer, 1998.

Topological (Shape) Optimization

The fluid (gas) domain in flow channel design problems



Sato et al. (2019). A topology optimization method in rarefied gas flow problems using the Boltzmann eqn.

$$\begin{cases} \frac{\partial f}{\partial t} + v \cdot \nabla_x f = -\frac{\rho}{k_n}\left(f - f^{eq}(\rho, \boldsymbol{u}, T)\right) \\ f(0, x, v) = f_0 \text{ on } D \\ f = f^{eq}(\rho_b, \boldsymbol{u}_b, T_b) \text{ on } \partial\Omega \end{cases}$$

The goal is to find

$$\Omega$$

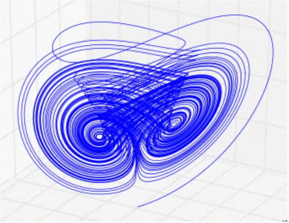
that minimizes an objective function

$$\int_t \int_{\Omega_e} \int_{\mathbb{R}^3} r(v) f(t, x, v)\, dv\, dx\, dt,$$

on some evaluation domain Ω_e.

Learning the Dynamics

"Chen" System [Chen-Ueta, 1999]



Y.-Nurbekyan-Negrini-Martin-Pasha, 2023. Optimal transport for parameter identification of chaotic dynamics via invariant measures. SIADS.

A general parameterized dynamical system may take the form

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} = v(x, y, z; \underbrace{\sigma, \rho, \beta}_{\theta}) \approx v(\mathbf{x}, \theta)$$

where $v \approx v(\cdot, \theta)$ can be

- polynomials,
- basis functions,
- neural networks, and so on,

where θ corresponds to

- expansion coefficients,
- neural network weights, etc.

Image Processing

f_ϵ

u

Denoising, Deblurring, Blind Deconvolution (nonlinear)...

$$f_\epsilon = A(\sigma)u + \epsilon$$

where $A(\sigma)$ could be

- Identity I (denoising)
- Known Kernel K (deblurring)
- Unknown Kernel $A(\sigma)$ (blind deconvolution, nonlinear)

The Forward Model $F(m)$

F is given; we just find m (e.g., PDEs).

- Pro: We know the best (exact) forward problem!
- Con: The forward and inverse problems are so nonlinear!

OR

F is not known; we are free to choose (e.g., XXX-net).

- Pro: The freedom to modify it to a “better” map
  - Over-Parametrization;
  - Model Extension;
  - Model Reduction.
- Con: Trial and error to build the model

How to Solve $F(m) = g$

- Linear Inverse Problem, i.e., $Am = g$
  (often combined with numerical linear algebra)
  - Direct Method
  - Iterative Method
  - Optimization-Based Method (e.g., least-squares min)

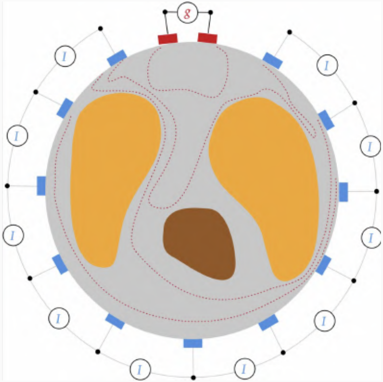
How to Solve $F(m) = g$

- Linear Inverse Problem, i.e., $Am = g$
  (often combined with numerical linear algebra)
  - Direct Method
  - Iterative Method
  - Optimization-Based Method (e.g., least-squares min)

- Nonlinear Inverse Problem, $F(m) = g$
  - Direct Method (challenging to construct) (**in today's talk**)
  - Iterative Method (e.g., nonlinear GMRES)
  - Optimization Method

Learn a Direct Inverse Map

Example: Calderón's Problem



$$\begin{cases} \nabla \cdot (a(x)\nabla u) = 0, & x \in \Omega \\ u(x) = \psi, & x \in \partial\Omega \end{cases}$$

