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 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.

[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. Correspondence to: Roberto Molinaro <roberto.molinaro@sam.math.ethz.ch>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

Given the widespread success of operator learning and other deep learning-based algorithms in the context of forward 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 order 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 significantly outperforms baselines on all these benchmarks.
- Given the inherent sensitivity of PDE inverse problems to noise and perturbations, we also demonstrate empirically that NIO is robust with respect to these factors as well as to the number and location of sensors and underlying grid resolutions. Moreover, we also show that, unlike baselines, NIO can handle fluctuations in the number of input measurements as well as generalize *out-of-distribution* without any significant loss of accuracy. Furthermore, NIO is found to be several orders of magnitude *faster* while still being more accurate than traditional PDE-constrained optimization methods.

Thus, we propose a new learning framework for the fast, robust and accurate solution of PDE inverse problems.

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, $0 \leq a, \sigma_a \in C(D)$. 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, $\Gamma_\pm = \{(z,v) \in \partial D \times V : \pm n_z \cdot v \geq 0\}$ 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., $u(\cdot,0) = u_t(\cdot,0) = 0$ and suitable boundary conditions. In particular, *sources* are placed on a subset of the boundary and it is common to consider point sources $s(t,z) = g(t)\delta_S(z)$ with $g(t) \in L^2([0,T])$ and $\delta_S(z)$ being the Dirac measure concentrated on a set $S \subset \partial D$. These waves are transmitted, reflected, and refracted through the medium. Under certain assumptions (see Sec. 2.3 in Symes (2009)), the *effective* source s can be treated as in $L^2([0,T] \times D)$, which ensures the well-posedness of the PDE. The resulting signal is recorded at a set of receivers $\mathcal{R} \subset \partial D$ on the boundary that take continuous measurement in time $[0,T]$. The boundary observation operator (2) for this wave inverse problem is the *Source-to-Receiver* (StR) operator,

$$\begin{aligned} &\Lambda_a : L^2([0,T] \times D) \mapsto L^2([0,T]; X_{\mathcal{R}}), \\ &\Lambda_a : s \mapsto u\big|_{[0,T]\times\mathcal{R}}, \end{aligned} \tag{15}$$

where $X_{\mathcal{R}}$ is the metric space for the (discrete) set $\mathcal{R}$. The inverse problem that underpins seismic imaging is

$$\begin{aligned} \mathcal{F}^{-1} :&\quad \mathcal{L}\left(L^2([0,T] \times D), L^2([0,T]; X_{\mathcal{R}})\right) \mapsto L^\infty(D), \\ \mathcal{F}^{-1} :&\quad \Lambda_a \mapsto a = \mathcal{F}^{-1}(\Lambda_a), \end{aligned} \tag{16}$$

with Λ_a being 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:

1. 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 $\mathcal{F}^{-1}$.
2. The learning task requires us to learn *mappings from operators to functions* for $\mathcal{F}^{-1}$ defined in (4).
3. 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, $\mathcal{N}^{\text{DON}} : \mathcal{X} \to \mathcal{Y}$, given by

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

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**) $\mathcal{N}^{\text{FNO}}$ (Li et al., 2021a) is a composition

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

For simplicity let us define $\mathcal{M} = Q \circ \mathcal{L}_T \circ \cdots \circ \mathcal{L}_1$, and $\mathcal{N}^{\text{FNO}} = \mathcal{M} \circ R$, where $R : (x, \overline{u}) \mapsto R(\overline{u}(x), x)$ is a "lifting operator" represented by a linear transformation $R : \mathbb{R}^{d_u \times d} \to \mathbb{R}^{d_v}$ 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 nonlinear projection, instantiated by a shallow neural network, such that $v^{T+1}(x) \mapsto \mathcal{N}^{\text{FNO}}(\overline{u})(x) = Q\left(v^{T+1}(x)\right)$. 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 coefficient 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).

3.4. The Architecture.

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

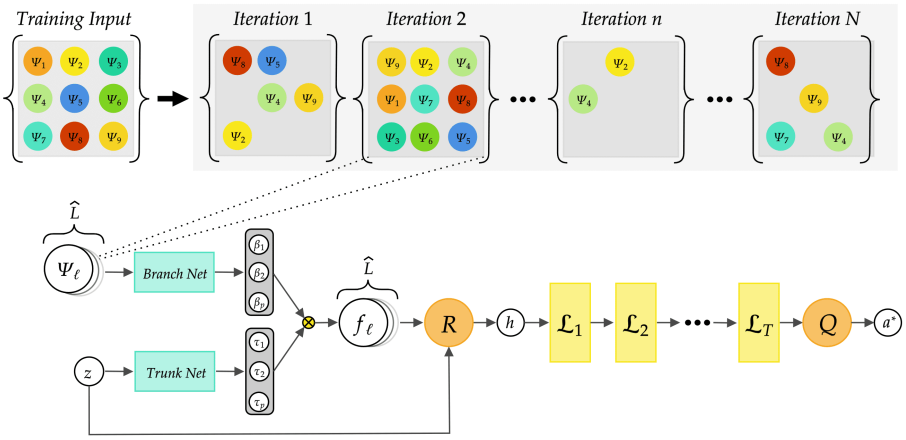


Figure 1. Schematic representation of Neural Inverse Operator (NIO) architecture, with R given by equation (27), and Randomized Batching.

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 nonlinear mode mixing operator $\mathcal{M}$. Instead, one can do so by passing the outputs of the DeepONet to the nonlinear layers of an FNO (20), which also performs the final inversion operator $\mathcal{I}$.

