ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Seminar for Applied Mathematics SAM

Neural Inverse Operators for Solving PDE Inverse Problems

R. Molinaro and Y. Yang and B. Engquist and S. Mishra

Research Report No. 2023-10
January 2023

Seminar für Angewandte Mathematik
Eidgenössische Technische Hochschule
CH-8092 Zürich
Switzerland

Neural Inverse Operators for Solving PDE Inverse Problems

Roberto Molinaro [1] **Yunan Yang** [2] **Björn Engquist** [3] **Siddhartha Mishra** [1 4]

Abstract

A large class of inverse problems for PDEs are only well-defined as mappings from operators to functions. Existing operator learning frameworks map functions to functions and need to be modified to learn inverse maps from data. We propose a novel architecture termed Neural Inverse Operators (NIOs) to solve these PDE inverse problems. Motivated by the underlying mathematical structure, NIO is based on a suitable composition of DeepONets and FNOs to approximate mappings from operators to functions. A variety of experiments are presented to demonstrate that NIOs significantly outperform baselines and solve PDE inverse problems robustly, accurately and are several orders of magnitude faster than existing direct and PDE-constrained optimization methods.

1. Introduction.

Partial differential equations (PDEs) are ubiquitous as mathematical models in the sciences and engineering (Evans, 2010). Often, solving PDEs entails solving the so-called *forward problem*. That is, given inputs such as initial and boundary conditions, coefficients, and sources, compute (observables of) the solution of the PDE. However, in many important contexts in applications, one is instead interested in solving the so-called *inverse problem* (Isakov, 2017). That is, given measurements of (observables of) the solution of a PDE, infer the underlying inputs.

A large class of such inverse problems takes the following abstract form: given observables as operators (mappings between function spaces), infer the underlying input coefficient (functions) of the associated PDE. A prototypical example is the well-studied *Calderón Problem* (Uhlmann, 2009) that arises in electrical impedance tomography (EIT) in medical imaging. Here, the observable is the Dirichlet-to-Neumann (DtN) operator that maps the voltage on the boundary to the current, and one is interested in inferring the underlying conductivity field. A related example is inverse wave scattering for geophysical applications. Other examples include optical tomography (Lai et al., 2019) where the observable is the so-called *Albedo operator*, and one needs to infer the scattering and absorption coefficients of the underlying medium. Another prominent example arises in *seismic imaging* in geophysics (Yilmaz, 2011) where the observable is the source-to-receiver (StR) operator, and the task at hand is to infer the underlying sub-surface properties such as wave velocity or material density. In many of these examples, the solution to the resulting inverse problem exists. It is unique and stable if and only if the inverse problem is posed as a *mapping from operators to functions*.

Given the nonlinear nature of most of these inverse problems, analytical solution formulas are not available except in some simple cases. Instead, iterative numerical algorithms based on the PDE-constrained optimization are commonly used to approximate the solution (Chavent, 2010). These algorithms repeatedly apply the forward and adjoint PDE solvers to converge to the unknown coefficient. However, a large number of iterations might be necessary, which leads to prohibitively high computational costs as numerous calls to PDE solvers are very expensive, particularly in two and three space dimensions. Moreover, these iterative algorithms can be sensitive to the choice of initial coefficient. Given these factors, the design of alternative approaches to solving inverse problems is imperative.

Data-driven approximation of PDEs is rapidly emerging as a powerful paradigm. Most of the available results pertain to *forward problems* for PDEs. A particularly popular framework is *operator learning*, where one seeks to learn the underlying forward solution operator of the PDE from data. Existing approaches to operator learning include Deep operator networks (DeepONet) (Lu et al., 2021; Mao et al., 2020; Cai et al., 2021) and their variants, as well as the so-called neural operators (Kovachki et al., 2021b), which include the widely used Fourier Neural Operators (FNO) and its variants (Li et al., 2021b; Pathak et al., 2022). Graph neural network-based algorithms (Boussif et al., 2022; Brandstetter et al., 2022) are also emerging as an alternative.

Given the widespread success of operator learning and other deep learning-based algorithms in the context of forward

[1]Seminar for Applied Mathematics (SAM), ETH, Zürich. [2]Institute for Theoretical Studies (ITS), ETH Zürich. [3]Department of Mathematics and the Oden Institute, The University of Texas at Austin, USA. [4]ETH AI Center, ETH, Zürich.

problems for PDEs, it is natural to investigate their utility in learning the solutions of the corresponding inverse problems from data. However, this task is very challenging as existing operator learning algorithms map functions to functions. On the other hand, the aforementioned inverse problems are only well-defined as maps from operators to functions. Hence, one needs non-trivial modifications of existing operator learning architectures to handle inverse problems. This is precisely the rationale for the current paper, where our main contributions are the following.

- Motivated by the underlying mathematical structure of the considered class of inverse problems, we propose a novel architecture, termed as *Neural Inverse Operators* (NIOs), for learning solutions of these inverse problems from data. NIOs compose (stack) two existing architectures, DeepONet and FNO, in a novel manner to map operators to functions.

- We test NIOs extensively on a suite of problems, including the Calderón problem in EIT, inverse wave scattering for object detection, reconstructing the absorption and scattering coefficients in optical tomography, and seismic wave migration to infer sub-surface properties. We show that NIO outperforms baselines on all these benchmarks and provides a fast, robust and accurate solution to the underlying inverse problem.

2. A Class of Inverse Problems.

2.1. Mathematical Framework.

Let $D \subset \mathbb{R}^d$ be a bounded open set, with (smooth) boundary ∂D. Let $T > 0$ and $\Omega = D$ or $\Omega = D \times (0, T)$, depending on whether the PDE is time-(in)dependent. Correspondingly, $\partial\Omega = \partial D$ or $\partial\Omega = \partial D \times (0, T)$, respectively. Let $a \in \mathcal{A}(D)$, with $\mathcal{A}$ denoting a suitable function space over D, be a coefficient. Then, an abstract PDE can be written as

$$\mathcal{D}_a(u) = s, \quad \mathcal{B}(u) = g, \tag{1}$$

where $u \in \mathcal{U}(\Omega)$ is the solution, $s \in \mathcal{S}(\Omega)$ is the source term and $g \in \mathcal{G}(\partial\Omega)$ is the boundary condition, for the PDE (1). Here, $\mathcal{D}_a : \mathcal{U} \mapsto \mathcal{F}$ and $\mathcal{B} : \mathcal{U} \mapsto \mathcal{G}$ are the differential and boundary operators, respectively and $\mathcal{U}, \mathcal{S}, \mathcal{G}$ are suitable function spaces, defined over their respective domains.

The *forward problem* for the abstract PDE (1) amounts to the following: given the coefficient $a \in \mathcal{A}$, source term $s \in \mathcal{S}$ and boundary condition $g \in \mathcal{G}$, find the solution $u \in \mathcal{U}$ of the PDE (1). Often, one is interested in not only the solution itself but also *observables* of the solution, which can be measured in practice. Since measurements are usually easier to perform at boundaries, a particularly relevant class of such observables are given by the following *boundary observation operator*,

$$\Lambda_a : \mathcal{G}(\partial\Omega) \mapsto \mathcal{H}(\partial\Omega), \tag{2}$$

which maps the boundary data $g \in \mathcal{G}(\partial\Omega)$ to a measurement $\Lambda_a(g) = h(u) \in \mathcal{H}(\partial\Omega)$, a function space on $\partial\Omega$. Thus, for a fixed coefficient a (and source s), solving the forward problem amounts to solving the PDE (1), with a given boundary data g to obtain the solution u and then post-processing u to compute the boundary observation operator $h(u) = \Lambda_a(g)$. Hence, one can rewrite the forward problem associated with the PDE (1) to obtain the map,

$$\mathcal{F} : \mathcal{A}(D) \mapsto \mathcal{L}\left(\mathcal{G}(\partial\Omega), \mathcal{H}(\partial\Omega)\right), \; a \mapsto \mathcal{F}(a) = \Lambda_a, \tag{3}$$

where Λ_a is the boundary observation operator (2) and $\mathcal{L}(X, Y)$ denotes continuous operators between function spaces X and Y.

In practice, one is often interested in the *inverse problem* associated with the PDE (1). For instance, in tomography (imaging), one needs to infer the unknown coefficient a from some measurements of the solution u. In general, this problem is *ill-posed*, and a single instance (or small number) of boundary conditions g and measurements $h(u)$ of the corresponding solutions u, do not suffice in inferring the underlying coefficient a. Instead, many deep mathematical results have provided suitable frameworks where such inverse problems can be well-posed. The inverse map for the forward problem (3) takes the form

$$\mathcal{F}^{-1} : \mathcal{L}\left(\mathcal{G}(\partial\Omega), \mathcal{H}(\partial\Omega)\right) \mapsto \mathcal{A}(D), \; \Lambda_a \mapsto a = \mathcal{F}^{-1}(\Lambda_a), \tag{4}$$

The rigorous guarantee of the existence and, more importantly, the uniqueness of this inverse map $\mathcal{F}^{-1}$, for a large class of PDEs, is a crowning achievement of the mathematical theory of inverse problems (Isakov, 2017). Moreover, one can also show Lipschitz or Hölder-*stability* of the inverse problem by proving estimates of the form,

$$\|\mathcal{F}^{-1}(a) - \mathcal{F}^{-1}(\overline{a})\|_{\mathcal{L}} \sim \|a - \overline{a}\|_{\mathcal{A}}^{\alpha}, 0 < \alpha \leq 1. \tag{5}$$

In some cases, the right-hand side of the above stability estimate is replaced by a logarithm of $\|a - \overline{a}\|_{\mathcal{A}}$, which only guarantees (weak) logarithmic stability.

After presenting this abstract framework, we provide four concrete examples of PDE inverse problems (see **SM** Figures 3-8 for illustrations) to which this abstract framework applies.

2.2. Calderón Problem (EIT).

Let the coefficient $0 < a \in C^2(D)$ represent the conductivity of the underlying medium (domain $D \subset \mathbb{R}^d$) and the associated PDE (1) is the following elliptic equation,

$$\begin{aligned} -\nabla \cdot \Big(a(z)\nabla u\Big) &= 0, \quad z \in D, \\ u(z) &= g(z), \quad z \in \partial D, \end{aligned} \tag{6}$$

with Dirichlet boundary value $g \in H^{\frac{1}{2}}(\partial D)$ representing the voltage and the current source term is $s = 0$. The associated boundary observation operator Λ_a is the well-known *Dirichlet-to-Neumann* (DtN) map,

$$\begin{aligned} &\Lambda_a : H^{1/2}(\partial D) \mapsto H^{-1/2}(\partial D), \\ &\Lambda_a[g] = a\frac{\partial u}{\partial \nu}\Big|_{\partial D}, \ \forall g \in H^{1/2}(\partial D), \end{aligned} \tag{7}$$

which maps the input voltage g into the current $a(z)\frac{\partial u}{\partial \nu} = a\,\nabla u \cdot \nu$ (with ν being the unit outward normal vector) at the boundary and u is the solution of (6).

The inverse problem, often referred to as the Calderón problem, constitutes the basis of EIT (Uhlmann, 2009). It aims to find the conductivity a of the medium, given different measurements of the DtN (voltage-to-current) pairs. Thus, this inverse problem falls into the considered abstract formalism and the inverse map (4) is given by,

$$\begin{aligned} &\mathcal{F}^{-1} : \mathcal{L}\left(H^{1/2}(\partial D), H^{-1/2}(\partial D)\right) \mapsto C^2(D), \\ &\mathcal{F}^{-1} : \Lambda_a \mapsto a = \mathcal{F}^{-1}(\Lambda_a), \end{aligned} \tag{8}$$

with $\mathcal{L}(\cdot,\cdot)$ denoting the corresponding bounded linear operators. This inverse problem is shown to be well-defined and (logarithmic-) stable (Clop et al., 2010).

2.3. Inverse Wave Scattering.

In many applications of interest, wave propagation in the frequency domain is used to infer material properties of the medium, modelled by the squared slowness $0 < a \in L^\infty(D)$. The associated PDE is the Helmholtz equation,

$$\begin{aligned} -\Delta u - \omega^2 a(z)u = 0, \quad z \in D, \\ u(z) = g(z), \quad z \in \partial D, \end{aligned} \tag{9}$$

for some frequency ω and Dirichlet boundary condition $g \in H^{\frac{1}{2}}(\partial D)$. The resulting boundary observation operator is again the Dirichlet-to-Neumann (DtN) map

$$\begin{aligned} &\Lambda_a : H^{1/2}(\partial D) \mapsto H^{-1/2}(\partial D), \\ &\Lambda_a[g] = \frac{\partial u}{\partial \nu}\Big|_{\partial D}, \ \forall g \in H^{1/2}(\partial D), \end{aligned} \tag{10}$$

where u is the solution to (9) with the coefficient a. The corresponding inverse problem amounts to inferring the wave coefficient a from the DtN map (10). Thus, it can be formulated similar to the inverse map (8). Its well-posedness and stability have been demonstrated for the Helmholtz equation in (Nachman, 1988) and references therein.

2.4. Radiative Transport and Optical Imaging.

In optical imaging or tomography, the material properties of the medium $D \subset \mathbb{R}^d$ are expressed in terms of the scattering and absorption coefficients, $0 \le a, \sigma_a \in C(D)$. The associated PDE is the well-known radiative transport equation (RTE) for the particle density $u(z, v)$ at location $z \in D$ and velocity $v \in V \subset \mathbb{R}^d$, given by

$$\begin{aligned} v \cdot \nabla_z u(z,v) + \sigma_a(z)u(z,v) = \frac{1}{\varepsilon}a(z)\mathcal{Q}[u], \quad z \in D, \\ u(z,v) = \phi(z,v), \quad z \in \Gamma_-, \end{aligned} \tag{11}$$

where $\mathcal{Q}[u] = \int k(v,v')u(z,v')dv' - u(z,v)$ is the collision term, ε is the Knudsen number, $\Gamma_\pm = \{(z,v) \in \partial D \times V : \pm n_z \cdot v \ge 0\}$ are the inflow (outflow) boundaries and n_z is the unit outer normal vector at $z \in \partial D$. Thus, the input to this problem is provided by the particle density, $u_{\Gamma_-} \in L^1(\partial D)$, prescribed on the inflow boundary. The associated boundary observation operator Λ_a defined in (2) is the so-called *Albedo* operator,

$$\Lambda_a : L^1(\Gamma_-) \mapsto L^1(\Gamma_+), \ \Lambda_a : u\big|_{\Gamma_-} = \phi \mapsto u\big|_{\Gamma_+}, \tag{12}$$

that maps the incident boundary values on Γ_- to the observed boundary values on the outflow boundary Γ_+.