Given Dirichlet-to-Neumann map

$\Lambda_a : \mathcal{H}^{1/2}(\partial\Omega) \longrightarrow \mathcal{H}^{-1/2}(\partial\Omega)$

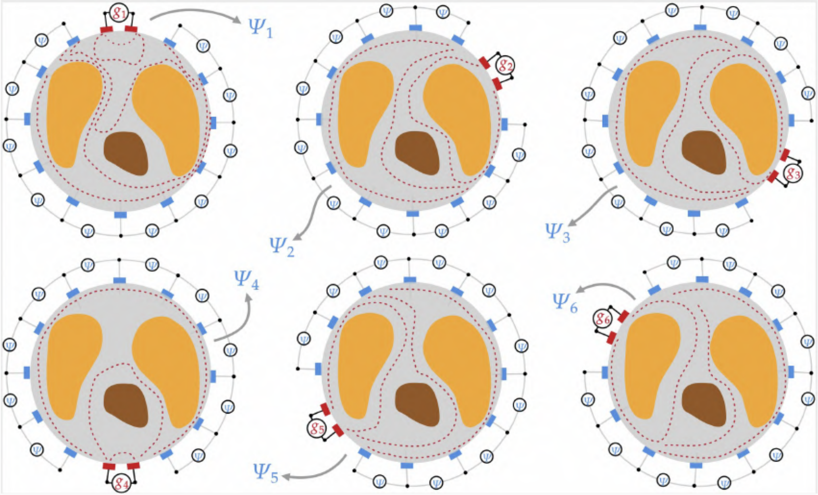
$\Lambda_a : \quad \psi \quad \longrightarrow a\nabla u_\psi \cdot \mathbf{n},$

the goal is to find

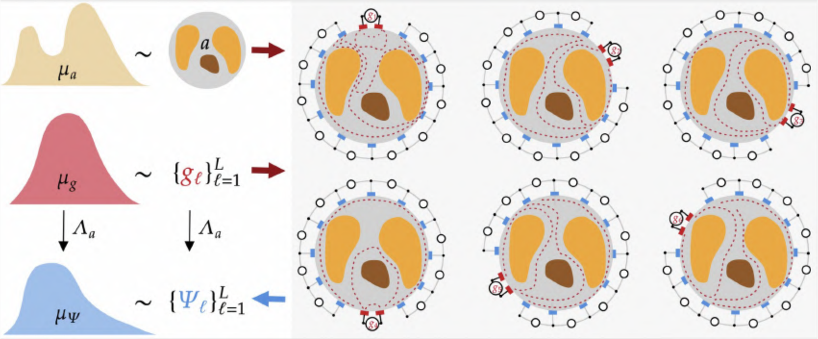
$$a(x), \quad x \in \Omega.$$

In suitable settings, it is provable that there exists an inverse problem with (log-) stability.

Recall that the data is an **operator** on the continuous level



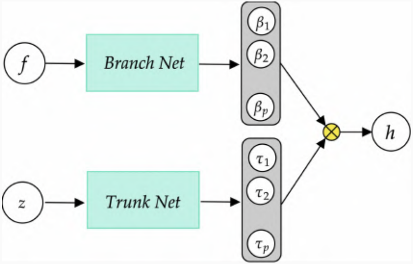
The Training Data Acquisition



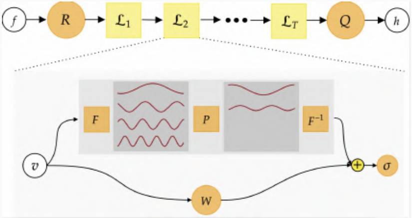
We provide $\left(a^{(i)}, \{\Psi_\ell^{(i)}\}_\ell\right)$ as the training data for $i = 1, \dots, n$ number of different parameter samples with $a^{(i)} \sim \mu_a$.

$$\{\Psi_\ell^{(i)}\}_\ell \approx \mu_\Psi = \Lambda_{a^{(i)}} \sharp \mu_g \,, \quad \mu_g \text{ fixed.}$$