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,

$$\begin{aligned}\mathcal{N}_{NIO} : \begin{pmatrix} z \\ \{\Psi_\ell\}_{\ell=1}^L \end{pmatrix} &\overset{\tau,\beta}{\longmapsto} \begin{pmatrix} \{\tau_k(z)\}_{k=1}^p \\ \{\beta_k\}_{k=1}^p \end{pmatrix} \dots \\ \dots &\overset{\mathcal{N}^{DON}}{\longmapsto} \{f_\ell(z)\}_{\ell=1}^L \overset{R}{\longmapsto} h(z) \overset{\mathcal{M}}{\longmapsto} a^*(z),\end{aligned} \tag{25}$$

for approximating the abstract inverse map $\mathcal{F}^{-1}$ (4). Here, the linear map R can be explicitly defined as

$$h(z) = R(f_1, \dots, f_L, z) = \frac{1}{L}\sum_{\ell=1}^{L} D_\ell f_\ell + Ez, \tag{26}$$

with $E, D_\ell \in \mathbb{R}^{d_v}$. In other words, the inputs $z \in D$ and $\Psi_\ell = \Lambda_a(g_\ell)$ (2), for $1 \leq \ell \leq L$, 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 first linearly transformed through R yielding $h(z)$, and finally *mixed* by the nonlinear component $\mathcal{M}$ of FNO resulting in an approximation of the underlying coefficient a^*. We observe that the DeepONet $\mathcal{N}^{DON}$ in $\mathcal{N}^{NIO}$ (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.

It is important to note that the model takes only z and $\{\Psi_\ell\}_{\ell=1}^L$ as input, rather than z and the input-output pair $\{(g_\ell, \Psi_\ell)\}_{\ell=1}^L$. This choice is motivated by the following consideration: let us define μ_g as the underlying measure (distribution) on the boundary data g, where $g_\ell \sim \mu_g$, and $\mu_\Psi = \Lambda_a \# \mu_g$ represents the pushforward measure given the boundary operator Λ_a. Hence, $\Psi_\ell \sim \mu_\Psi$ for all $\ell = 1, ..., L$, with $\{\Psi_\ell\}_{\ell=1}^L$ representing the empirical distribution approximating μ_Ψ. This, together with the injectivity of the boundary observation operator (2), implies that $\{\Psi_\ell\}_{\ell=1}^L$ suffices to provide statistical information about the operator Λ_a given μ_g satisfying certain properties. We denote L as the number of samples discretizing the measure μ_Ψ or, with abuse of notation, the operator Λ_a.

The construction above addresses point (1) in Section 3.1. However, it does not necessarily satisfy point (2) in its current form. For the model to effectively process μ_Ψ as the input rather than a particular discrete realization of it, the following desirable properties should be met:

- The architecture should exhibit invariance under permutations of the input measurements $\{\Psi_\ell\}_{\ell=1}^L$ as they are i.i.d. samples of μ_Ψ.
- The learning framework must be able to handle an empirical measure of the distribution μ_Ψ with an arbitrary sample size $\widetilde{L}$. In particular, the input sample size at the training and testing stages could be different.
- The performance of the model should be independent of the sample size $\widetilde{L}$ used to discretize μ_Ψ.

To address the first two points, we can modify the architec-

ture by redefining the linear transformation R as follows:

$$h(z) = R(f_1, \dots, f_L, z) = \frac{D}{L} \sum_{\ell=1}^{L} f_\ell + Ez, \quad E, D \in \mathbb{R}^{d_v} \tag{27}$$

To ensure the performance is independent of $\widetilde{L}$, a naive and inefficient approach would involve constructing all possible permutations of $\{\Psi_\ell\}_{\ell=1}^{L}$ with size $\widehat{L}$ for all $\widehat{L} = 2, \dots, L$ and providing them as input to the model. However, inspired by the well-known *bagging* algorithm widely used in machine learning, we propose an efficient and novel method to tackle this point which we term *randomized batching*. In this approach, given the sequence $\{\Psi_\ell\}_{\ell=1}^{L}$, during each training iteration, an integer number $\widehat{L}$ is randomly drawn from the set $\{2, \dots, L\}$. Then, $\widehat{L}$ samples are randomly picked from $\{\Psi_\ell\}_{\ell=1}^{L}$ and the new sequence $\{\Psi_k\}_{k=1}^{\widehat{L}}$ are fed into the model at each iteration during training.

We refer to the architecture (25) with the linear transformation R given in (27), and incorporating the *randomized batching* algorithm, as *Neural Inverse Operator* (see Figure 1 for a schematic representation of the architecture).

4. Empirical Results.

We empirically test NIO on benchmark PDE inverse problems below. The exact details of the training, as well as the architecture and hyperparameter choices, are presented in **SM** D.

As baselines in the following experiments, we choose two models. First, we consider a DeepONet with a CNN as the branch net in (17), which performs mixing of the input function within the branch itself

$$\mathcal{N}_{DONet} : \begin{pmatrix} z \\ \{\Psi_\ell\}_{\ell=1}^{L} \end{pmatrix} \overset{\tau, \beta_{cqn}}{\longmapsto} \begin{pmatrix} \{\tau_k(z)\}_{k=1}^{p} \\ \{\beta_k\}_{k=1}^{p} \end{pmatrix} \overset{\mathcal{N}^{DON}}{\longmapsto} a^*(z). \tag{28}$$

Second, we consider a fully convolutional image-to-image neural network architecture (details in **SM** D.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 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 $a(x, y) = \exp\left(\sum_{k=1}^{m} c_k \sin(k\pi x)\sin(k\pi y)\right)$, 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 NIO (and the baselines) are presented in Table 1. As the table shows, NIO is the best-performing model, outperforming the next-best FCNN model by almost halving the errors. Moreover, the total errors are very small ($< 1\%$) with NIO.

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 $\ell = 1, \dots, 32$ and $f = [-16, \dots, -1, 1, 14, 15, 16]$. 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 and 2 show that NIO 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 larger error of 8.75% for this numerical inversion test (see **SM** Section E.8 for details).