The corresponding inverse problem aims to infer the medium properties characterized by the scattering and absorption coefficients a, σ_a from the measurements of the Albedo operator. It leads to the following inverse map,

$$\begin{aligned} &\mathcal{F}^{-1} : \mathcal{L}\left(L^1(\Gamma_-), L^1(\Gamma_+)\right) \mapsto C(D), \\ &\mathcal{F}^{-1} : \Lambda_a \mapsto a = \mathcal{F}^{-1}(\Lambda_a). \end{aligned} \tag{13}$$

The well-posedness and Lipschitz-stability of this inverse map were shown in (Bal & Jollivet, 2008).

2.5. Seismic Imaging.

Seismic imaging is widely used in geophysics to infer and reconstruct sub-surface material properties for various applications such as CO_2 storage monitoring and seismic hazard assessment. Given a domain $D \subset \mathbb{R}^d$, we are interested in reconstructing the velocity coefficient $0 < a \in L^\infty(D)$ by sending in acoustic waves from the top boundary into the medium and measuring the response in the time domain. The associated PDE is the acoustic wave equation,

$$u_{tt}(t,z) + a^2(z)\Delta u = s, \ (z,t) \in D \times [0,T]. \tag{14}$$

with a time-dependent source term s. Here, u is the pressure variation. The wave equation is supplemented with zero initial conditions, i.e., $u(:,0) = u_t(:,0) = 0$ and suitable boundary conditions. In particular, *sources* are placed on a subset of the boundary, $\Sigma \subset \partial D \times [0,T]$, resulting in a source term of the form $s = g\delta_\Sigma$ (with δ_S being the Dirac measure concentrated on a set. These waves are transmitted, reflected, and refracted through the medium. The resulting signal is recorded at a set of receivers on the boundary

given by $\mathcal{R} \subset D \times [0,T]$. Hence, the boundary observation operator (2) in this case becomes the *Source-to-Receiver* (StR) operator (Symes, 2009),

$$\Lambda_a : L^2(\Sigma) \mapsto H^1(\mathcal{R}),\ \Lambda_a : g \mapsto u\big|_{\mathcal{R}}. \tag{15}$$

The inverse problem that underpins seismic imaging is

$$\begin{aligned} &\mathcal{F}^{-1} : \mathcal{L}\left(L^2(\Sigma), H^1(\mathcal{R})\right) \mapsto L^\infty(D), \\ &\mathcal{F}^{-1} : \Lambda_a \mapsto a = \mathcal{F}^{-1}(\Lambda_a), \end{aligned} \tag{16}$$

with Λ_a is the StR operator (15). Thus, seismic imaging aims to infer the subsurface spatial medium properties from spatial-temporal StR signals. This process is also termed as *migration*, or *Full waveform Inversion* (FWI) in the literature (Deng et al., 2021). There have been studies on the well-posedness of the inverse problem for the wave equation (14) (Liu & Oksanen, 2016; Stefanov et al., 2016; Caday et al., 2019) although they do not directly apply to the setting considered here.

3. Neural Inverse Operators.

In this section, we present the neural network architecture for the proposed Neural Inverse Operators (NIOs).

3.1. Learning Task and Challenges.

All four examples described in the previous section were particular instances of the abstract framework summarized in (4). Thus, the solution of the inverse problem (4) boils down to inferring (learning) the inverse map $\mathcal{F}^{-1}$ from relevant data. Given sufficient training data in the form of pairs $\left(\Lambda_a, \mathcal{F}^{-1}(\Lambda_a)\right)$ (or given the injectivity of the forward map, data in the form of pairs (a, Λ_a)), we aim to learn the inverse map $\mathcal{F}^{-1}$ and evaluate it on *test* (unseen) data. This task is very challenging on account of the following factors:

- The learning task requires us to learn *mappings from operators to functions* for $\mathcal{F}^{-1}$ defined in (4).
- The inputs to the inverse map $\mathcal{F}^{-1}$ (4) are specified on the boundaries $\partial\Omega$ whereas the output is the coefficient a, defined in the interior of the underlying domain D. Thus, there is a mismatch in the domains of the inputs and outputs for the inverse map $\mathcal{F}^{-1}$.
- In general, the inverse map $\mathcal{F}^{-1}$ (4) may only be weakly stable, for instance, either in terms of small values of the Hölder exponent α in (5) or even only logarithmic-stable. In these cases, the learning task can be very sensitive to noises from the input, and additional regularization terms might be necessary.

3.2. Existing Operator Learning Architectures.

Before proposing a suitable architectures for learning the inverse map $\mathcal{F}^{-1}$ (4), we briefly summarize existing operator learning architectures to examine whether they can be useful in this context. To this end, let $D \subset \mathbb{R}^{d_x}, U \subset \mathbb{R}^{d_u}$ and $\mathcal{X} = \mathcal{X}(D)$ and $\mathcal{Y} = \mathcal{Y}(U)$ be suitable function spaces. Then, a **DeepONet** (Lu et al., 2021) is the operator, $\mathcal{N}^{\mathrm{DON}} : \mathcal{X} \to \mathcal{Y}$, given by

$$\mathcal{N}^{\mathrm{DON}}(\overline{u})(y) = \sum_{k=1}^{p} \beta_k(\overline{u})\tau_k(y), \quad \overline{u} \in \mathcal{X},\ y \in U, \tag{17}$$

where the **branch-net** $\boldsymbol{\beta}$ is a neural network that maps $\mathcal{E}(\overline{u}) = (\overline{u}(x_1), \ldots, \overline{u}(x_m)) \in \mathbb{R}^m$, evaluations of the input $\overline{u}$ at sensor points $\boldsymbol{x} := (x_1, \ldots, x_m) \in D$, to $\mathbb{R}^p$:

$$\boldsymbol{\beta} : \mathbb{R}^m \to \mathbb{R}^p,\ \mathcal{E}(\overline{u}) \mapsto (\beta_1(\mathcal{E}(\overline{u})), \ldots, \beta_p(\mathcal{E}(\overline{u})), \tag{18}$$

and the **trunk-net** $\boldsymbol{\tau}(y) = (\tau_1(y), \ldots, \tau_p(y))$ is another neural network mapping,

$$\boldsymbol{\tau} : U \to \mathbb{R}^p, \quad y \mapsto (\tau_1(y), \ldots, \tau_p(y)). \tag{19}$$

Thus, a DeepONet combines the branch net (as coefficient functions) and trunk net (as basis functions) to create a mapping between functions.

On the other hand, a Fourier neural operator (**FNO**) $\mathcal{N}^{\mathrm{FNO}}$ (Li et al., 2021a) is a composition

$$\mathcal{N}^{\mathrm{FNO}} : \mathcal{X} \mapsto \mathcal{Y} : \quad \mathcal{N}^{\mathrm{FNO}} = Q \circ \mathcal{L}_T \circ \cdots \circ \mathcal{L}_1 \circ R. \tag{20}$$

It has a "lifting operator" $\overline{u}(x) \mapsto R(\overline{u}(x), x)$, where R is represented by a (shallow) neural network $R : \mathbb{R}^{d_u} \times \mathbb{R}^d \to \mathbb{R}^{d_v}$ where d_u is the number of components of the input function, d is the dimension of the domain and d_v is the "lifting dimension" (a hyperparameter). The operator Q is a non-linear projection, instantiated by a shallow neural network, such that $v^{L+1}(x) \mapsto \mathcal{N}^{\mathrm{FNO}}(\overline{u})(x) = Q\left(v^{L+1}(x)\right)$. Each *hidden layer* $\mathcal{L}_\ell : v^\ell(x) \mapsto v^{\ell+1}(x)$ is of the form

$$v^{\ell+1}(x) = \sigma\left(W_\ell \cdot v^\ell(x) + b_\ell(x) + \left(K_\ell v^\ell\right)(x)\right),$$

with $W_\ell \in \mathbb{R}^{d_v \times d_v}$ a weight matrix (residual connection), $b_\ell(x) \in \mathbb{R}^{d_v}$ a bias function, σ an activation function, and the *non-local Fourier layer*,

$$K_\ell v^\ell = \mathcal{F}_N^{-1}\left(P_\ell(k) \cdot \mathcal{F}_N v^\ell(k)\right),$$

where $\mathcal{F}_N v^\ell(k)$ denotes the (truncated)-Fourier coefficients of the discrete Fourier transform (DFT) of $v^\ell(x)$, computed based on the given N grid values in each direction. Here, $P_\ell(k) \in \mathbb{C}^{d_v \times d_v}$ is a complex Fourier multiplication matrix indexed by $k \in \mathbb{Z}^d$, and $\mathcal{F}_N^{-1}$ denotes the inverse DFT.

Both operator learning frameworks (DeepONet and FNO) and their variants map *functions to functions*. Hence, they cannot directly be used to learn the inverse map $\mathcal{F}^{-1}$ (4), which maps *operators to functions*. Therefore, we need to modify and adapt these architectures to learn the inverse map. The following sections present our proposed approach.

3.3. A Motivating (Formal) Calculation.

We start by providing a heuristic motivation for our proposed architecture to learn the inverse map (4). To this end and for definiteness, we consider the inverse wave scattering problem for the Helmholtz equation (9), presented in section 2.3. Given the domain $D \subset \mathbb{R}^d$, we consider the following eigenvalue problem with Neumann boundary conditions,

$$\begin{aligned} -\Delta\varphi_k &= \lambda_k\varphi_k, \quad \forall z \in D. \\ \frac{\partial\varphi_k}{\partial\nu}\big|_{\partial D} &= 0, \quad \int_D \varphi_k dz = 0. \end{aligned} \tag{21}$$

By standard PDE theory (Evans, 2010), there exist eigenvalues $0 \leq \lambda_k \in \mathbb{R}$ for $k \in \mathbb{N}$, and the corresponding eigenfunctions $\{\varphi_k\}$ form an orthonormal basis for $L^2(D)$. We fix $K \in \mathbb{N}$ sufficiently large and without loss of generality, we assume $\omega = 1$ in the Helmholtz equation (9) to consider the following Dirichlet boundary-value problems,

$$\begin{aligned} -\Delta u_k - a(z)u_k = 0, \quad z \in D,\ 1 \leq k \leq K, \\ u(z) = g_k(z), \quad z \in \partial D, \end{aligned} \tag{22}$$

where $g_k = \varphi_k\big|_{\partial D}$. Using (21) and (22), we prove in **SM** B, the following formal *representation formula* for all $1 \leq k \leq K$,

$$\int_D a u_k \varphi_k dz = \int_D \lambda_k u_k \varphi_k dz - \int_{\partial D} g_k \frac{\partial u_k}{\partial \nu} d\sigma(z). \tag{23}$$

The formula (23) can be used to construct an approximation to the coefficient $a \in L^2(D)$ in the following manner. Writing $a \approx \sum_{\ell=1}^{K} a_\ell \varphi_\ell$ (using the orthonormality of φ's) for K sufficiently large, we can evaluate the coefficients a_ℓ by solving the following Matrix equation for $A = \{a_\ell\}_{\ell=1}^K$,

$$\begin{aligned} \mathbf{C}A = B,\ \mathbf{C}_{k\ell} = \int_D u_k\varphi_k\varphi_l dx, \quad \forall k,l, \\ B_k = \int_D \lambda_k u_k \varphi_k dz + \int_{\partial D} g_k \frac{\partial u_k}{\partial \nu} d\sigma(z), \forall k. \end{aligned} \tag{24}$$

Further setting $\Psi_k = \frac{\partial u_k}{\partial \nu}$, we observe that the formal approximation of the coefficent a relies on the following building blocks,

- **Basis construction:** The operations $\mathcal{B}_k : z \mapsto (\varphi_k(z), \lambda_k)$, $1 \leq k \leq K$, that form a basis. Note that they are independent of the coefficient a.
- **PDE Solve:** The operation $\mathcal{E}_k : (g_k, \Psi_k) \mapsto \left(\{u_j^k\}_{j=1}^K, \int_{\partial D} g_k \Psi_k d\sigma(z)\right)$ that amounts to (approximately) inferring the coefficients of the solution u_k of the Helmholtz equation (22), given the Dirichlet g_k and Neumann Ψ_k boundary values. A part of the right-hand side term B_k is also appended to this operation. Once the coefficients u_j^k are computed, the approximation u_k to the solution of (22) is readily computed in terms of the basis $\{\varphi_k\}$ by setting $u_k \approx \sum_{j=1}^{K} u_j^k \varphi_j$.
- **Mode Mixing:** The previous two operations were restricted to individual modes, i.e., to each k, for $1 \leq k \leq K$. However, to construct the coefficients $\mathbf{C}_{kl}$ in (24), we need to mix different modes. One way to do so is through multiplication. We denote this operation by $\mathcal{M} : \left(\{\varphi_k\}_{k=1}^K, \{u_k\}_{k=1}^K\right) \mapsto \left(\{u_k\varphi_k\varphi_\ell\}_{k,\ell=1}^K, \{\lambda_k u_k \varphi_k\}_{k=1}^K\right)$.
- **Matrix Inversion:** In the final step, we need to build the Matrix $\mathbf{C}$ in (24) and (approximately) invert it. This operation can be summarized by $\mathcal{I} : \left(\{u_k\varphi_k\varphi_\ell\}_{k,\ell=1}^K, \{\lambda_k u_k\varphi_k\}_{k=1}^K\right) \mapsto \sum_{j=1}^K a_j\varphi_j$, with $A = \{a_j\}$ being the solution of (24).

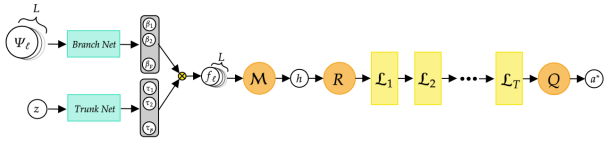


Figure 1. Schematic representation of the Fully-Connected Neural Inverse Operator (NIO) architecture. See (25) for notation.