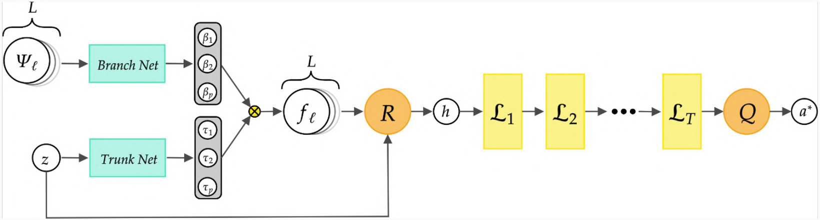
The Neural Network Architecture



NN_1: DeepONet



NN_2: FNO



DeepONet FNO

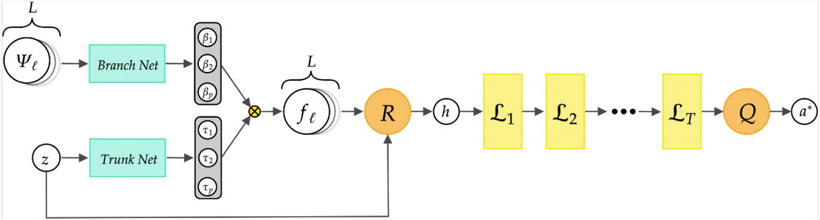
The proposed Neural Inverse Operator (NIO)

An Intuition:

DeepONet: $\{\Psi_\ell\} \mapsto \{f_\ell\}$ (analogy: $\underline{\{a\nabla u_\psi \cdot \mathbf{n}\} \text{ on } \partial\Omega}$ to $\underline{\{u_\psi\} \text{ on } \Omega}$)

FNO: $\{f_\ell\} \mapsto a$ (analogy: $\underline{\{u_\psi\} \text{ on } \Omega}$ to $\underline{a \text{ on } \Omega}$)

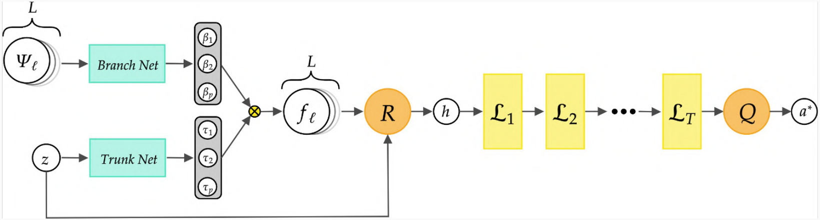
The Neural Network Architecture



DeepONet (NN_1) FNO (NN_2)

$$NIO\left(\Lambda_a \sharp \mu_g\right) = NN_2\left(NN_1 \sharp \underbrace{\left(\Lambda_a \sharp \mu_g\right)}_{\mu_\Psi} \right) \to a.$$

The Neural Network Architecture

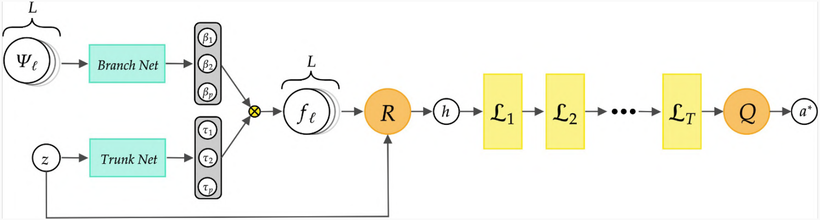


DeepONet (NN_1) FNO (NN_2)

$$NIO\left(\Lambda_a\sharp\mu_g\right) = NN_2\left(NN_1\sharp\underbrace{\left(\Lambda_a\sharp\mu_g\right)}_{\mu_\Psi}\right) \to a.$$

One concern: NN does not know $\{\Psi_\ell\}$ are samples of μ_Ψ and similarly $\{f_\ell\}$ are samples of an underlying distribution.