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, $a(x, y) = \sum_{k=1}^{m} \exp\left(-c(x - c_{1,k})^4 - c(y - c_{2,k})^4\right)$, 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, $m = \mod(\overline{m})$, $\overline{m} \sim \mathcal{U}([1, 4])$ 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 4096 training samples and present the relative (median) test errors, on

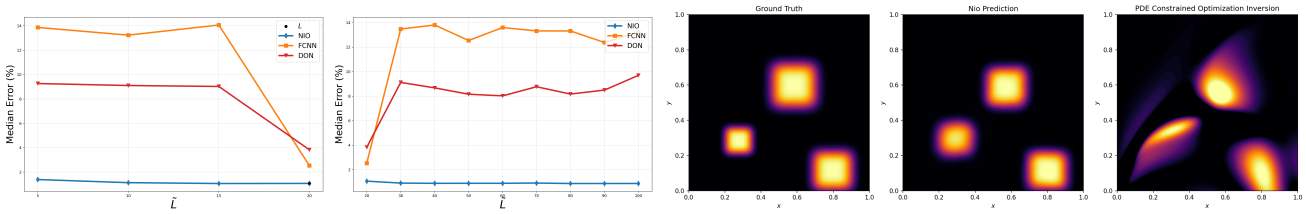


Figure 2. Inverse Wave Scttering. Left Panel: testing median L^1-error VS number of input samples $\widetilde{L}$ (left: $\widetilde{L} < L$, right: $\widetilde{L} > L$). The black dot represents the number of training samples L. Right Panel: ground truth (left), NIO reconstruction (middle), and reconstruction with the PDE constrained optimization method (right) for an out-of-distribution test sample

a test set of 1600 samples, in Table 1. Again, NIO 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 NIO 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-receiver* 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, NIO 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.

4.5. Robustness and Computational Efficiency of NIO.

In the **SM** E, we provide further details on the performance of NIO and its comparison to the considered baselines in terms of several factors. Here, we present a succinct summary of these results. First, inverse problems are characterized by sensitivity to perturbations of different types. Hence, it is imperative to test the robustness of the proposed NIO framework to these factors. To this end, we start with adding noise to the inputs of each of the benchmarks at test time and present the resulting test errors in **SM** Table 8 to observe that NIO (as well as the baselines) are not sensitive to this noise. Next, in **SM** Table 9, we plot the test errors for the models when the underlying grid resolution is varied at test time to observe that NIO is not sensitive to the input grid resolution. Finally, in **SM** Table 10, we present errors when the locations of input sensors are chosen randomly at test time, in contrast to sensors located on a uniform grid at training, to see that NIO is also robust to this variation. However, these factors are less salient in this context than the discretization of the input boundary observation operator. The key element of this discretization is the number of input samples. As argued in the previous section, NIO has been constructed to be *independent* of the number of samples that discretize the boundary observation operator. We test this behavior for the inverse wave scattering problem (see **SM** E.2 for other benchmarks) and present the errors in Figure 2, where the number of input samples at the testing stage are decreased (left) and increased (right) at test time. We clearly see that NIO is very robust to this fluctuation, completely contrasting to the baselines whose performance

deteriorates, and the test errors increase by almost an order of magnitude. This demonstrates that NIO genuinely learns the underlying inverse map rather than just a discrete representation of it. Finally, in **SM** E.6, it is shown that NIO *generalizes* very well *out-of-distribution* for a variety of problems, outperforming baselines considerably.

We conclude this section by comparing NIO with the well-established PDE-constrained optimization algorithms. To this end, we consider an out-of-distribution example for the inverse wave scattering benchmark and compare the results of NIO and PDE-constrained optimization (see details in **SM** E.8) in Figure 2 (Right) to observe that NIO is much more accurate in reconstructing the scatterers than PDE-constrained optimization. The L^1-test errors are 2.3% for NIO and 11.1% for PDE-constrained optimization. Moreover, it took less than 1 sec of inference time for NIO (on a CPU) compared to a single GPU training time of 8.5 hours for the PDE-constrained optimization problem. Thus, this experiment demonstrates the real efficiency of NIO as it is 5 times more accurate while being 4 orders of magnitude faster than the PDE-constrained optimization algorithm.

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**		**NIO**	
	$L^1\downarrow$	$L^2\downarrow$	$L^1\downarrow$	$L^2\downarrow$	$L^1\downarrow$	$L^2\downarrow$
Calderón Problem Trigonometric	1.97%	2.36%	1.49%	1.82%	**0.85%**	**1.05%**
Calderón Problem Heart&Lungs	0.95%	3.69%	0.27%	1.62%	**0.18%**	**1.16%**
Inverse Wave Scattering	3.83%	7.41%	2.53%	7.55%	**1.07%**	**2.94%**
Radiative transport	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%**

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 composition of DeepONets and FNOs, augmented with suitable architectural priors (definition of R in (27)), and trained with *randomized batching*, to guarantee invariance of the generalization error to the different discretization of the input operator. Our architecture is motivated by the underlying structure of the inverse map. 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 and, more importantly, robust approximations to the unknown coefficients with small errors (see **SM** E). Finally, a series of experiments were also presented to demonstrate that NIO is robust with respect to various factors such as varying sensor locations, grid resolutions, noise, and discretizations of the input operator while being able to generalize *out-of-distribution* and being more accurate and much faster than direct and PDE constrained optimization algorithms. 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.

Acknowledgements

We would like to express our deepest gratitude to Nicholas Nelsen and Tobias Rohner for their constructive feedbacks.

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.

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.