3.4. Fully-Connected Neural Inverse Operator.

The formal approximation of the inverse map $\mathcal{F}^{-1}$ (4) for the Helmholtz equation by formulas (23)-(24) cannot be directly used in practice as one cannot solve the PDE (22) without knowing the coefficient a. However, the building blocks enumerated above motivate either an iterative fixed-point procedure or, in our case, a learning algorithm approximating $\mathcal{F}^{-1}$ from data. To this end, we observe that the basis construction $z \mapsto \varphi_k(z)$ amounts to a particular instantiation of a trunk-net (19) of a DeepONet. Similarly, the PDE solve map $\mathcal{E}_k : (g_k, \Psi_k) \mapsto \{u_j^k\}_{j=1}^K$ is a particular instance of the application of a branch-net (18) of a DeepONet. Moreover, they can be combined in a DeepONet (17) to approximate the solutions u_k of the PDE (22). However, a DeepONet (17) is linear in its trunk-net basis functions and thus cannot represent the non-linear mode mixing operator $\mathcal{M}$. Instead, one can do so by passing the outputs of the DeepONet through another (densely) connected neural network. Another possibility is to pass these outputs through the non-linear lifting layer of an FNO (20).

Finally, the inversion operator $\mathfrak{I}$, an operator inversion at the continuous level, can be approximated with an FNO.

These heuristic considerations are generalized to the abstract formalism of the inverse problem (4) and motivate us to propose the composition (stacking) of DeepONets and FNO to result in the following map (also illustrated in Figure 1),

$$\mathcal{N}_{FC} : \begin{pmatrix} z \\ \{\Psi_\ell\}_{\ell=1}^L \end{pmatrix} \overset{\tau,\beta}{\longmapsto} \begin{pmatrix} \{\tau_p(z)\}_{p=1}^P \\ \{\beta_p\}_{p=1}^P \end{pmatrix} \dots \\ \dots \overset{\mathcal{N}^{DON}}{\longmapsto} \{f_\ell(z)\}_{\ell=1}^L \overset{\mathcal{M}}{\longmapsto} h(z) \overset{\mathcal{N}^{FNO}}{\longmapsto} a^*(z), \tag{25}$$

for approximating the abstract inverse map $\mathcal{F}^{-1}$ (4).

In other words, the inputs $z \in D$ and $\Psi_\ell = \Lambda_a(g_\ell)$ (2), for $1 \le \ell \le L$, are fed into the trunk- and branch-nets of a DeepONet $\mathcal{N}^{DON}$ (17), respectively, to create L representations $\{f_\ell\}_{\ell=1}^L$, defined in the interior of the underlying domain. These representations are then *mixed*, for instance, by a dense fully-connected neural network $\mathcal{M}$, to yield a mixed representation $h(z)$, which in turn is passed through an FNO $\mathcal{N}^{FNO}$ (20), resulting in an approximation of the underlying coefficient a^*. We term $\mathcal{N}_{FC}$ as the fully connected *Neural Inverse Operator* (FC-NIO).

We observe that the DeepONet $\mathcal{N}^{DON}$ in FC-NIO (25) is flexible enough to handle inputs defined on the boundary, i.e., Ψ_ℓ, and produce outputs f_ℓ, defined on the interior of the underlying domain. Finally, the injectivity of the boundary observation operator (2) implies that it suffices to provide the output $\Psi_\ell = \Lambda_a(g_\ell)$ rather than the input-output pair (g_ℓ, Ψ_ℓ).

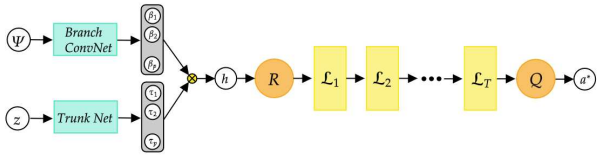


Figure 2. Schematic representation of the Convolutional Neural Inverse Operator architecture. See (27) for notation.

3.5. Convolutional Neural Inverse Operator.

The PDE (22) admits a Green's function representation, (Engquist & Zhao, 2018):

$$u_k(z) = \int_{\partial D} \frac{\partial G}{\partial \nu}(z, y) g_k(y) d\sigma(z), \tag{26}$$

for a suitable Green's function G. Moreover, this Green's function is translation invariant, i.e., $G(z, y) = G(z - y)$, at least for the free-space case. Given this representation, the PDE solve step in section 3.3 allows for a significantly more parsimonious representation by replacing the fully-connected branch-net β (18) by a *convolutional neural network* (CNN). Interpreting the boundary values g_k for $1 \le k \le K$ as the K channels of a CNN allows us to mimic the Green's function representation (26). The same kernel is now used for all the boundary inputs to create approximations to the solution u_k of (22).

Generalizing this representation to the abstract inverse problem (4) results in the following architecture, (also illustrated in Figure 2),

$$\mathcal{N}_{Con} : \begin{pmatrix} z \\ \{\Psi_\ell\}_{\ell=1}^L \end{pmatrix} \overset{\tau,\beta_{con}}{\longmapsto} \begin{pmatrix} \{\tau_p(z)\}_{p=1}^P \\ \{\beta_p\}_{p=1}^P \end{pmatrix} \dots \\ \dots \overset{\mathcal{N}^{DON}}{\longmapsto} h(z) \overset{\mathcal{N}^{FNO}}{\longmapsto} a^*(z), \tag{27}$$

for approximating the abstract inverse map $\mathcal{F}^{-1}$ (4). Note that β_{con} is a convolutional neural network, and the inputs Ψ_ℓ are now mixed through the channel mixing in CNNs. We term this architecture (27) as *Convolutional Neural Inverse Operator* (CNIO). Clearly, CNIO allows for a more sparse representation than FC-NIO. Hence, we expect it to be more expressive for a similar number of model parameters.

4. Empirical Results.

Both the fully-connected (25) and convolutional (27) versions of NIO are designed to approximate the abstract form (4) of the PDE inverse problem as they map operators to functions. We empirically test NIO on benchmark PDE inverse problems below. Based on the design, CNIO is expected to outperform FC-NIO, and this was indeed verified in all experiments we have performed. Hence, we will report only the results with the convolutional version of NIO, i.e., CNIO (27). Moreover, the exact details of the training, as well as the architecture and hyperparameter choices, are presented in **SM** C.

As baselines in the following experiments, we choose two models. First, we consider an end-to-end DeepONet architecture with a CNN as the branch net in (17). In other words, we consider (27) but *ablate* the FNO part at the end by only considering,

$$\mathcal{N}_{Donet} : \begin{pmatrix} z \\ \{\Psi_\ell\}_{\ell=1}^L \end{pmatrix} \overset{\tau,\beta_{con}}{\longmapsto} \begin{pmatrix} \{\tau_p(z)\}_{p=1}^P \\ \{\beta_p\}_{p=1}^P \end{pmatrix} \overset{\mathcal{N}^{DON}}{\longmapsto} a^*(z). \tag{28}$$

Note that channel mixing in the CNN allows for mode mixing to a certain degree in this architecture. Second, we consider a fully convolutional image-to-image neural network architecture (details in **SM** C.1). A variant of this architecture was already used in seismic imaging (full waveform inversion) in (Deng et al., 2021). We have extended this architecture significantly to apply it to the abstract inverse problem (4).

4.1. Calderón Problem for EIT.

We start with the Calderón inverse problem for the elliptic equation (6) on the computational domain $D = [0,1]^2$, with source $s = 0$. The training (and test) data are generated by sampling from a probability distribution on the conductivity coefficient a. Once a sample conductivity is drawn, a set of Dirichlet boundary conditions $\{g_\ell\}_{\ell=1}^L$ are drawn from a probability distribution on the boundary values. For each g_ℓ, the underlying elliptic equation is solved numerically with a standard five-point finite difference scheme and the current, $\Psi_\ell = a\frac{\partial u}{\partial \nu}$, is evaluated on the boundary. We choose the boundary data g_ℓ, for $1 \leq \ell \leq L = 20$ as the boundary values of $\cos(\omega(x\cos(\theta_\ell) + y\sin(\theta_\ell)))$, with $\theta_\ell = \frac{2\pi\ell}{20}$. For the coefficient a, we sample from **trigonometric functions** by setting $a(x,y) = \exp\big(\sum_{k=1}^m c_k \sin(k\pi x)\sin(k\pi y)\big)$, with $m = \mod(\overline{m})$ where $\overline{m} \sim \mathcal{U}([1,5])$ and $\{c_k\} \sim \mathcal{U}([-1,1]^m)$. All the models are trained with 4096 training coefficient samples, and the relative (percentage) test errors (with respect to 2048 test samples) in both L^1 and L^2 norms for CNIO (and the baselines) are presented in Table 1. As the table shows, CNIO is the best-performing model, outperforming the next-best FCNN model by almost halving the errors. Moreover, the total errors are very small (less than 1%) with CNIO. In contrast, we also experimented with FCNIO in this particular example to observe L^1-errors of 5.9%, i.e., almost an order of magnitude bigger than CNIO, reaffirming the intuition that the convolutional version of NIO is much more expressive than the fully-connected variant.

As a second experiment for EIT, we consider a more practical example suggested in (Muller & Siltanen, 2012), where the authors model the EIT imaging of the heart and lungs of a patient using electrodes on the body. This *discontinuous heart and lungs phantom* is depicted in **SM** Figure 4. The underlying domain is the unit circle, and the elliptic equation (6) is solved with a standard finite element scheme. The boundary conditions are given by $g_\ell(\theta) = \frac{1}{2\pi}\exp(i2\pi\theta f_\ell)$, with $\ell = 1,\ldots,32$ and $f = [-16,\ldots,-1,1,14,15,16]$. The coefficient a is modelled by adding 8% white noise to the location, shape, and conductivity of the configuration of heart and lungs shown in **SM** Figure 4. The input of the learning operators is obtained by computing the Fourier transform at frequencies f of the difference between the Neumann trace of the PDE solution with the coefficient a and the one with the unit coefficient $a = 1$. Again, the results presented in Table 1 show that CNIO is the best-performing model and yields very low reconstruction errors, solving this practical problem with high accuracy. In contrast, a traditional direct method such as the D-bar method (Muller & Siltanen, 2012) has a very large error of 8.75% (see **SM** Figure 13) for this numerical inversion test.

4.2. Inverse Wave Scattering.

In this problem, the Helmholtz equation (9) is considered on the domain $D = [0,1]^2$, and the task is to learn coefficients sampled from a distribution, $a(x,y) = \sum_{k=1}^m \exp\big(-c(x-c_{1,k})^2 - c(y-c_{2,k})^2\big)$, with $c = 2\times 10^4/3$. It represents a homogeneous medium with square-shaped inclusions, randomly spread in the domain (see **SM** Figure 5). Here, $m = \mod(\overline{m})$, $m \sim \mathcal{U}([1,5])$ and $\{(c_{1,k}, c_{2,k})\} \sim \mathcal{U}([0,1]^{m\times 2})$. For each draw of the coefficient, 20 Dirichlet boundary values are prescribed, exactly as in the EIT experiment with trigonometric coefficients. The corresponding (approximate) solutions of the Helmholtz equation (9) are computed with a central finite difference scheme, and the Neumann trace is evaluated to represent the DtN map. We train the models with 10000 training samples and present the relative (median) test errors, on a test set of 4000 samples, in Table 1. Again, CNIO is the best-performing model, beating the next-best FCNN by a significant margin.

4.3. Radiative Transport Equation and Optical Imaging.

Next, we consider the radiative transport equation (11) in the domain $X \times V$, where $X = [0,1]$ and $V = [-1,1]$, with $\varepsilon = 1$. Consequently, $\Gamma_- = \{(0,v), v \in [0,1]\} \cup \{(1,v), v \in [-1,0]\}$. The task is to infer the absorption and scattering coefficients from the Albedo operator (12). To this end, we fix $k(v,v') = 1$, $\sigma_a = 1 - a$ in (11) and draw the absorption coefficient a from the distribution, $a(x) = c\chi_{[-1/2,1/2]}(rx - x_0) + 1$, with χ denoting the characteristic function and with $c \sim \mathcal{U}([0.5,1])$, $x_0 \sim \mathcal{U}([0,1])$ and $r \sim \mathcal{U}([0,0.8])$. Once the coefficient is drawn, boundary conditions on the inflow boundary Γ_- are imposed by setting $\phi_\ell(0,v) = \exp\big(-200\,(v-v_\ell)^2\big)$, and $\phi_\ell(1,v) = 0$, if $v_\ell > 0$, and $\phi_\ell(0,v) = 0$, and $\phi_\ell(1,v) = \exp\big(-200\,(v-v_\ell)^2\big)$, if $v_\ell < 0$, with v_ℓ being the ℓ-th quadrature point used to approximate the integral term in (11), $1 \leq \ell \leq 32$. Then, the radiative transport equation is approximated with a finite-element method, and the resulting solution u_ℓ is evaluated at the outflow boundary Γ_+ as the output of the Albedo operator (12). All the models are trained on 4096 training samples, and the relative median test errors on a test set of 2048 samples are presented in Table 1, demonstrating that CNIO significantly outperforms both FCNN and DeepONet, also in this case, resulting in low test errors even for the underlying discontinuous absorption coefficient.

4.4. Seismic Imaging.

In the final test, we model seismic imaging by considering the acoustic wave equation (14) in the space-time domain $[0,1]^2 \times [0,T]$ and the task at hand is to learn the underlying squared-slowness coefficients a from the *source-to-*

receivers map (15). To this end, we choose two types of coefficients from (Deng et al., 2021), the so-called *Style-A* and *CurveVel-A* datasets. For each medium, waves are generated at source locations $(x_{s_\ell,0})$ on the vertical boundary, for $\ell = 1, \ldots, 5$. The corresponding acoustic wave equation is solved with a finite difference scheme, and the temporal data is recorded at receivers on the vertical boundary. We follow (Deng et al., 2021) and train all the models with 55000 and 22000 training samples for the *Style-A* and *CurveVel-A* datasets, respectively, and present the resulting (median) relative test errors, on a test set of 7000 and 6000 samples, in Table 1. We observe from the table that even for this problem, CNIO is either outperforming or on par with FCNN. This is particularly noteworthy as the FCNN architecture was demonstrated to be one of the states of the art on this problem in (Deng et al., 2021) among several machine learning models.

5. Related Work.

Existing methods for solving the class of PDE inverse problems considered here include the so-called *Direct Methods*, such as the D-bar method (Isaacson et al., 2004) for EIT and the so-called Imaging condition (Claerbout, 1985) for seismic inversion. Iterative methods approximating fixed points (Bakushinsky & Kokurin, 2005) are also used, including the very popular gradient-based PDE-constrained optimization methods (Chavent, 2010). Another approach falls in the category of Bayesian formulation of inverse problems (Tarantola, 2005; Stuart, 2010), which also quantifies the uncertainty of the solution. Finally, directly learning some PDE inverse operators from data has been considered in (Maarten et al., 2022) and references therein.

Table 1. Relative median L^1-error and L^2-error computed over testing samples for different benchmarks and models with the best performing model highlighted in bold.