The Neural Network Architecture



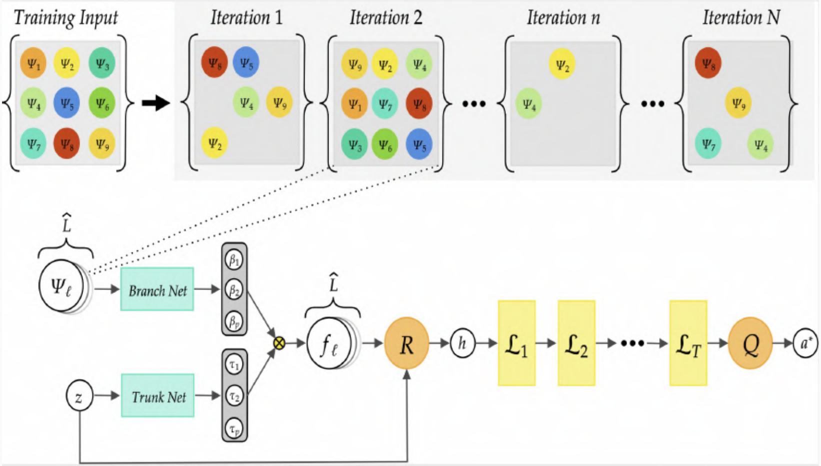
DeepONet (NN_1) FNO (NN_2)

$$NIO\left(\Lambda_a \sharp \mu_g\right) = NN_2\left(NN_1 \sharp \underbrace{\left(\Lambda_a \sharp \mu_g\right)}_{\mu_\Psi}\right) \to a.$$

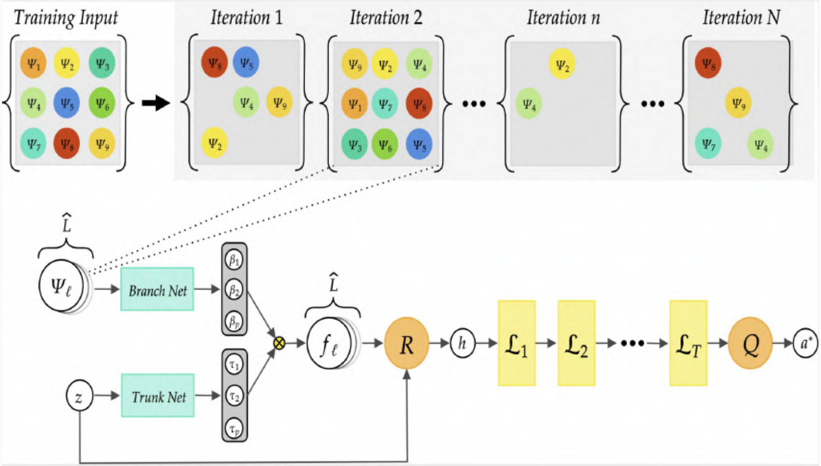
One concern: NN does not know $\{\Psi_\ell\}$ are samples of μ_Ψ and similarly $\{f_\ell\}$ are samples of an underlying distribution.

We want: (1) permutation invariant; (2) different a can have different L; (3) testing data can have a different L

The Training Scheme — Bagging — "Randomized Batching"



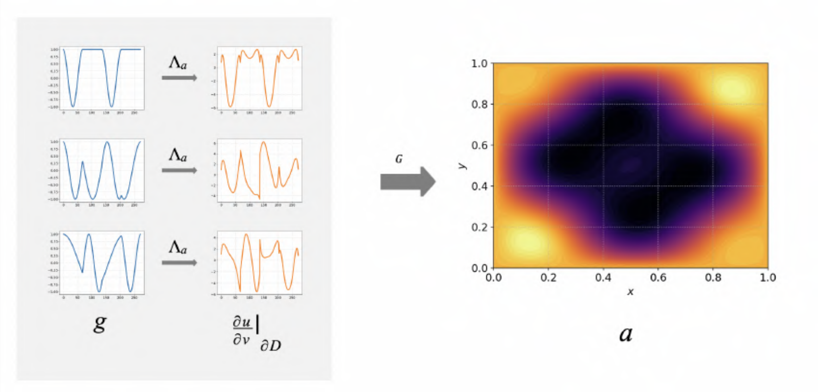
The Training Scheme — Bagging — “Randomized Batching”



Rich theoretical analysis in Bagging from statistical learning.