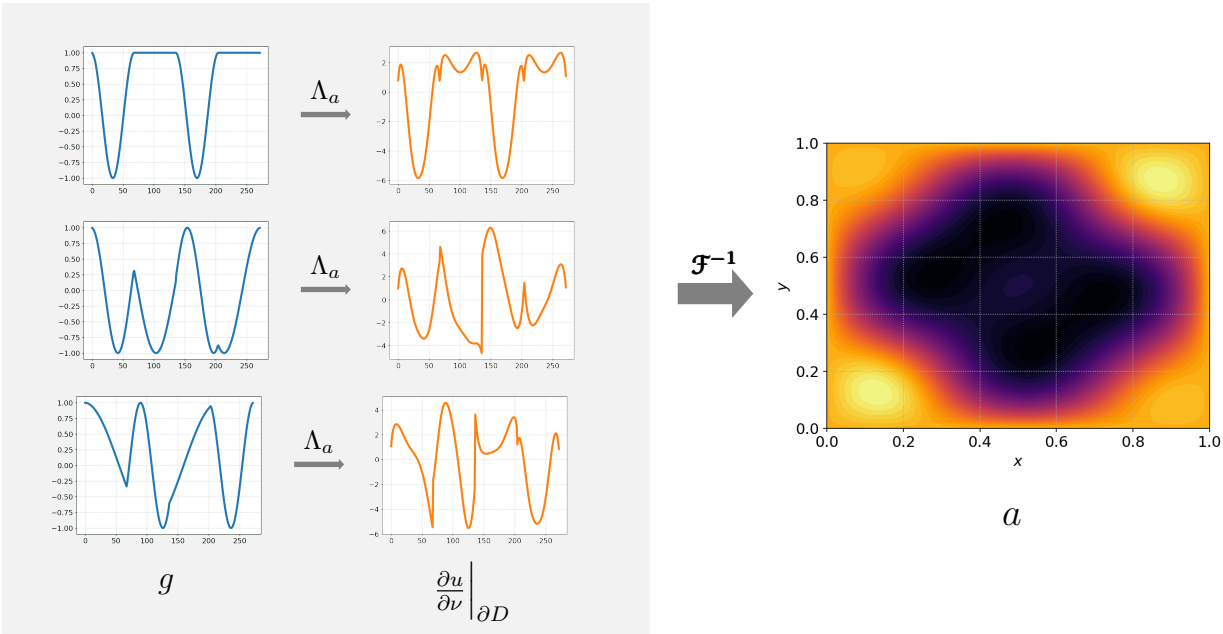


Figure 3. Illustration of a typical input (left) and output (right) sample for the Calderón Problem for EIT with trigonometric coefficients. The input is the Dirichlet-to-Neumann (DtN) map (7), represented here by three Dirichlet Boundary conditions (Voltage) to Current pairs, and the output is the conductivity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

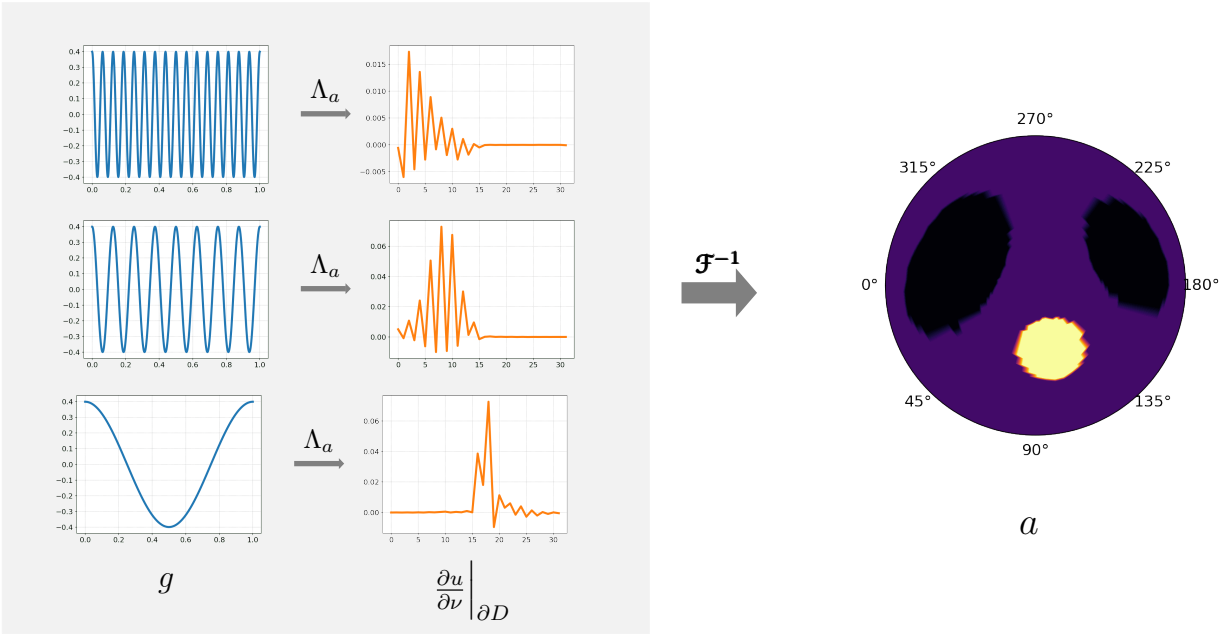


Figure 4. Illustration of EIT for the discontinuous heart-lung Phantom of (Muller & Siltanen, 2012). Left: Input through the DtN (voltage-to-current) map. Right: Conductivity field showing the Phantom of the heart and lungs.