	DONet		FCNN		CNIO	
	$L^1\downarrow$	$L^2\downarrow$	$L^1\downarrow$	$L^2\downarrow$	$L^1\downarrow$	$L^2\downarrow$
Calderón Problem Trigonometric	1.57%	2.0%	1.22%	1.5%	**0.67%**	**0.86%**
Calderón Problem Heart&Lungs	0.44%	2.76%	0.18%	1.34%	**0.098%**	**1.11%**
Inverse Wave Scattering	2.65%	5.5%	1.24%	3.4%	**0.91%**	**2.82%**
Radiative transport	2.82%	5.04%	1.55%	3.72%	**1.19%**	**2.82%**
Seismic Imaging CurveVel - A	3.91%	5.79%	**2.65%**	5.05%	**2.65%**	**4.77%**
Seismic Imaging Style - A	3.42%	4.77%	3.12%	4.62%	**3.02%**	**4.42%**

6. Discussion.

For PDEs, written in the abstract form (1), we consider a large class of inverse problems that are only well-defined when the underlying inverse operator (4), maps an operator (the boundary observation operator (2)) to the underlying coefficient (a function). The resulting inverse problem amounts to inferring the unknown coefficient a from data pairs $(\Lambda_a, \mathcal{F}^{-1}(\Lambda_a))$ representing the observation operator. Existing operator learning frameworks such as DeepONets (17) and FNOs (20) only map functions to functions. Hence, one needs to adapt them to be able to learn *mappings between operators and functions* in order to solve the inverse problem (4). To this end, we have proposed a novel architecture, termed *Neural Inverse Operators* (NIO), based on a suitable composition of DeepONets and FNOs. Our architecture is motivated by the underlying structure of the inverse map and comes in fully-connected (25) or convolutional (27) variants. We tested the NIO on a variety of benchmark inverse problems. These include the Calderón Problem in electrical impedance tomography, inverse wave scattering modelled with the Helmholtz equation, optical imaging using the radiative transport equation, and seismic imaging with the acoustic wave equation. For all these problems, NIO outperformed baselines significantly and provided accurate approximations to the unknown coefficients with small errors (see **SM** D).

One major challenge in inverse problems is (poor) stability. However, our tests showed that the learned NIOs are very robust to both large noise in the observed data and varying grid sizes in PDE solvers, which is a source of modelling error (see **SM** Tables 6 and 7). Therefore, we have provided a unifying machine learning framework that can accurately solve a large class of PDE inverse problems arising in applications in science and engineering. The most attractive aspect of NIO is the extremely low computational cost at inference compared to traditional PDE-constrained optimization methods while still retaining satisfactory accuracy levels. As an example, the inference time for the inverse wave scattering with CNIO is 0.5 secs. On the other hand, obtaining similar accuracy with a PDE-constrained optimization approach will require parallelization and approximately 800 Helmholtz PDE solves of 2 secs each, *making NIO at least three orders of magnitude faster*.

As this is the first paper where an end-to-end machine learning framework is proposed for learning maps between operators and functions, various extensions are possible. For instance, other architectures, such as recently proposed LOCA (Kissas et al., 2022), VIDON (Prasthofer et al., 2022), or graph-based approaches (Boussif et al., 2022; Brandstetter et al., 2022), can be adapted in this context. Problems in higher-dimensional (particularly with seismic) imaging need to be considered to explore how NIOs scale with increasing problem size. Finally, approximation bounds and universality results, in the spirit of (Lanthaler et al., 2022; Kovachki et al., 2021a) need to be derived in order to place NIOs on a solid theoretical footing.

References

Bakushinsky, A. B. and Kokurin, M. Y. *Iterative methods for approximate solution of inverse problems*, volume 577. Springer Science & Business Media, 2005.

Bal, G. and Jollivet, A. Stability estimates in stationary inverse transport. *Inverse Problems & Imaging*, 2(4):427, 2008.

Boussif, O., Assouline, D., Benabbou, L., and Bengio, Y. MAgNet: Mesh Agnostic Neural PDE solver. *arXiv preprint arXiv:2210.05495*, 2022.

Brandstetter, J., Worrall, D. E., and Welling, M. Message Passsing Neural PDE solvers. *arXiv preprint arXiv:2202.03376*, 2022.

Caday, P., de Hoop, M. V., Katsnelson, V., and Uhlmann, G. Scattering control for the wave equation with unknown wave speed. *Archive for Rational Mechanics and Analysis*, 231(1):409–464, 2019.

Cai, S., Wang, Z., Lu, L., Zaki, T. A., and Karniadakis, G. E. DeepM&Mnet: Inferring the electroconvection multiphysics fields based on operator approximation by neural networks. *Journal of Computational Physics*, 436: 110296, 2021.

Chavent, G. *Nonlinear least squares for inverse problems: theoretical foundations and step-by-step guide for applications*. Springer Science & Business Media, 2010.

Claerbout, J. F. *Imaging the earth's interior*, volume 1. Blackwell scientific publications Oxford, 1985.

Clop, A., Faraco, D., and Ruiz, A. Stability of Calderón's inverse conductivity problem in the plane for discontinuous conductivities. *Inverse Problems & Imaging*, 4(1):49, 2010.

Deng, C., Feng, S., Wang, H., Zhang, X., Jin, P., Feng, Y., Zeng, Q., Chen, Y., and Lin, Y. OpenFWI: Large-Scale Multi-Structural Benchmark Datasets for Seismic Full Waveform Inversion. *arXiv preprint arXiv:2111.02926*, 2021.

Engquist, B. and Zhao, H. Approximate separability of the Green's function of the Helmholtz equation in the high-frequency limit. *Comm. Pure Appl. Math.*, 71(11): 2220–2274, 2018.

Evans, L. C. *Partial differential equations*, volume 19. American Mathematical Soc., 2010.

Goodfellow, I., Bengio, Y., and Courville, A. *Deep learning*. MIT press, 2016.

Isaacson, D., Mueller, J. L., Newell, J. C., and Siltanen, S. Reconstructions of chest phantoms by the D-bar method for electrical impedance tomography. *IEEE Transactions on medical imaging*, 23(7):821–828, 2004.

Isakov, V. *Inverse Problems for Partial Differential Equations*. Springer, 2017.

Kissas, G., Seidman, J. H., Guilhoto, L. F., Preciado, V. M., Pappas, G. J., and Perdikaris, P. Learning operators with coupled attention. *Journal of Machine Learning Research*, 23(215):1–63, 2022.

Kovachki, N., Lanthaler, S., and Mishra, S. On universal approximation and error bounds for Fourier Neural Operators. *Journal of Machine Learning Research*, 22:Art–No, 2021a.

Kovachki, N., Li, Z., Liu, B., Azizzadensheli, K., Bhattacharya, K., Stuart, A., and Anandkumar, A. Neural operator: Learning maps between function spaces. *arXiv preprint arXiv:2108.08481v3*, 2021b.

Lai, R.-Y., Li, Q., and Uhlmann, G. Inverse problems for the stationary transport equation in the diffusion scaling. *SIAM Journal on Applied Mathematics*, 79(6):2340–2358, 2019.

Lanthaler, S., Mishra, S., and Karniadakis, G. E. Error estimates for DeepONets: A deep learning framework in infinite dimensions. *Transactions of Mathematics and Its Applications*, 6(1):tnac001, 2022.

Li, Z., Kovachki, N. B., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A., and Anandkumar, A. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021a.

Li, Z., Zheng, H., Kovachki, N., Jin, D., Chen, H., Liu, B., Azizzadenesheli, K., and Anandkumar, A. Physics-informed neural operator for learning partial differential equations. *arXiv preprint arXiv:2111.03794*, 2021b.

Liu, S. and Oksanen, L. A Lipschitz stable reconstruction formula for the inverse problem for the wave equation. *Transactions of the American Mathematical Society*, 368 (1):319–335, 2016.

Lu, L., Jin, P., Pang, G., Zhang, Z., and Karniadakis, G. E. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3):218–229, 2021.

Maarten, V., Lassas, M., and Wong, C. A. Deep learning architectures for nonlinear operator functions and nonlinear inverse problems. *Mathematical Statistics and Learning*, 4(1):1–86, 2022.

Mao, Z., Lu, L., Marxen, O., Zaki, T., and Karniadakis, G. E. DeepMandMnet for hypersonics: Predicting the coupled flow and finite-rate chemistry behind a normal shock using neural-network approximation of operators. Preprint, available from arXiv:2011.03349v1, 2020.

Muller, J. and Siltanen, S. *Linear and nonlinear inverse problems with practical applications*. SIAM, 2012.

Nachman, A. Reconstructions from boundary measurements. *Ann. Math.*, 128(3):531–576, 1988.

Pathak, J., Subramanian, S., Harrington, P., Raja, S., Chattopadhyay, A., Mardani, M., Kurth, T., Hall, D., Li, Z., Azizzadenesheli, K., Hassanzadeh, p., Kashinath, K., and Anandkumar, A. Fourcastnet: A global data-driven high-resolution weather model using adaptive Fourier neural operators. *arXiv preprint arXiv:2202.11214*, 2022.

Prasthofer, M., De Ryck, T., and Mishra, S. Variable input deep operator networks. *arXiv preprint arXiv:2205.11404*, 2022.

Stefanov, P., Uhlmann, G., and Vasy, A. On the stable recovery of a metric from the hyperbolic DN map with incomplete data. *Inverse Problems & Imaging*, 10(4): 1141, 2016.

Stuart, A. M. Inverse problems: a Bayesian perspective. *Acta numerica*, 19:451–559, 2010.

Symes, W. W. The seismic reflection inverse problem. *Inverse problems*, 25(12):123008, 2009.

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

Uhlmann, G. Electrical impedance tomography and Calderón's problem. *Inverse problems*, 25(12):123011, 2009.

Yilmaz, O. *Seismic Data Analysis*. Society for exploration geophysicists, 2011.

Supplementary Material for:
Neural Inverse Operators for solving PDE Inverse problems.

A. Depiction of PDE Inverse Problems.

In the following figures, we illustrate the different PDE inverse problems considered in the main text.

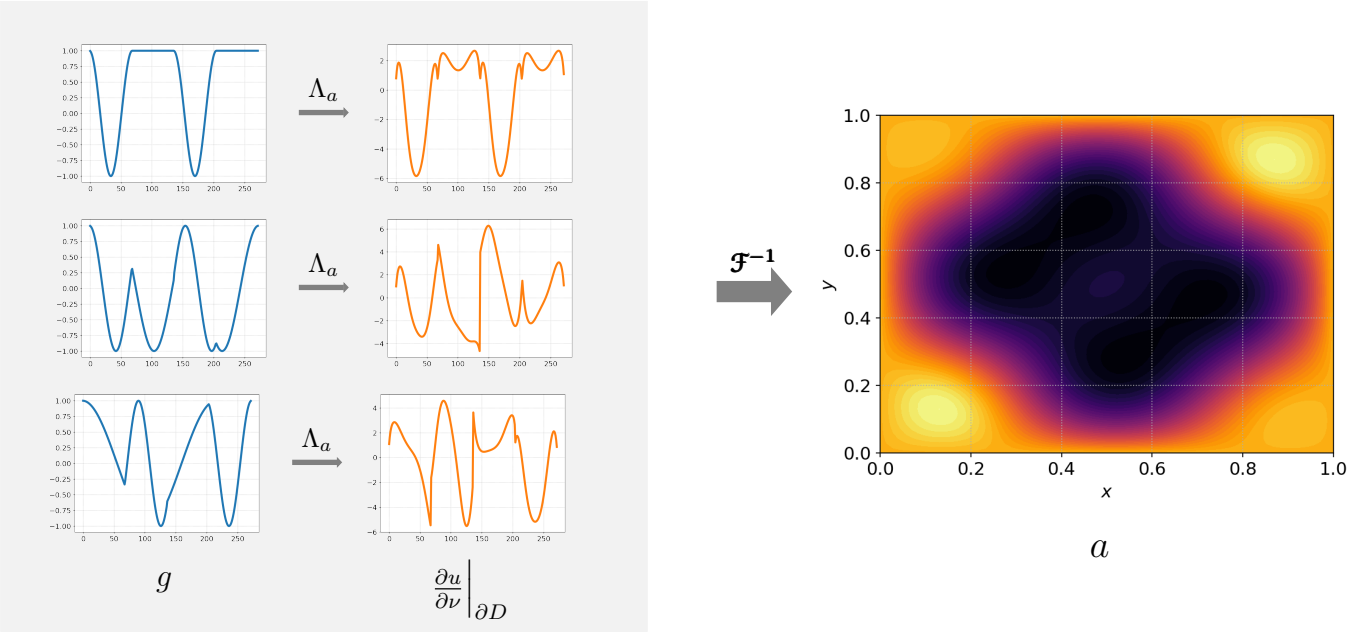


Figure 3. Illustration of a typical input (left) and output (right) sample for the Calderón Problem for EIT with trigonometric coefficients. The input is the Dirichlet-to-Neumann (DtN) map (7), represented here by three Dirichlet Boundary condition (Voltage) to Current pairs and the output is the conductivity coefficient a.