Numerical Results

EIT Examples



$$\begin{cases} \nabla \cdot (a(x)\nabla u) = 0, & x \in \Omega \\ u(x) = \psi, & x \in \partial\Omega \end{cases}$$

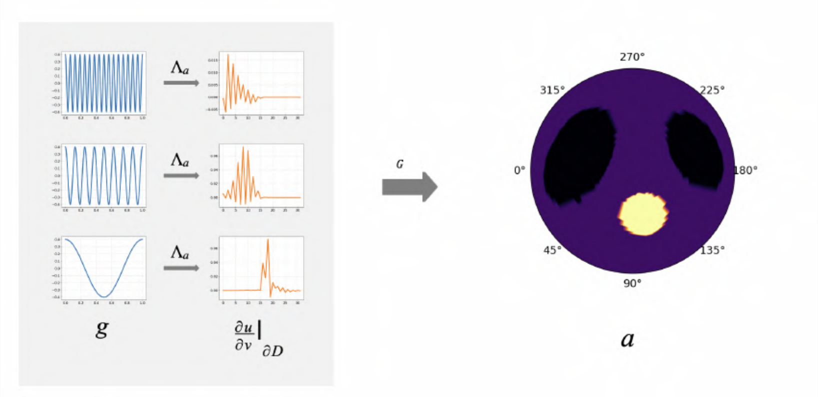
Given DtN map

$$\Lambda_a : \mathcal{H}^{1/2}(\partial\Omega) \longrightarrow \mathcal{H}^{-1/2}(\partial\Omega)$$

$$\Lambda_a : \qquad \psi \qquad \longrightarrow a\nabla u_\psi \cdot \mathbf{n},$$

the goal is to find $a(x)$, $x \in \Omega$.

EIT Examples



$$\begin{cases} \nabla \cdot (a(x)\nabla u) = 0, & x \in \Omega \\ u(x) = \psi, & x \in \partial\Omega \end{cases}$$

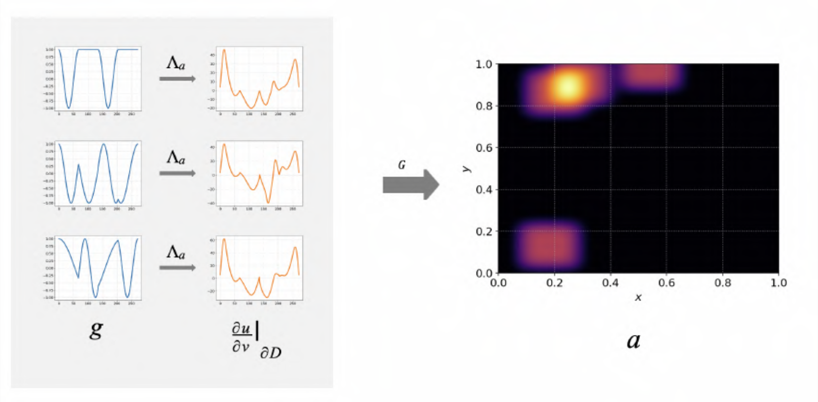
Given DtN map

$\Lambda_a : \mathcal{H}^{1/2}(\partial\Omega) \longrightarrow \mathcal{H}^{-1/2}(\partial\Omega)$

$\Lambda_a : \quad \psi \quad \longrightarrow a\nabla u_\psi \cdot \mathbf{n}$,

the goal is to find $a(x)$, $x \in \Omega$.