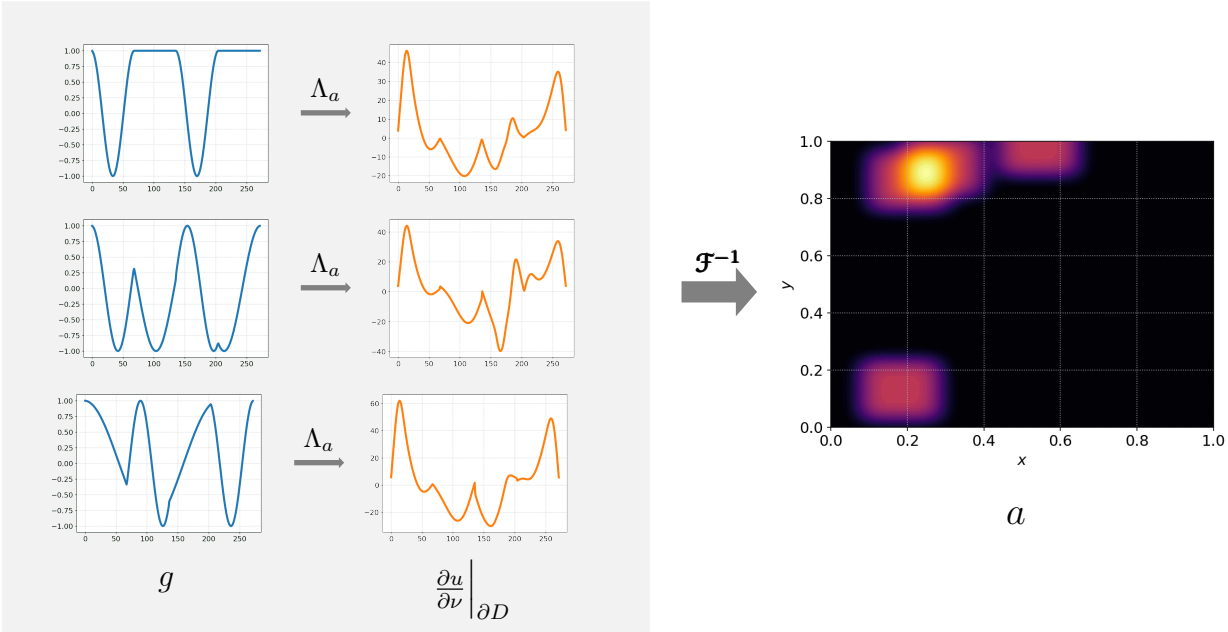


Figure 5. Illustration of detection of inclusions through the Inverse Wave Scattering with the Helmholtz equation. Left: Input represented through 3 samples for the DtN map. Right: Coefficient a.

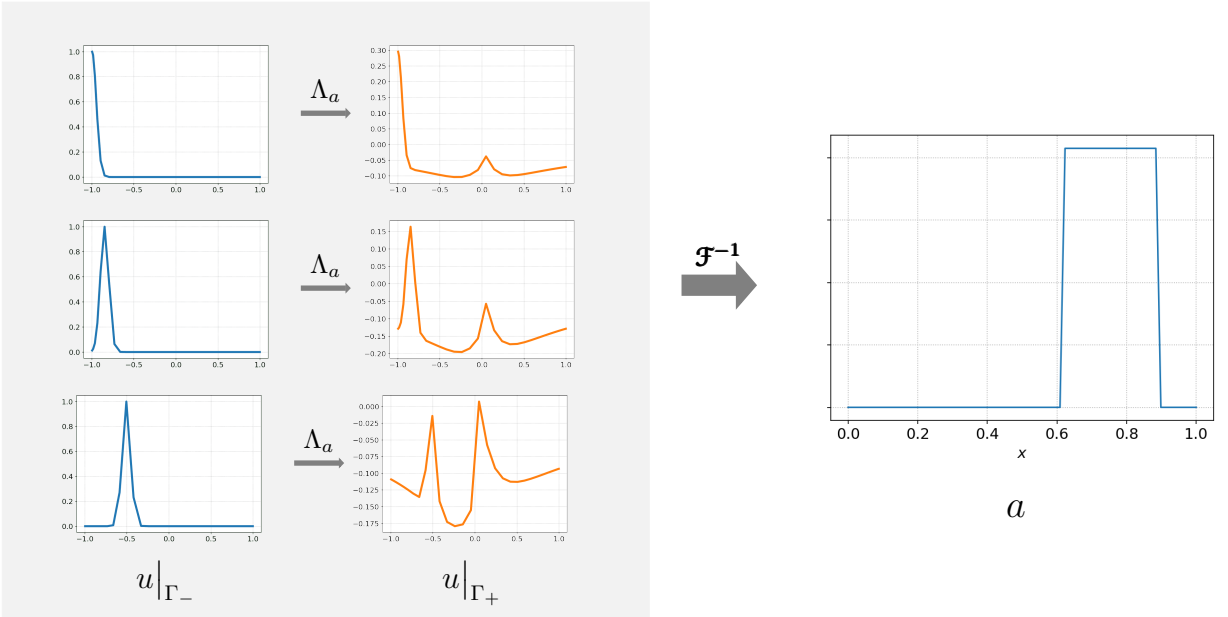


Figure 6. Illustration of Optimal Imaging through the Radiative Transport Equation. Left: Input is the Albedo operator (12) illustrated with three mappings between the inflow and outflow boundaries. Right: Output is the Scattering coefficient.