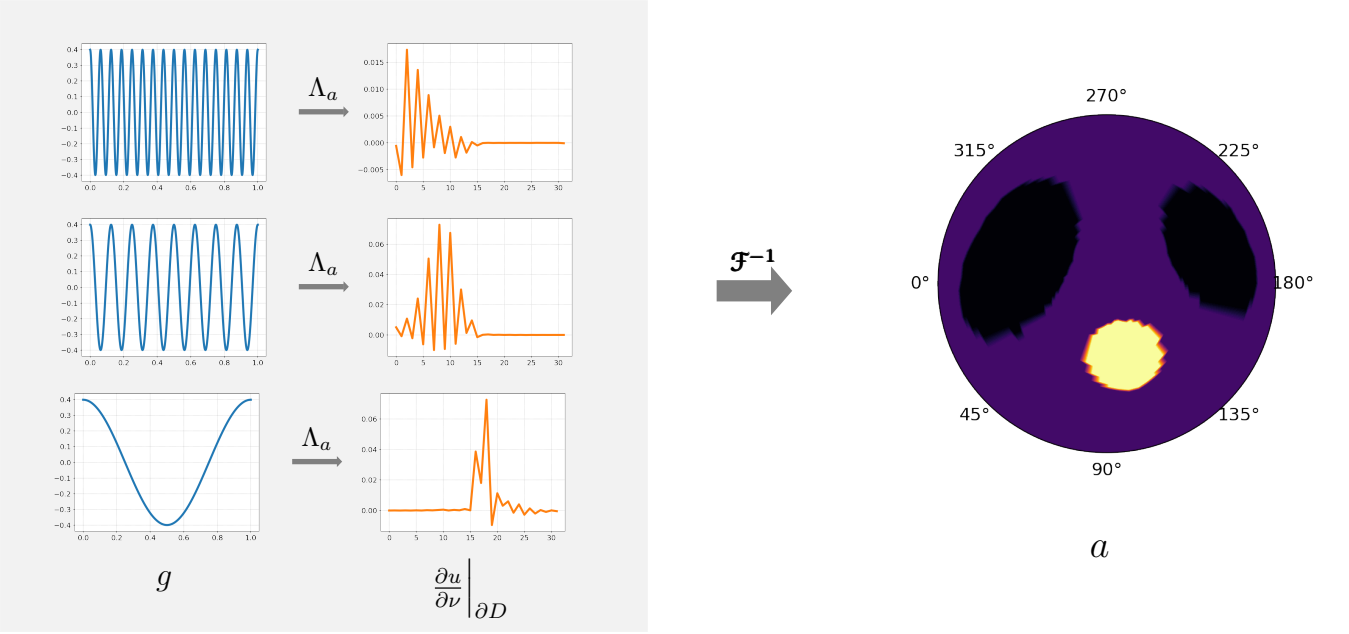


Figure 4. Illustration of EIT for the discontinuous heart-lung Phantom of (Muller & Siltanen, 2012). Left: Input through the DtN (voltage-to-current) map. Right: Conductivity field showing the phantom of heart and lungs.

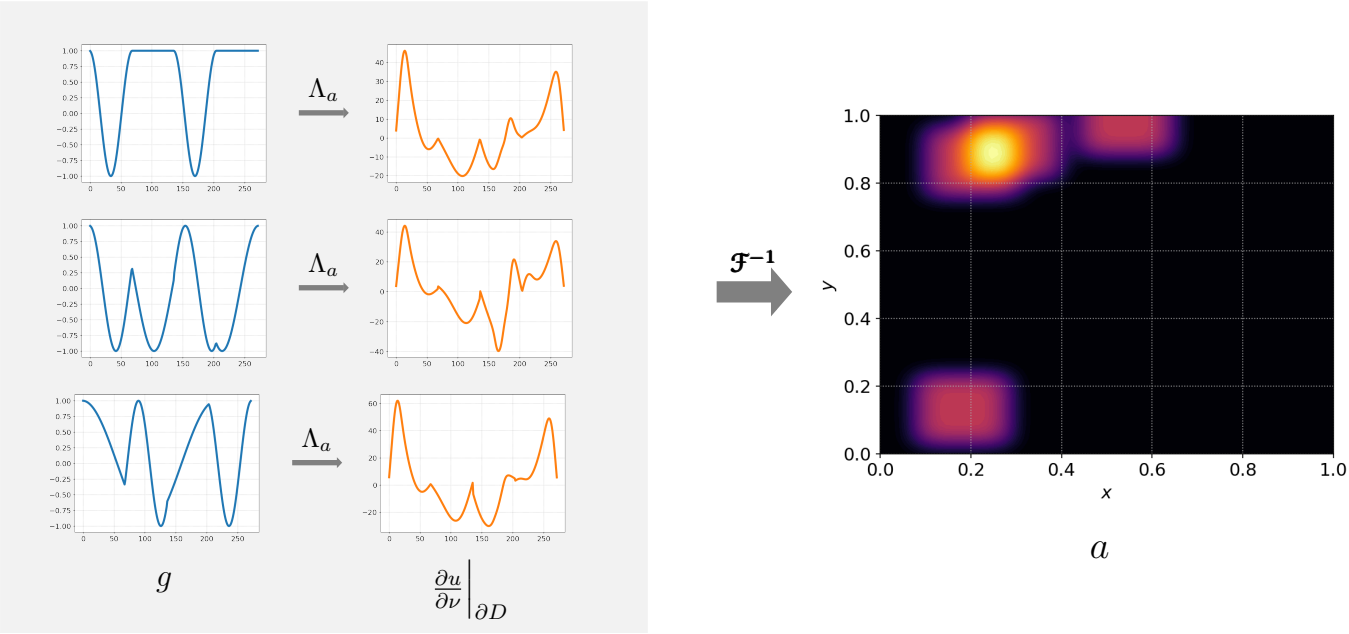


Figure 5. Illustration of detection of inclusions through the Inverse Wave Scattering with the Helmholtz equation. Left: Input represented through 3 samples for the DtN map. Right: Coefficient a.

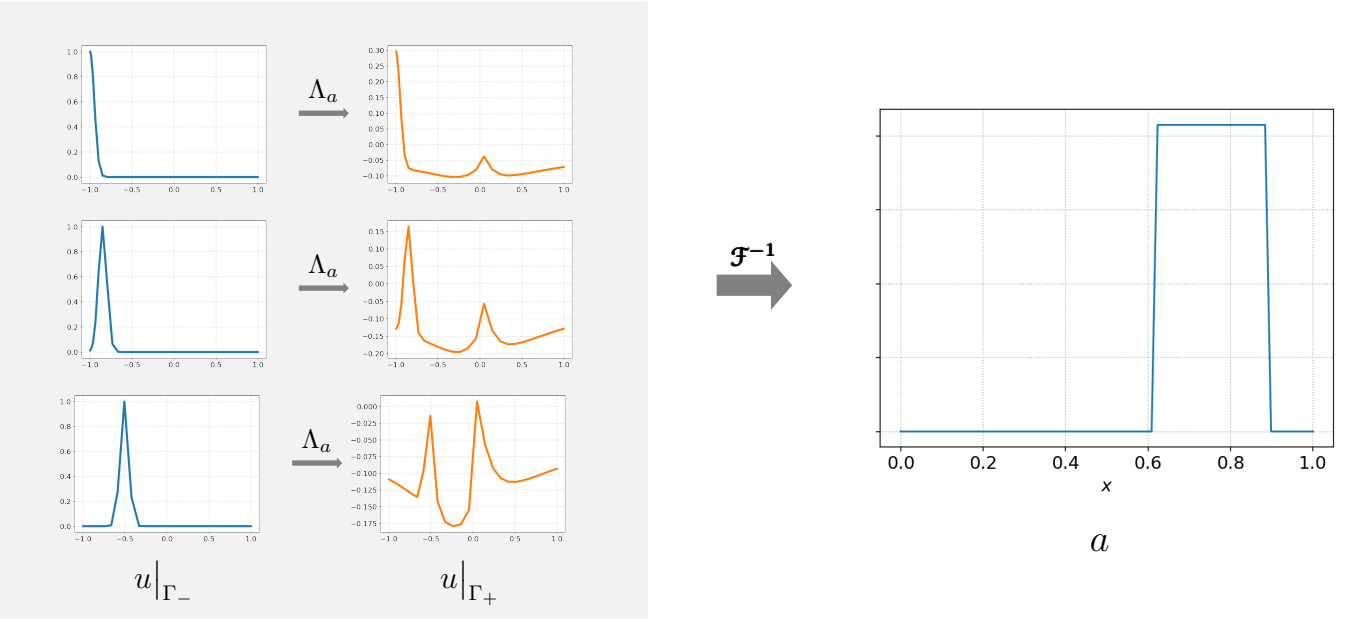


Figure 6. Illustration of Optimal Imaging through the Radiative Transport Equation. Left: Input is the Albedo operator (12) illustrated with 3 mappings between the inflow and outflow boundaries and Right: Output is the Scattering coefficient.

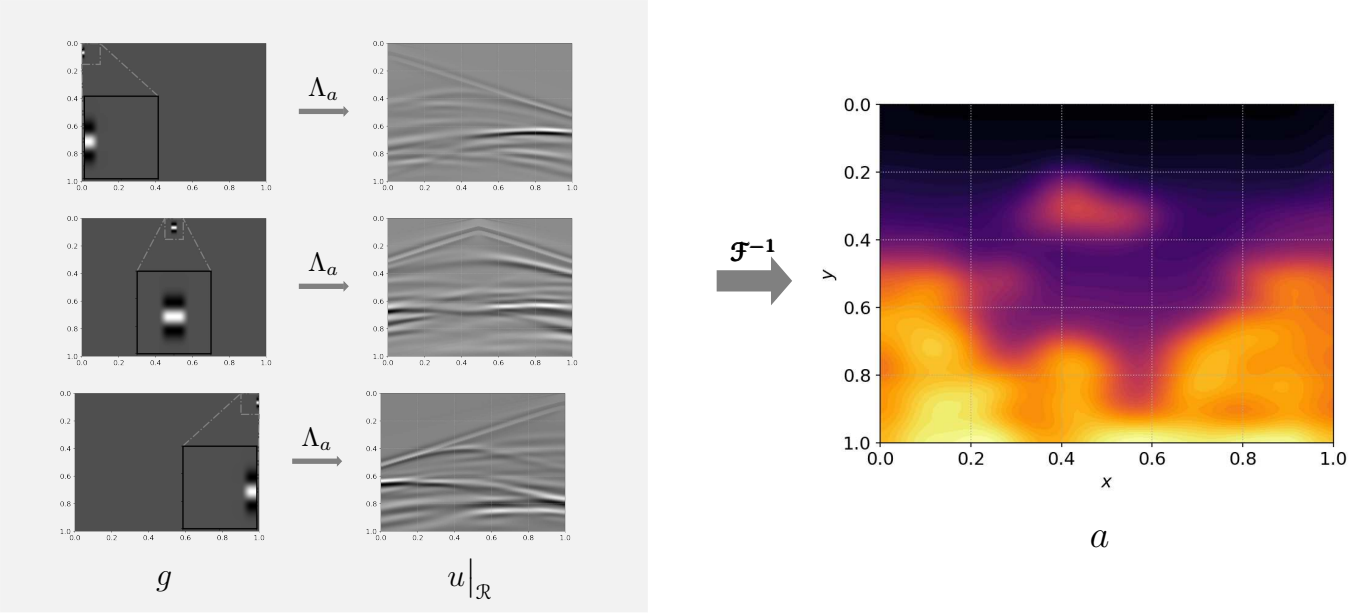


Figure 7. Illustration of Seismic Imaging. Left: Input is Source-to-Receiver map (15) between Incident waves generated at Sources to Temporal signals recorder at Receivers. Right: Output is the velocity coefficient, corresponding to *Style A* dataset of (Deng et al., 2021).

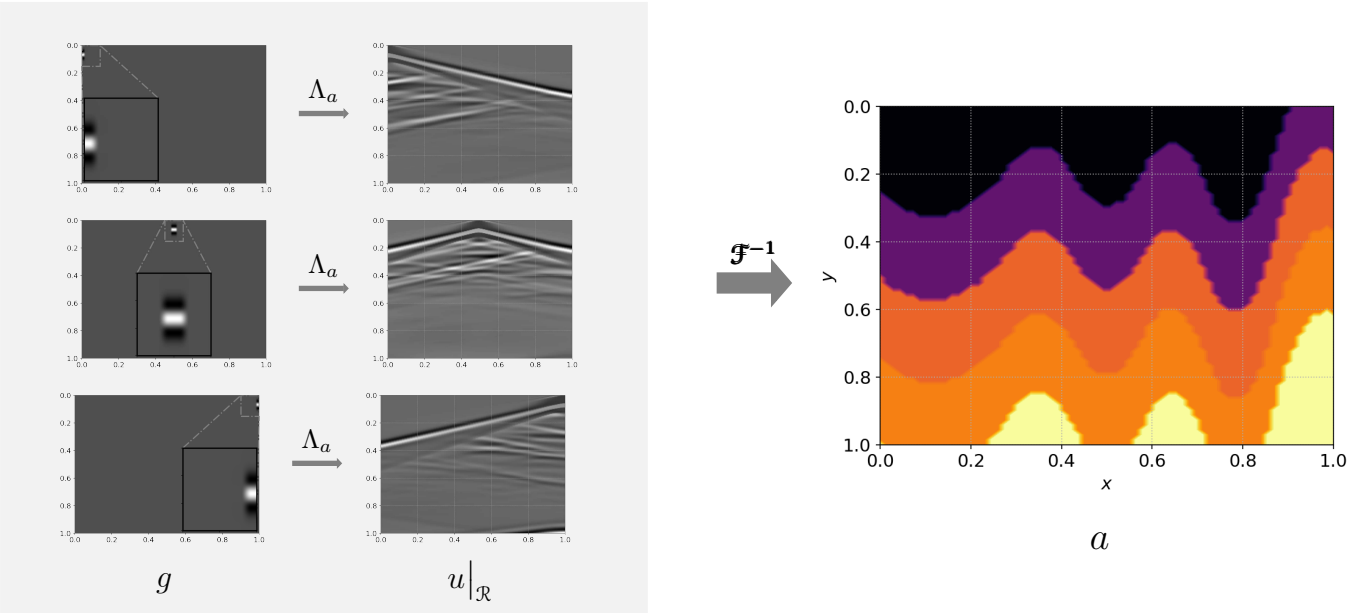


Figure 8. Illustration of Seismic Imaging. Left: Input is Source-to-Receiver map (15) between Incident waves generated at Sources to Temporal signals recorder at Receivers. Right: Output is the velocity coefficient, corresponding to *CurveVel A* dataset of (Deng et al., 2021)

B. Proof of Formula (23) in Main Text.

Below, we prove the representation formula (23) in the main text.

Proof. Multiplying u_k (the solution of (22)) to Eqn (21) and integrating over space, we obtain,

$$\int_D u_k \Delta \varphi_k dz + \lambda_k \int_D \varphi_k dz = 0$$

Integrating by parts in the above equation and using the Gauss-Green formula yields,

$$-\int_D \langle \nabla u_k, \nabla \varphi_k \rangle dz + \int_{\partial D} u_k \underbrace{\frac{\partial \varphi_k}{\partial \nu}}_{=0} ds(z) + \lambda_k \int_D u_k \varphi_k dz = 0. \tag{29}$$

Note that the Neumann boundary conditions from (21) in the above.

Similarly, multiplying the solution φ_k of the Neumann problem (21) to the Eqn (22) and repeating the above integration parts yields,

$$-\int_D \langle \nabla u_k, \nabla \varphi_k \rangle dz + \int_{\partial D} g_k \frac{\partial u_k}{\partial \nu} ds(z) + \int_D a(z) u_k \varphi_k dz = 0. \tag{30}$$

Formula (23) follows by subtracting (29) from (30). □

C. Architecture and Training Details

Below, details concerning the model architectures and training are discussed.