EIT Examples



$$\begin{cases} \nabla \cdot (a(x)\nabla u) = 0, & x \in \Omega \\ u(x) = \psi, & x \in \partial\Omega \end{cases}$$

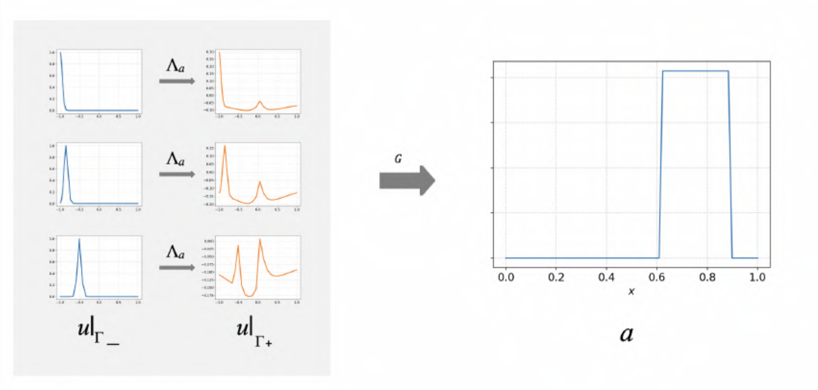
Given DtN map

$\Lambda_a : \mathcal{H}^{1/2}(\partial\Omega) \longrightarrow \mathcal{H}^{-1/2}(\partial\Omega)$

$\Lambda_a : \quad \psi \quad \longrightarrow a\nabla u_\psi \cdot \mathbf{n},$

the goal is to find $a(x)$, $x \in \Omega$.

RTE Inversion Examples



$$v \cdot \nabla_z u(z,v) + \sigma_a(z) u(z,v) = \frac{1}{\epsilon} a(z) Q[u],\ z \in D$$

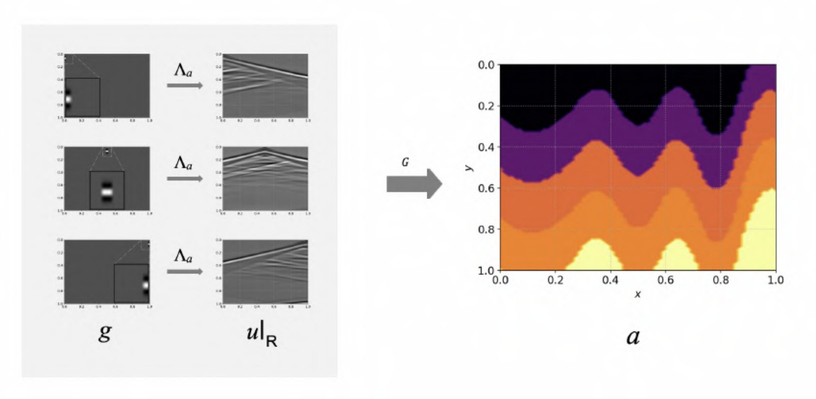
$$u(z,v) = \phi(z,v),\ z \in \Gamma_-$$

Given the Albedo operator

$$\Lambda_a : L^1(\Gamma_-) \mapsto L^1(\Gamma_+)$$

$$\Lambda_a : u\big|_{\Gamma_-} = \phi \mapsto u\big|_{\Gamma_+}$$

Wave Inversion Results



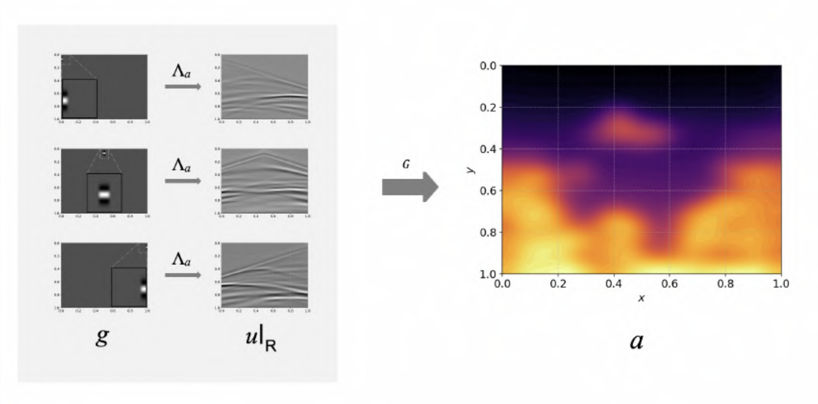
$$u_{tt}(t,z) + a(z)^2 \Delta u = s, \qquad (z,t) \in D \times [0,T],$$