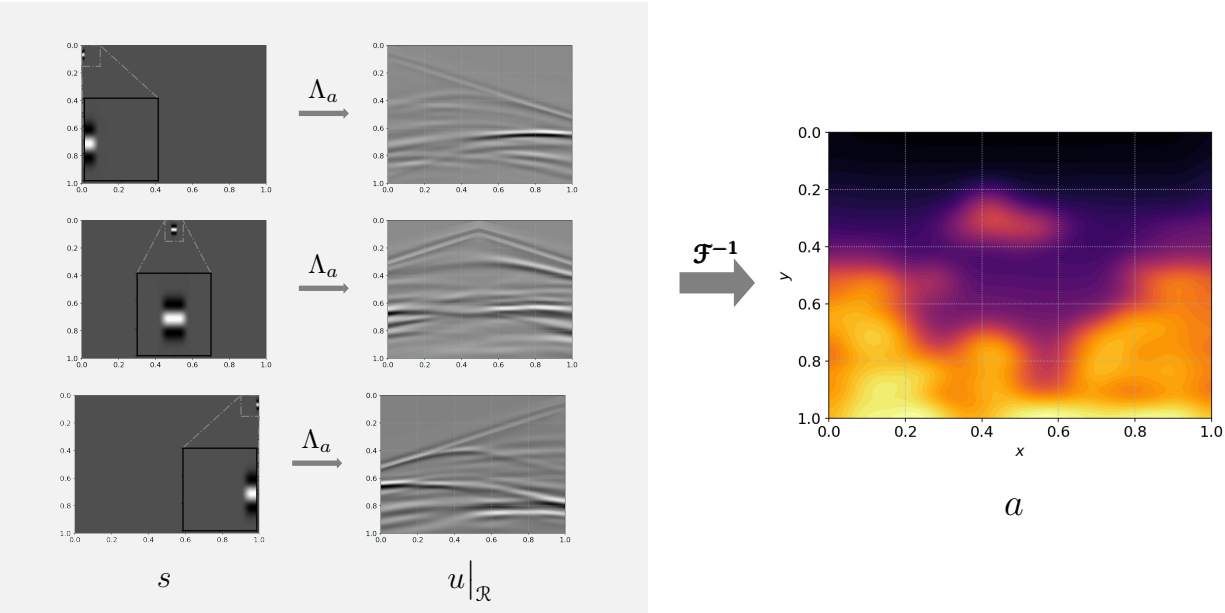


Figure 7. Illustration of Seismic Imaging. Left: Input is Source-to-Receiver map (15) between Incident waves generated at Sources to Temporal signals recorder at Receivers. Right: Output is the velocity coefficient, corresponding to *Style A* dataset of (Deng et al., 2021).

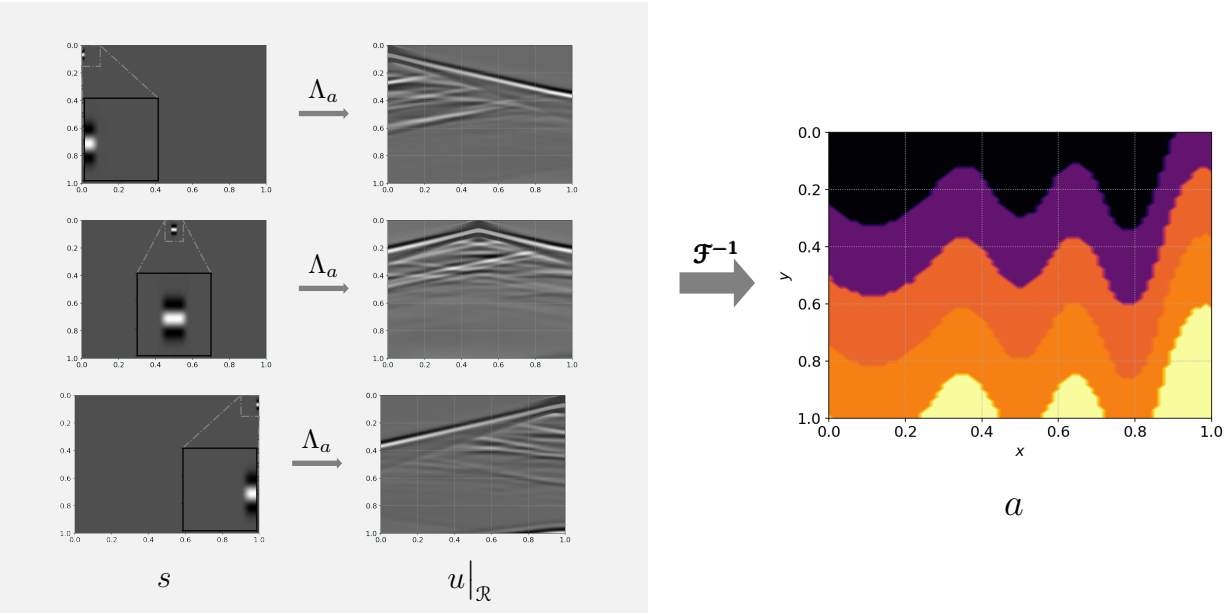


Figure 8. Illustration of Seismic Imaging. Left: Input is Source-to-Receiver map (15) between Incident waves generated at Sources to Temporal signals recorder at Receivers. Right: Output is the velocity coefficient, corresponding to *CurveVel A* dataset of (Deng et al., 2021)

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. Mathematical Description of Heart and Lungs Phantom

To describe the phantom for the heart and lungs, we define the following sets of points on the domain D:

$$\begin{aligned}
s_h &= \left\{ (x,y) \in D \text{ s.t } \sqrt{e_{h,1}(x-c_{h,1})^2 + e_{h,2}(y-c_{h,2})^2} < 0.2 \right\} \\
s_{l_1} &= \left\{ (x,y) \in D \text{ s.t } \sqrt{e_{l_1,1}\left((\cos(\alpha)x_1 + \sin(\alpha)x_2) - c_{l_1,1}\right)^2 + e_{l_1,2}\left((-\sin(\alpha)x + \cos(\alpha)y) - c_{l_1,2}\right)^2} < 0.5 \right\} \\
s_{l_2} &= \left\{ (x,y) \in D \text{ s.t } \sqrt{e_{l_2,1}\left((\cos(\alpha)x_1 + \sin(\alpha)x_2) - c_{l_2,1}\right)^2 + e_{l_2,2}\left((-\sin(\alpha)x + \cos(\alpha)y) - c_{l_2,2}\right)^2} < 0.4 \right\}
\end{aligned} \tag{31}$$

Here, $e_{h,1} = 0.8$, $e_{h,2} = 1$, $e_{l_1,1} = 3$, $e_{l_1,2} = 1$, $e_{l_2,1} = 3$, and $e_{l_2,2} = 1$ represent the eccentricities of the ellipses describing the heart and lungs. The center locations of the heart and lungs are given by $c_{h,1} = -0.1$, $c_{h,2} = 0.4$, $c_{l_1,1} = 0.5$, $c_{l_1,2} = 0.2$, $c_{l_2,1} = -0.6$, and $c_{l_2,2} = 0.1$ and the orientation of the lungs by $\alpha = \frac{\pi}{7}$. Then the body conductivity is defined as,

$$a(x,y) = \begin{cases} a_h & (x,y) \in s_h \\ a_{l_1} & (x,y) \in s_{l_1} \\ a_{l_2} & (x,y) \in s_{l_2} \\ a_b & \text{else} \end{cases}$$

with $a_h = 2$, $a_{l_1} = a_{l_2} = 0.7$, $a_b = 1$.