C.1. Architecture Details

C.1.1. Feed Forward Dense Neural Networks

Given an input $y \in \mathbb{R}^m$, a feed-forward neural network (also termed as a multi-layer perceptron) transforms it to an output, through a layer of units (neurons) which compose of either affine-linear maps between units (in successive layers) or scalar non-linear activation functions within units (Goodfellow et al., 2016), resulting in the representation,

$$u_\theta(y) = C_{L_t} \circ \sigma \circ C_{L_t - 1} \ldots \circ \sigma \circ C_2 \circ \sigma \circ C_1(y). \tag{31}$$

Here, $\circ$ refers to the composition of functions, and σ is a scalar (non-linear) activation function. For any $1 \leq \ell \leq L_t$, we define

$$C_\ell z_\ell = W_\ell z_\ell + b_\ell, \text{ for } W_\ell \in \mathbb{R}^{d_{\ell+1} \times d_\ell}, z_\ell \in \mathbb{R}^{d_\ell}, b_\ell \in \mathbb{R}^{d_{\ell+1}}., \tag{32}$$

and denote,

$$\theta = \{W_\ell, b_\ell\}_{\ell=1}^{L_t}, \tag{33}$$

to be the concatenated set of (tunable) weights for the network. Thus, in the machine learning terminology, a feed-forward neural network (31) consists of an input layer, an output layer, and L_t hidden layers with d_ℓ neurons, $1 < \ell < L_t$. In all numerical experiments, the trunk net of DeepONet is a feed-forward neural network. Moreover, we consider a uniform number of neurons across all the layers of the network $d_\ell = d_{\ell-1} = d$, $1 < \ell < L_t$.

C.1.2. Fully Convolutional Neural Network

Fully convolutional neural networks are a special class of convolutional networks which are independent of the input resolution. We use them as a strong baseline for PDE inverse problems in the results presented in Table 1. The networks consist of an *encoder* and *decoder*, both defined by a composition of linear and non-linear transformations:

$$\begin{aligned} E_{\theta_e}(y) &= C_L^e \circ \sigma \circ C_{L-1}^e \ldots \circ \sigma \circ C_2^e \circ \sigma \circ C_1^e(y), \\ D_{\theta_d}(z) &= C_L^d \circ \sigma \circ C_{L-1}^d \ldots \circ \sigma \circ C_2^d \circ \sigma \circ C_1^d(z), \\ u_\theta(y) &= D_{\theta_d} \circ E_{\theta_e}(y). \end{aligned} \tag{34}$$

The affine transformation C_ℓ commonly corresponds to a *convolution* operation in the encoder, and *transposed convolution* (also known as *deconvolution*), in the decoder.

The (de)convolution is performed with a kernel $W_\ell \in \mathbb{R}^{k_\ell \times k_\ell}$, stride s, and padding p. It takes as input a tensor $z_\ell \in \mathbb{R}^{w_\ell \times h_\ell \times c_\ell}$ with c_ℓ being the number of input channels, and computes $z_{\ell+1} \in \mathbb{R}^{w_{\ell+1} \times h_{\ell+1} \times c_{\ell+1}}$. Therefore, a (de)convolutional affine transformation can be uniquely identified with the tuple $(k_\ell, s, p, c_\ell, c_{\ell+1})$.

A visual representation of the convolutional architectures used for the benchmark problems is depicted in Figures 9, 10, 11. The number of channels c is selected with cross-validation. The architecture used for seismic imagining is referred to as InversionNet in (Deng et al., 2021).

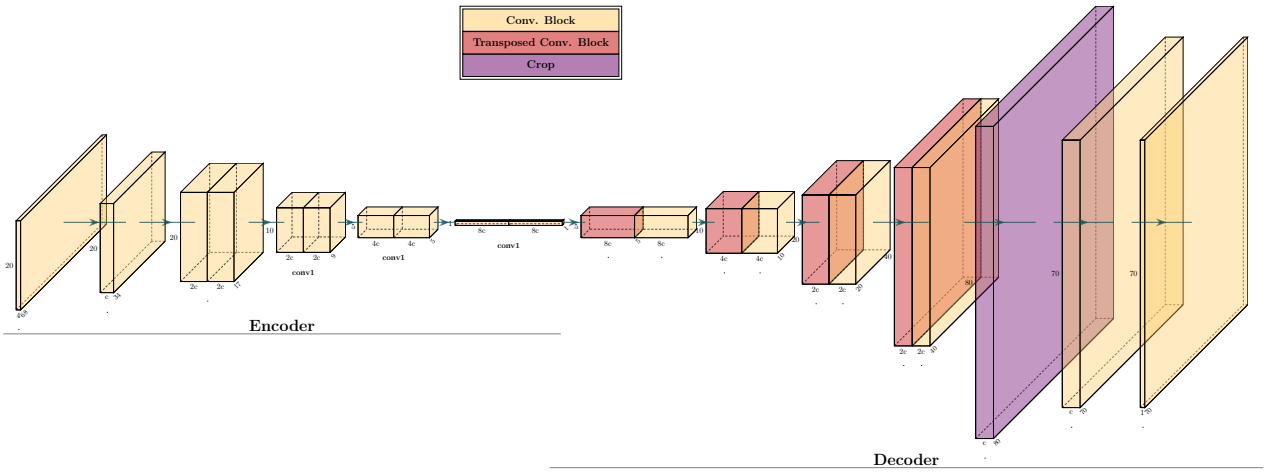


Figure 9. Schematic representation of the Fully-Convolutional Neural Network architecture used for the Calderón Problem and Inverse Wave Scattering with the Helmholtz Equation.

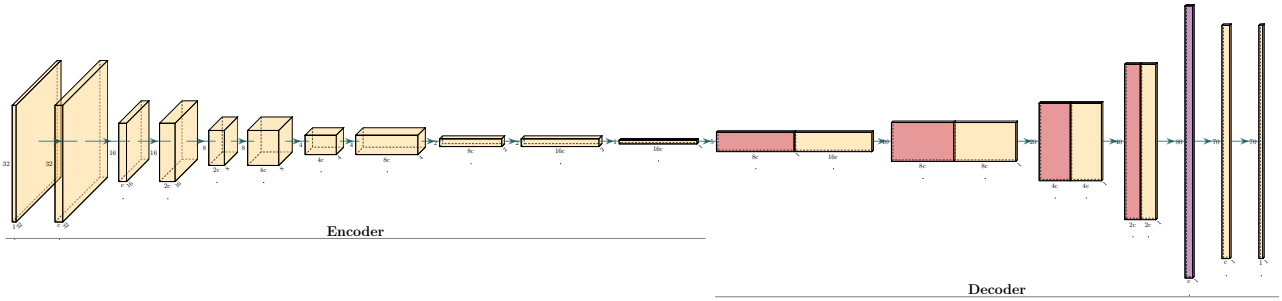


Figure 10. Schematic representation of the Fully-Convolutional Neural Network architecture used for the optical Imaging with Radiative transport Equation.

C.1.3. DEEPONET

The architectures of the branch and trunk are chosen according to the benchmark addressed. In particular, we employ standard feed-forward neural networks as trunk-net in all the experiments. In contrast, the branch is obtained as a composition of the encoder of the fully convolutional networks depicted in figures 9, 10, 11 and a linear transformation from $\mathbb{R}^n$ to $\mathbb{R}^p$, where n denotes the number of channels in the last layer of the encoder and p the number of basis functions. Moreover, $c = 64$ for the Calderón, scattering wave, and radiative transport problems.

Therefore, the architecture of the branch is fixed. The number of the trunk hidden layers L_t, units d, and p are chosen through cross-validation. On the other hand, the activation function σ is chosen to be a leaky ReLU.

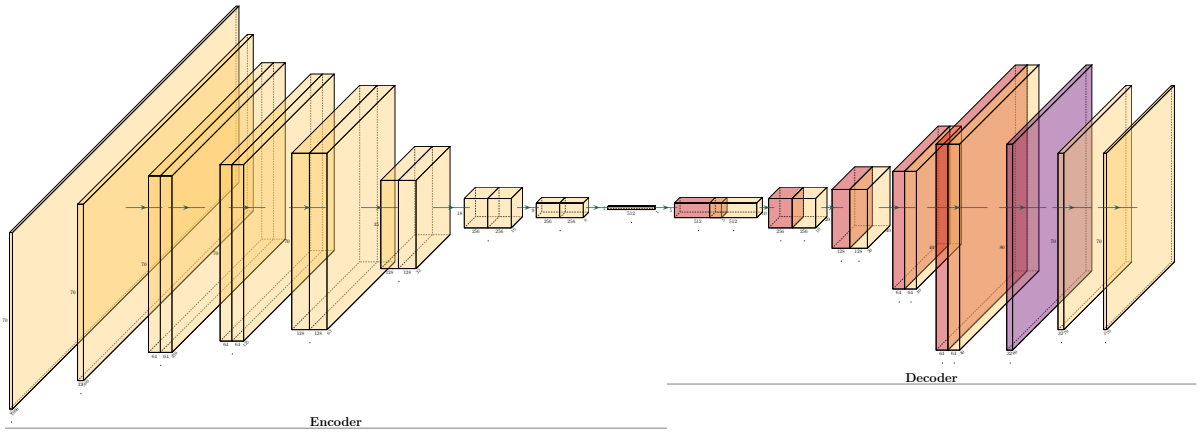


Figure 11. Schematic representation of the Fully-Convolutional Neural Network architecture used for the Seismic imaging problems.

C.1.4. FOURIER NEURAL OPERATOR

We use the implementation of the FNO model provided by the authors of (Li et al., 2021a). Specifically, the lifting R is defined by a linear transformation, and the projection Q to the target space is performed by a shallow neural network with a single hidden layer with 128 neurons and $GeLU$ activation function. The same activation function is also used for all the Fourier layers. Moreover, $b_\ell(x) = 0$, for all $\ell = 1, \dots, T$ and the weight matrix W_ℓ used in the residual connection derives from a convolutional layer defined by $(k_\ell = 1, s = 1, p = 0, c_\ell = d_v, c_{\ell+1} = d_v)$, for all $1 < \ell < T$.

C.1.5. CONVOLUTIONAL NEURAL INVERSE OPERATOR

In all numerical experiments, the proposed architecture is constructed as a composition of DeepONet and Fourier Neural Operator, both defined above. Hence, the model includes the following hyperparameters: the number of layers L_t and neurons d of the DeepONet trunk, the number of basis functions p, and the lifting dimension d_v, the number of Fourier layers T and number of (truncated)-Fourier coefficients k, of FNO.

C.2. Training Details

The training of the models, including the baselines, is performed with the ADAM optimizer, with a learning rate η for 1000 epochs (120 epochs in the Seismic imaging problem) and minimizing the L^1-loss function. We also use a step learning rate scheduler and reduce the learning rate of each parameter group by a factor γ every epoch. We train the models in mini-batches of size 256, and a weight decay of magnitude w is used. Moreover, the input and output data are transformed with a suitable map before training. Observe that the testing error reported in Table 1 has been obtained on the non-transformed output data. We consider 3 different data transformations:

1. scaling of the inputs and outputs between -1 and 1 (*Scaling*);
2. normalizing the input by subtracting the function mean, dividing by the standard deviation function, and scaling the outputs between -1 and 1 (*Inputs-Normalization*);
3. normalizing both the inputs and the outputs (*Normalization*);

All the parameters mentioned above, including the type of data transformation (*Identity*, *Scaling*, *Inputs-Normalization*, *Normalization*), are chosen through cross-validation.

At every epoch, the relative L^1 error is computed on the validation set, and the set of trainable parameters resulting in the

lowest error during the entire process is saved for testing. Early stopping is used to interrupt the training if the best validation error does not improve after 50 epochs.

The cross-validation is performed by running a random search over a chosen range of hyperparameters values and selecting the configuration, realizing the lowest relative L^1 error on the validation set. For instance, the model size (minimum and maximum number of trainable parameters) covered in this search are reported in Table 2.

The results of the random search, i.e., the best-performing hyperparameter configurations for each model and each benchmark, are reported in tables 4, 3 and 5. The FCNN hyperparameters reported in the table for the seismic imaging problem are those used in (Deng et al., 2021). For instance, differently from the input normalization used here, in (Deng et al., 2021), a log-transform of the inputs and scaling of the outputs is employed instead.

	Calderón Problem Trigonometric	**Calderón Problem Heart&Lungs**	**Inverse Wave Scattering**	**Radiative Transport**	**Seismic Imaging CurveVel - A**	**Seismic Imaging Style - A**
DONet	4,603,918 9,073,268	9,542,414 14,011,764	4,603,918 9,073,268	9,542,314 14,011,264	11,988,494 16,457,844	11,988,494 16,457,844
FCNN	1,070,339 68,315,139	4,308,067 275,366,659	1,070,339 68,315,139	2,475,107 39,526,787	24,409,123 24,409,123	24,409,123 24,409,123
CNIO	5,638,487 45,878,313	10,576,983 50,816,809	5,638,487 45,878,313	9,578,483 12,225,689	11,988,494 16,457,844	11,988,494 16,457,844

Table 2. Minimum (Top sub-row) and maximum (Bottom sub-row) number of trainable parameters among the random-search hyperparameters configurations for all the models in every problem reported in Table 1 in main text.

	Calderón Problem Trigonometric	**Calderón Problem Heart&Lungs**	**Inverse Wave Scattering**	**Radiative Transport**	**Seismic Imaging CurveVel - A**	**Seismic Imaging Style - A**
η	0.001	0.001	0.001	0.001	0.001	0.001
γ	1	0.98	1	1	0.98	0.98
w	0	0	0	0 0	1e-6	1e-6
Data Trans	Norm	Norm-Inp	Norm	Identity	Inputs-Norm	Norm
p	100	100	100	25	400	400
L_t	15	15	12	12	8	8
d	500	500	200	200	500	200
Trainable Params	8,160,668	13,099,164	5,055,368	9,940,114	12,491,744	14,088,944

Table 3. DeepONet best-performing hyperparameters configuration for different benchmark problems.