Given the *Source-to-Receiver* (StR) operator,

$$\Lambda_a : L^2([0,T] \times D) \mapsto L^2([0,T]; X_R),$$

$$\Lambda_a : s \mapsto u\big|_{[0,T]\times R},$$

Wave Inversion Results



$$u_{tt}(t,z) + a(z)^2 \Delta u = s,$$
$$(z,t) \in D \times [0,T],$$

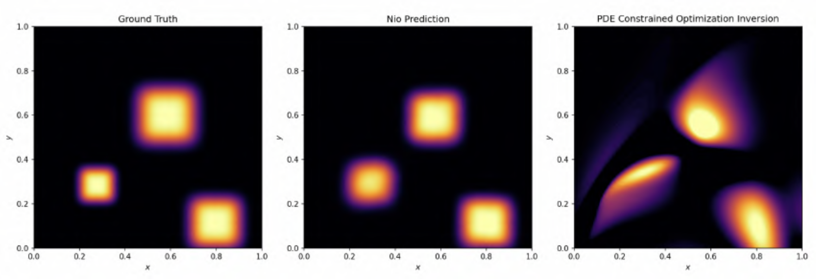
Given the *Source-to-Receiver* (StR) operator,

$$\Lambda_a : L^2([0,T] \times D) \mapsto L^2([0,T];X_R),$$
$$\Lambda_a : s \mapsto u\big|_{[0,T]\times R},$$

Testing Performance Comparison

	DONet		FCNN		NIO	
	$L^1 \downarrow$	$L^2 \downarrow$	$L^1 \downarrow$	$L^2 \downarrow$	$L^1 \downarrow$	$L^2 \downarrow$
EIT Trigonometric	1.97%	2.36%	1.49%	1.82%	**0.85%**	**1.05%**
EIT Heart&Lungs	0.95%	3.69%	0.27%	1.62%	**0.18%**	**1.16%**
EIT Inclusion Detection	3.83%	7.41%	2.53%	7.55%	**1.07%**	**2.94%**
Optical Imaging	2.35%	4.35%	1.46%	3.71%	**1.1%**	**2.9%**
Seismic Imaging - CurveVel - A	3.98%	5.86%	**2.65%**	5.05%	2.71%	**4.71%**
Seismic Imaging - Style - A	3.82%	5.17%	3.12%	4.63%	**3.04%**	**4.36%**

Compare with PDE-Constrained Optimization



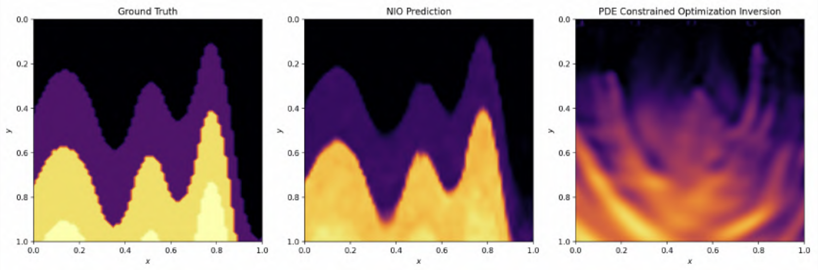
The ill-posed Calderón Problem (inverse Darcy flow)

$$\begin{cases} \nabla \cdot ({\color{red}a(x)}\nabla u) = 0, & x \in \Omega \\ u(x) = \psi, & x \in \partial\Omega \end{cases}$$

$$\min_{a \in A(D)} \sum_{i=1}^{L} \mathrm{dist}(\mathcal{F}_i(a), d_i^{\mathrm{obs}}) \quad s.t. \text{ PDE constraints}$$