The training coefficients are obtained by adding 8% white noise to all the parameters above. Specifically, given $y = [e_{h,1}, e_{h,2}, e_{l_1,1}, e_{l_1,2}, e_{l_2,1}, e_{l_2,2}, c_{h,1}, c_{h,2}, c_{l_1,1}, c_{l_1,2}, c_{l_2,1}, c_{l_2,2}, a_h, a_{l_1}, a_{l_2}]$, we define the perturbed version of the parameter vector y as $\widetilde{y} = y(1 + 0.08\xi)$, where $\xi \sim \mathcal{N}(0,1)$ is a random variable drawn from the standard normal distribution. The coefficient is then defined as:

$$\widetilde{a}(x,y) = \begin{cases} \widetilde{a}_h & (x,y) \in \widetilde{s}_h \\ \widetilde{a}_{l_1} & (x,y) \in \widetilde{s}_{l_1} \\ \widetilde{a}_{l_2} & (x,y) \in \widetilde{s}_{l_2} \\ \widetilde{a}_b & \text{else} \end{cases}$$

where $\widetilde{s}_h$, $\widetilde{s}_{l_1}$ and $\widetilde{s}_{l_2}$ are defined as above, but with the parameters replaced by their corresponding perturbed values in $\widetilde{y}$.

D. Architecture and Training Details

Below, details concerning the model architectures and training are discussed. The implementation of the experiments of the paper is realized within the PyTorch framework and available at

- `https://github.com/mroberto166/nio.git`

D.1. Architecture Details

D.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 nonlinear activation functions within units (Goodfellow et al., 2016), resulting in the representation,

$$u_\theta(y) = C_{L_t} \circ \sigma \circ C_{L_t-1} \ldots \circ \sigma \circ C_2 \circ \sigma \circ C_1(y). \tag{32}$$

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

$$C_\ell z_\ell = W_\ell z_\ell + b_\ell, \text{ for } W_\ell \in \mathbb{R}^{d_{\ell+1} \times d_\ell}, z_\ell \in \mathbb{R}^{d_\ell}, b_\ell \in \mathbb{R}^{d_{\ell+1}}, \tag{33}$$

and denote,

$$\theta = \{W_\ell, b_\ell\}_{\ell=1}^{L_t}, \tag{34}$$

to be the concatenated set of (tunable) weights for the network. Thus, in the machine learning terminology, a feed-forward neural network (32) 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 $d_\ell = d_{\ell-1} = d$, $1 < \ell < L_t$.

D.1.2. Fully Convolutional Neural Network

Fully convolutional neural networks are a special class of convolutional networks which can be evaluated for virtually any resolution of the input. 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 nonlinear transformations:

$$\begin{aligned} E_{\theta_e}(y) &= C_L^e \circ \sigma \circ C_{L-1}^e \ldots \circ \sigma \circ C_2^e \circ \sigma \circ C_1^e(y), \\ D_{\theta_d}(z) &= C_L^d \circ \sigma \circ C_{L-1}^d \ldots \circ \sigma \circ C_2^d \circ \sigma \circ C_1^d(z), \\ u_\theta(y) &= D_{\theta_d} \circ E_{\theta_e}(y). \end{aligned} \tag{35}$$

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, 11,12. The *convolutional block* (or *transposed convolution block*) is the composition of a convolution (or transposed convolution) operation, batch normalization, and activation function (Leaky ReLU). The cropping operation involves adding negative padding to the edges of a tensor to achieve the desired output width and height. 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).

D.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 12, and a linear transformation from

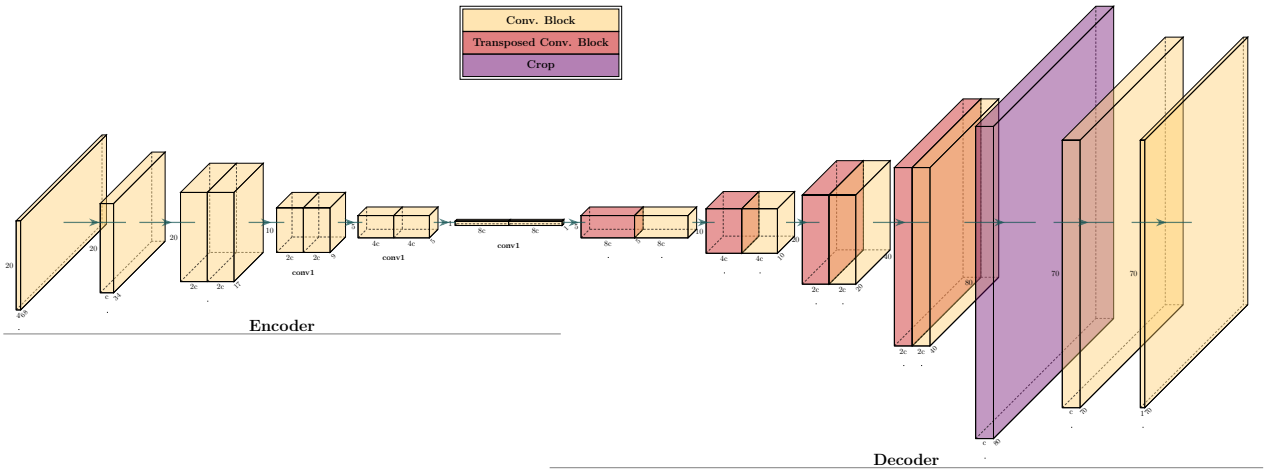


Figure 9. Schematic representation of the Fully-Convolutional Neural Network (FCNN) architecture used for the Calderón problem with Trigononmetric coefficients and for the Inverse Wave Scattering with Helmholtz Equation.