	Calderón Problem Trigonometric	**Calderón Problem Heart&Lungs**	**Inverse Wave Scattering**	**Radiative Transport**	**Seismic Imaging CurveVel - A**	**Seismic Imaging Style - A**
η	0.0005	0.0005	0.0005	0.0005	0.0001	0.0001
γ	1	1	1	1	1	1
w	0	0	0	1e-6	1e-4	1e-4
Data Trans	Scaling	Inputs-Norm	Scaling	Scaling	Log-Scaling	Log-Scaling
c	128	128	128	16	64	64
Trainable Params	68,315,139	275,366,659	68,315,139	2,475,107	24,409,123	24,409,123

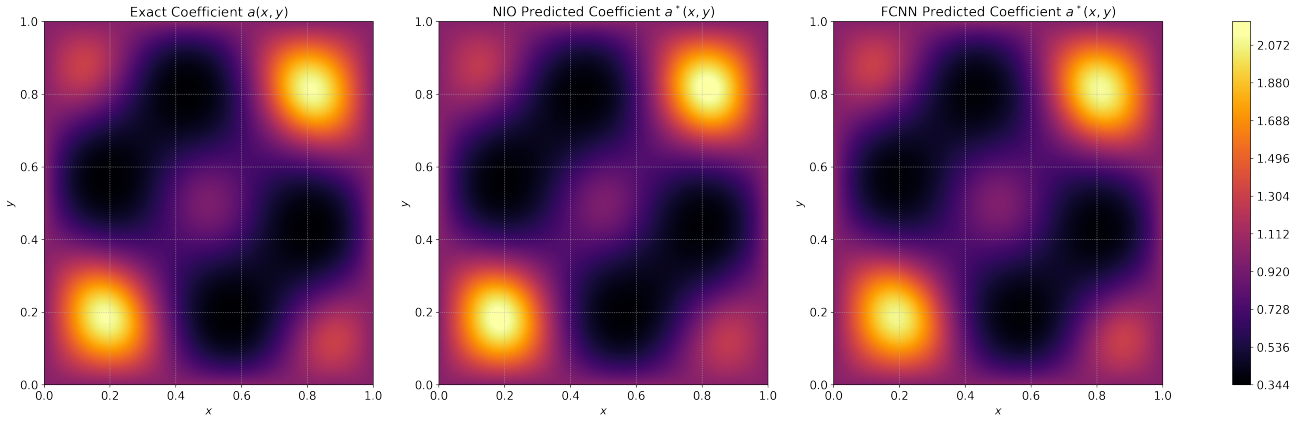
Table 4. Fully convolutional neural network best-performing hyperparameters configuration for different benchmark problems.

	Calderón Problem Trigonometric	Calderón Problem Heart&Lungs	Inverse Wave Scattering	Radiative Transport	Seismic Imaging CurveVel - A	Seismic Imaging Style - A
η	0.001	0.001	0.001	0.001	0.001	0.001
γ	0.98	1	0.98	0.98	1	0.97
w	0	0	1e-6	0	0	0
Data Trans	Scaling	Norm	Scaling	Scaling	Inputs-Norm	Norm
p	25	25	100	400	500	500
L_t	8	8	6	6	8	8
d	200	200	200	100	256	256
k	25	50	25	32	16	16
d_v	64	64	64	128	64	32
L	4	3	3	4	3	3
Trainable Params	25,344,763	40,519,099	20,192,953	11,952,789	14,088,944	12,491,744

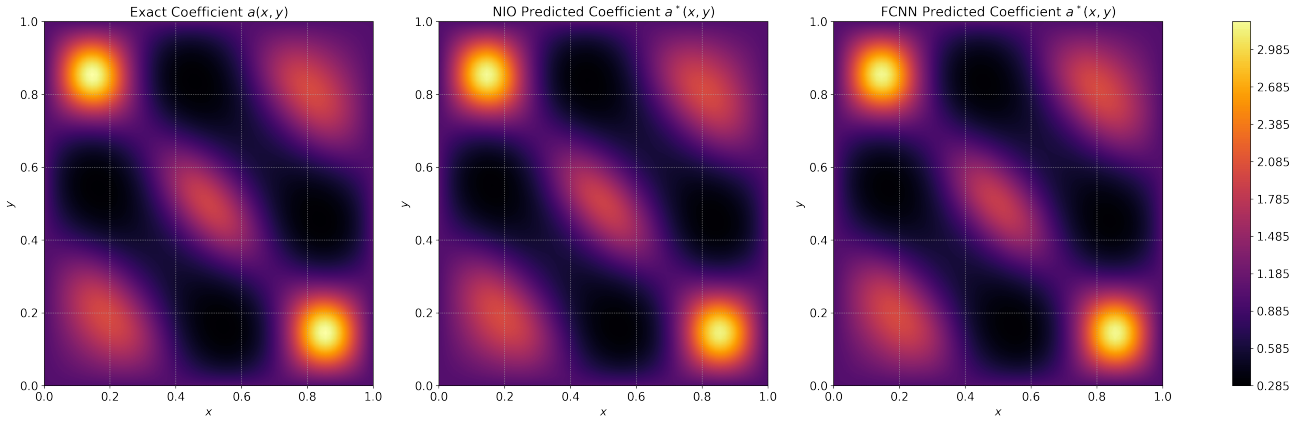
Table 5. Convolutional Neural Inverse Operator best-performing hyperparameters configuration for different benchmark problems.

D. Further Experimental Results

D.1. Illustration of Results reported in Table 1.

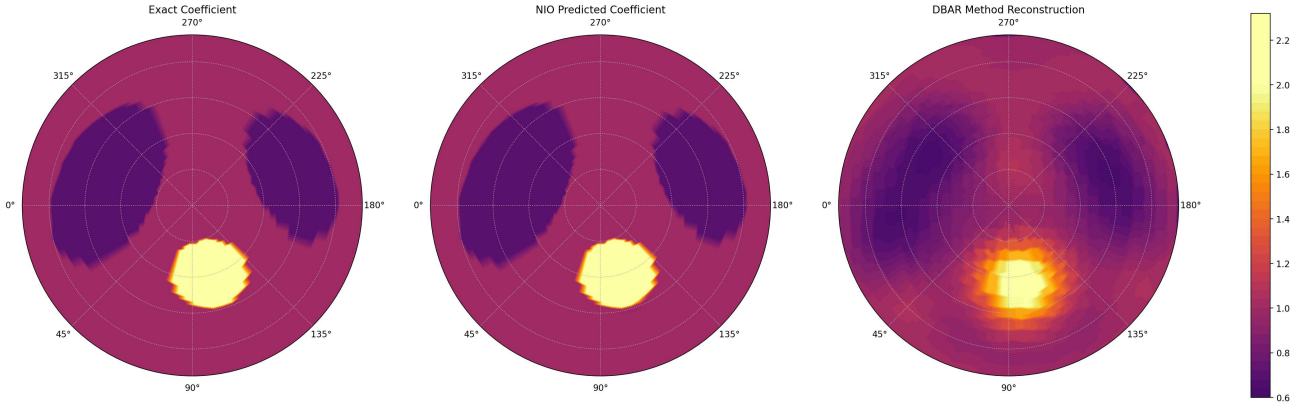


(a) Test Sample 1

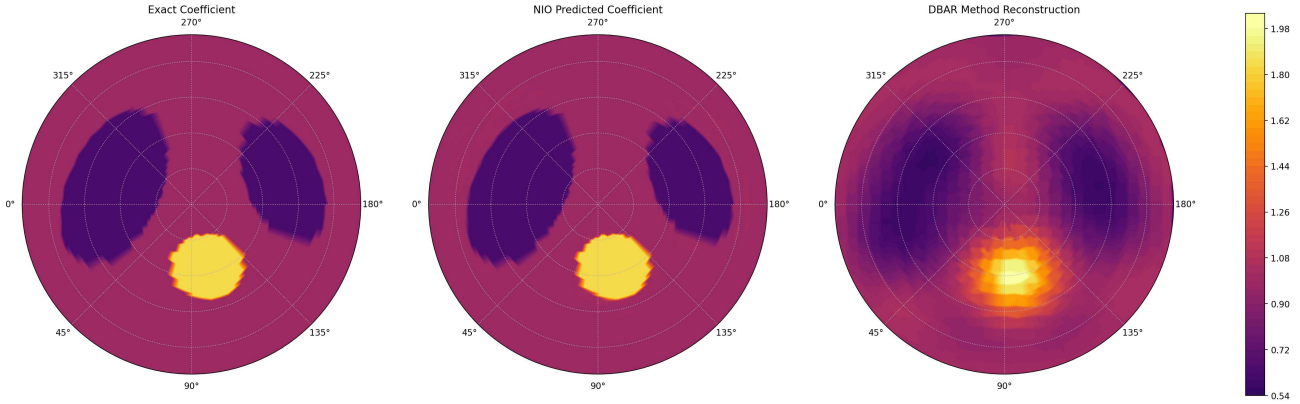


(b) Test Sample 2

Figure 12. Exact and predicted coefficients for two different test samples (Rows) for the Calderón problem with Trigononmetric - coefficients. Left Column: Ground Truth. Middle Column: CNIO reconstruction. Right Column: FCNN Reconstruction.



(a) Test Sample 1



(b) Test Sample 2

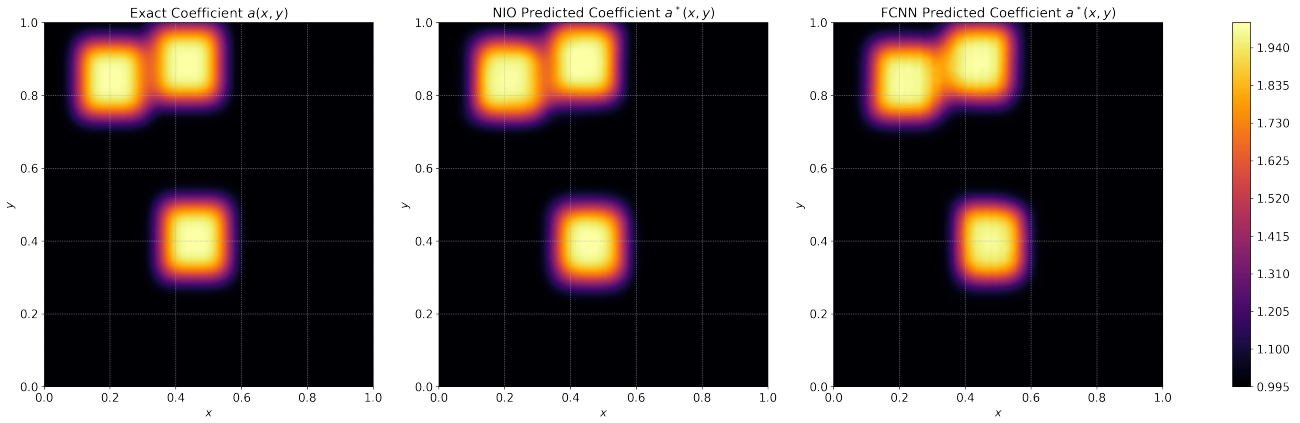
Figure 13. Exact and predicted coefficients for two different test samples (Rows) for the Calderón problem with Trigononmetric - coefficients. Left Column: Ground Truth. Middle Column: CNIO reconstruction. Right Column: Reconstruction with the D-bar Direct method of (Muller & Siltanen, 2012)

We start by elaborating on the results obtained on the benchmarks and presented in Table 1. In Figure 12, we show two randomly drawn test samples for the Calderón Problem for inferring conductivity with trigonometric coefficients by EIT. For both these test samples, we see that CNIO (and FCNN) can accurately approximate the ground truth without any visible artifacts. This observation correlates with very small test errors with CNIO. At least for these two samples, there appears to be little visible difference between CNIO and FCNN. Nevertheless, the results from Table 1 demonstrate that CNIO outperforms FCNN considerably on this problem by almost halving the test error.

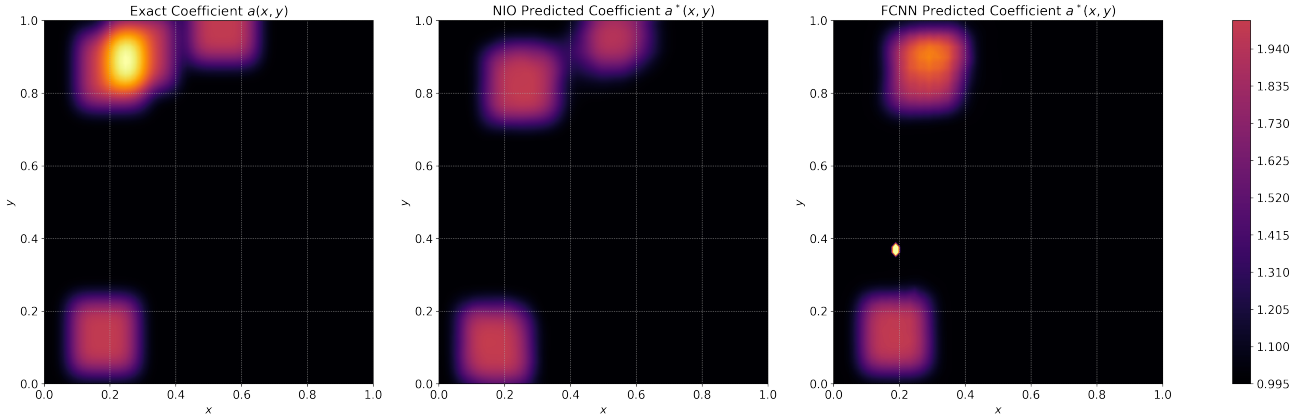
Next, in Figure 13, we focus on the discontinuous heart-lungs Phantom inferred with EIT. We follow the problem setup of (Muller & Siltanen, 2012) and plot the ground truth and the CNIO reconstruction for a couple of randomly chosen test samples. For comparison, we also plot the reconstruction with the D-bar method. This method is a direct method (Muller & Siltanen, 2012), which is widely used in the context of EIT. As observed from the figure, CNIO reconstructs the ground truth to very high accuracy, consistent with the very small errors presented in Table 1.

On the other hand, the D-bar method is relatively inaccurate and provides a blurred and diffusive reconstruction of the shapes. In fact, the L^1-test error for the D-bar method is an unacceptably high 8.75%, compared to the almost 0.1% test error with CNIO. This is even more impressive when one looks at the run times. The D-bar method takes approximately 2 hours to run for a single sample, whereas the inference time for CNIO is only 0.5 seconds. Thus, we can provide a method which *two orders of magnitude more accurate while being four orders of magnitude faster to run*. This highlights the massive gain in performance with machine learning-based methods, such as CNIO, compared to traditional direct methods.