Difficulty in PDE-Constrained optimization: high wavenumber (i.e., edges)

Compare with PDE-Constrained Optimization



The full waveform inversion (FWI) problem

$$u_{tt}(t,z) + a^2(z)\Delta u = s,$$
$$(z,t) \in D \times [0,T],$$

$$\min_{a \in A(D)} \sum_{i=1}^{L} \mathrm{dist}(\mathcal{F}_i(a), d_i^{\mathrm{obs}}) \quad s.t. \text{ PDE constraints}$$

Difficulty in PDE-Constrained optimization: local minima

Conclusions

Summary and Future Directions

Summary

- We consider a large class of PDE-based inverse problems that are "solvable" only when providing a data operator (e.g., DtN, Albedo).

Summary and Future Directions

Summary

- We consider a large class of PDE-based inverse problems that are "solvable" only when providing a data operator (e.g., DtN, Albedo).
- Learn mappings from operators to functions to solve inverse problem.

Summary and Future Directions

Summary

- We consider a large class of PDE-based inverse problems that are "solvable" only when providing a data operator (e.g., DtN, Albedo).
- Learn mappings from <u>operators</u> to <u>functions</u> to solve inverse problem.
- Motivated by the inverse-problem nature, we proposed a novel architecture, termed **Neural Inverse Operator** (NIO), based on a composition of DeepONet and FNO.

Summary and Future Directions

Summary

- We consider a large class of PDE-based inverse problems that are "solvable" only when providing a data operator (e.g., DtN, Albedo).
- Learn mappings from <u>operators</u> to <u>functions</u> to solve inverse problem.
- Motivated by the inverse-problem nature, we proposed a novel architecture, termed **Neural Inverse Operator** (NIO), based on a composition of DeepONet and FNO.
- Random batching to inform FNO the input is an *empirical distribution*.

Summary and Future Directions

Summary

- We consider a large class of PDE-based inverse problems that are "solvable" only when providing a data operator (e.g., DtN, Albedo).
- Learn mappings from operators to functions to solve inverse problem.
- Motivated by the inverse-problem nature, we proposed a novel architecture, termed **Neural Inverse Operator** (NIO), based on a composition of DeepONet and FNO.
- Random batching to inform FNO the input is an *empirical distribution*.

Summary and Future Directions

Summary

- We consider a large class of PDE-based inverse problems that are "solvable" only when providing a data operator (e.g., DtN, Albedo).
- Learn mappings from operators to functions to solve inverse problem.
- Motivated by the inverse-problem nature, we proposed a novel architecture, termed **Neural Inverse Operator** (NIO), based on a composition of DeepONet and FNO.
- Random batching to inform FNO the input is an *empirical distribution*.

Future Directions

1. Conduct theoretical analysis on the generalization error

Summary and Future Directions

Summary

- We consider a large class of PDE-based inverse problems that are "solvable" only when providing a data operator (e.g., DtN, Albedo).
- Learn mappings from operators to functions to solve inverse problem.
- Motivated by the inverse-problem nature, we proposed a novel architecture, termed **Neural Inverse Operator** (NIO), based on a composition of DeepONet and FNO.
- Random batching to inform FNO the input is an *empirical distribution*.

Future Directions

1. Conduct theoretical analysis on the generalization error
2. How do different ways of representing Λ_a affect convergence?

Summary and Future Directions

Summary

- We consider a large class of PDE-based inverse problems that are "solvable" only when providing a data operator (e.g., DtN, Albedo).
- Learn mappings from operators to functions to solve inverse problem.
- Motivated by the inverse-problem nature, we proposed a novel architecture, termed **Neural Inverse Operator** (NIO), based on a composition of DeepONet and FNO.
- Random batching to inform FNO the input is an *empirical distribution*.

Future Directions

1. Conduct theoretical analysis on the generalization error
2. How do different ways of representing Λ_a affect convergence?
3. How does the PDE inverse problem stability improve using statistical learning-type of algorithms?

Acknowledgment

Thanks for the workshop organizers!

All my collaborators.