In Figure 14, we plot the results of two randomly chosen test samples for the Inverse wave scattering problem and compare the ground truth with the reconstruction with CNIO and FCNN. In the first sample (top row), both models accurately reconstruct the ground truth coefficient with very little visible difference between the competing models. In contrast, in the



(a) Test Sample 1



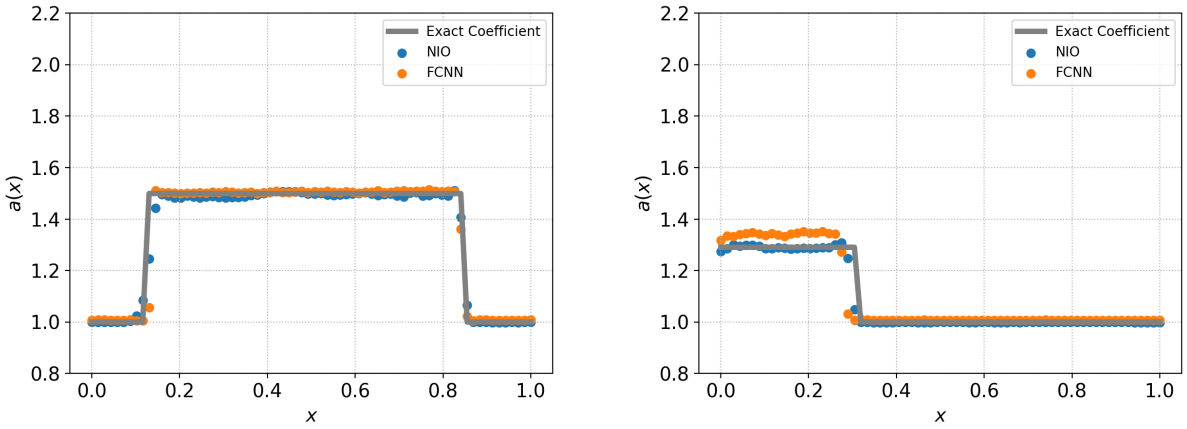
(b) Test Sample 2

Figure 14. Exact and predicted coefficients for two different test samples (Rows) for the Inverse Wave Scattering with Helmholtz Equation. Left Column: Ground Truth. Middle Column: CNIO reconstruction. Right Column: FCNN Reconstruction.

second sample, the reconstruction with CNIO and FCNN are noticeable differences. In particular, FCNN cannot reconstruct the small rectangular scatterer (at the top right of the square domain), whereas CNIO can reconstruct it. This possibly explains why CNIO is significantly more accurate (see Table 1) for this experiment in reconstructing scatterers.

In Figure 15, we plot two randomly chosen test samples to recover the absorption coefficient with optical imaging for the Radiative transport equation (11). The ground truth and reconstructions obtained with CNIO and FCNN are shown. For the first test sample, both models can provide an accurate reconstruction with a sharp resolution of the discontinuities in the absorption coefficient. On the other hand, for the second sample (Figure 15 Right), we see that FCNN gets the correct location but the wrong magnitude of the discontinuity, whereas CNIO can approximate both accurately, probably accounting for the significant gain in accuracy on this problem (see Table 1 of main text).

In Figures 16 and 17, we show two randomly chosen test samples for Seismic imaging of the subsurface property (squared slowness) by the acoustic wave equation (14), corresponding to the *CurveVel-A* and *Style A* datasets (considered in (Deng et al., 2021)), respectively. Both figures show that CNIO and FCNN reconstruct the coefficient reasonably accurately, although there are slight differences between the models. Nevertheless, coupled with quantitative results from Table 1, we can conclude that CNIO is at least on par with FCNN, which was shown to be one of the state-of-the-art models in this context in (Deng et al., 2021).



(a) Test Sample 1 (b) Test Sample 2

Figure 15. Exact and predicted absorption coefficients for two different test samples, obtained with optical imaging for the radiative transport Equation.

D.2. Sensitivity of Reconstructions to Noise

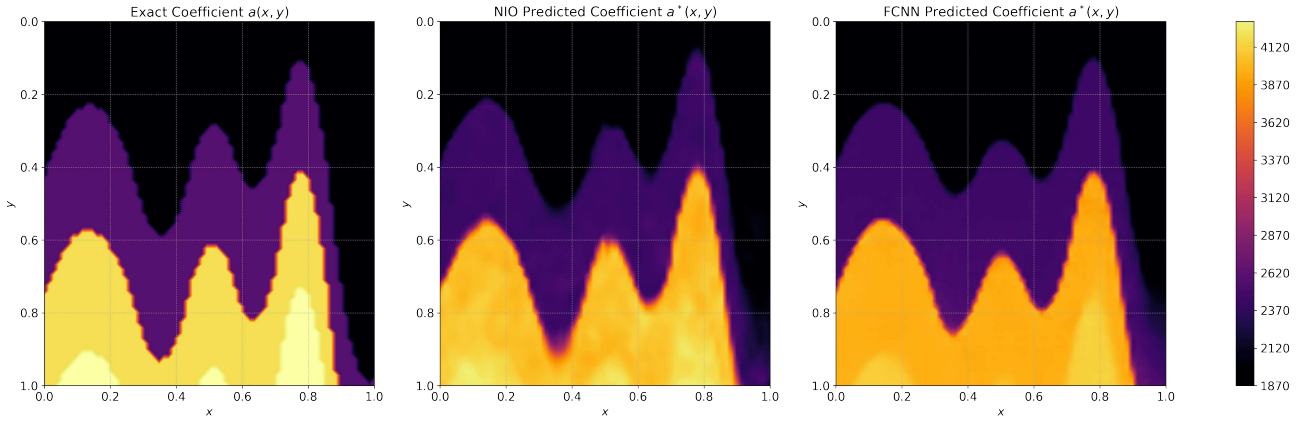
Inverse problems, such as the abstract PDE inverse problem (4), can be very sensitive to noise as the stability estimate (5) indicates, and reconstruction methods have to show some robustness with respect to noisy measurements in order to be practically useful. To test the robustness of CNIO (and competing models) to noise, we take all the benchmark test problems reported in Table 1 of the main text and add varying amounts of noise to the inputs to each model at test time. The resulting test errors, with varying noise levels, are presented in Table 6. This table shows that CNIO (as well as DOnet and FCNN) is very robust to this measurement noise. Even adding 10% noise to the inputs at the time of inference only leads to a minor deterioration of accuracy.

	DONet			**FCNN**			**CNIO**		
$\boldsymbol{\sigma}$	1%	5%	10%	1%	5%	10%	1%	5%	10%
Calderón Problem Trigonometric	1.58%	1.89%	2.76%	1.25%	1.74%	2.8%	0.72%	1.13%	1.86%
Calderón Problem Heart&Lungs	0.43%	0.47%	0.61%	0.18%	0.28%	0.48%	0.10%	0.15%	0.22%
Inverse Wave Scattering	2.65%	2.65%	2.76%	1.24%	1.25%	1.28%	0.91%	0.97%	1.0%
Radiative transport	2.20%	2.92%	3.05%	1.59%	2.06%	2.87%	1.23%	1.35%	1.58%
Seismic Imaging CurveVel - A	3.91%	4.05%	4.42%	2.65%	2.67%	2.65%	2.67%	2.79%	3.00%
Seismic Imaging Style - A	3.45%	3.68%	4.00%	3.12%	3.12%	3.14%	3.03%	3.25%	3.65%

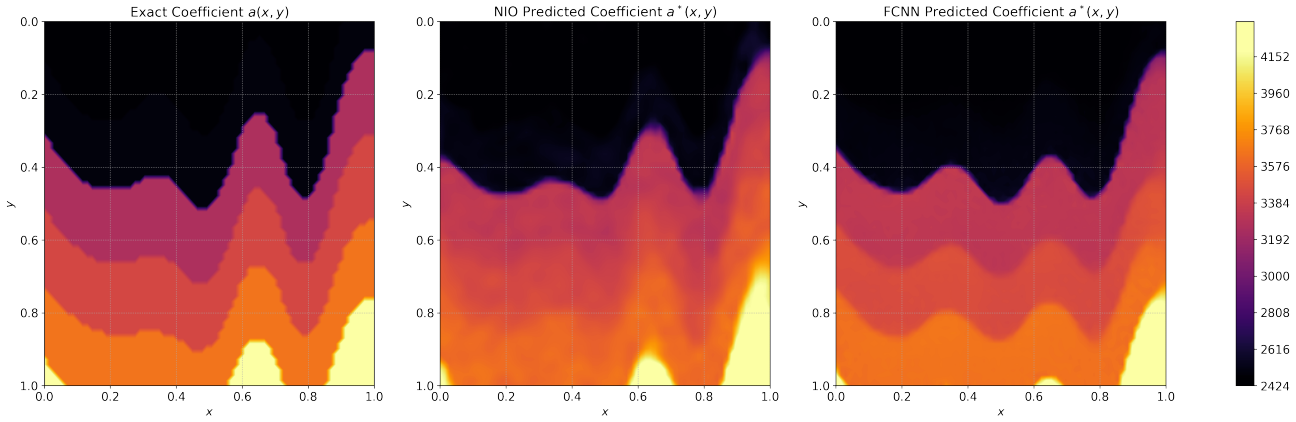
Table 6. Relative median L^1-error computed over σ-noisy testing samples for different benchmarks with different models.

D.3. Robustness of Reconstructions to Varying Grid Sizes.

Although the inputs and outputs to the inverse problem (4) are continuous objects in principle, in practice, one has to deal with discretized versions of both inputs and outputs. This is true when the ground truth is generated by numerical simulations and observed through other forms of measurement. It is highly desirable that an operator learning algorithm be robust to the resolutions at which it is tested; see (Kovachki et al., 2021b) for further discussion on this topic. To test if the proposed NIO architecture is robust with respect to resolution, we focus on the inverse wave scattering with the Helmholtz equation

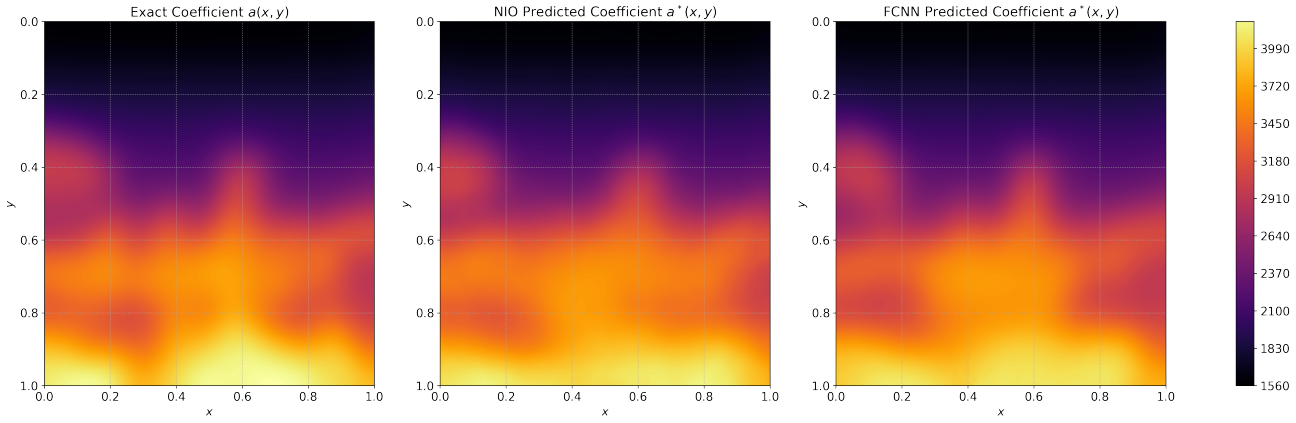


(a) Test Sample 1

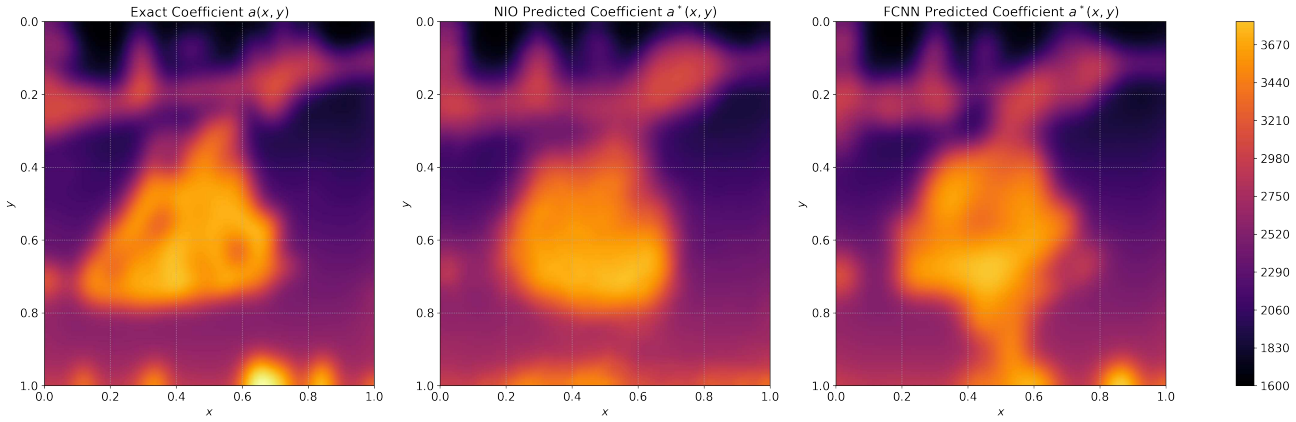


(b) Test Sample 2

Figure 16. Exact and predicted coefficients for two different test samples (Rows) for the Seismic Imaging with the acoustic wave equation with *CurveVel A* data set. Left Column: Ground Truth. Middle Column: CNIO reconstruction. Right Column: FCNN Reconstruction.



(a) Test Sample 1



(b) Test Sample 2

Figure 17. Exact and predicted coefficients for two different test samples (Rows) for the Seismic Imaging with the acoustic wave equation with *Style A* data set. Left Column: Ground Truth. Middle Column: CNIO reconstruction. Right Column: FCNN Reconstruction.

example, where CNIO was trained with data obtained from a finite difference scheme on a uniform 70×70 grid. To test the robustness with respect to resolution, we use this trained model to also infer at two different resolutions, namely at 50×50 and 100×100, and present the results, together with DeepONet and FCNN baselines in Table 7 to observe that CNIO (and the baselines) is robust to varying resolutions. The accuracy does not change much, even if the test resolutions are changed.

	DONet		**FCNN**		**CNIO**	
Resolution	50×50	100×100	50×50	100×100	50×50	100×100
Inverse Wave Scattering	2.65%	2.44%	0.79%	0.79%	0.57%	0.53%

Table 7. Relative median L^1-error computed over testing samples generated at different resolutions (grid sizes).