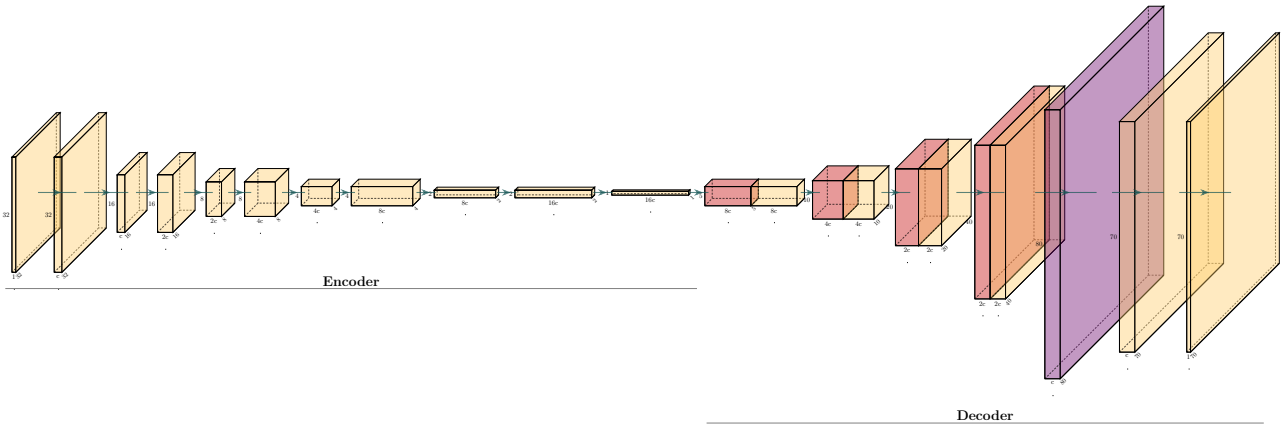


Figure 10. Schematic representation of the Fully-Convolutional Neural Network architecture used for the optical imaging for the Calderón Problem with Heart&Lungs phantom.

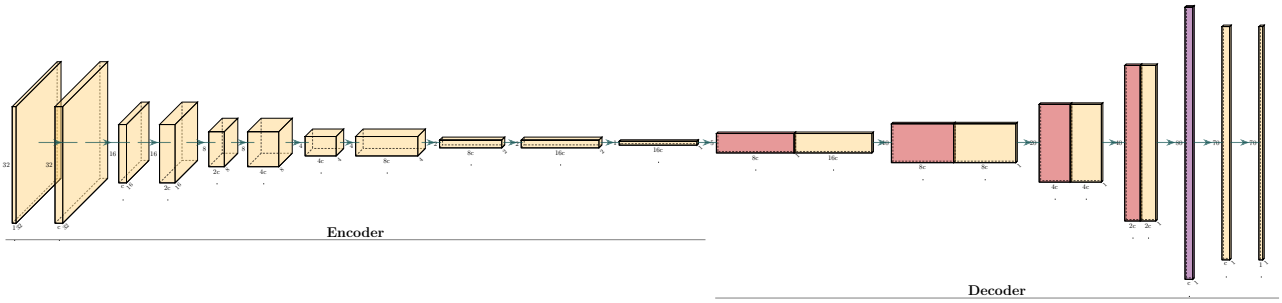


Figure 11. Schematic representation of the Fully-Convolutional Neural Network architecture used for the optical imaging for the radiative transport Equation.

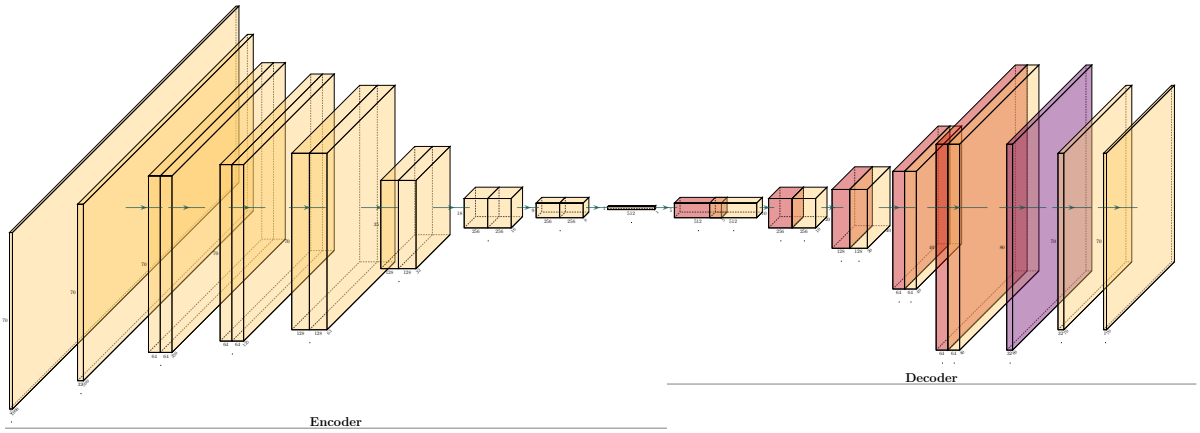


Figure 12. Schematic representation of the Fully-Convolutional Neural Network architecture used for the seismic imaging problems.

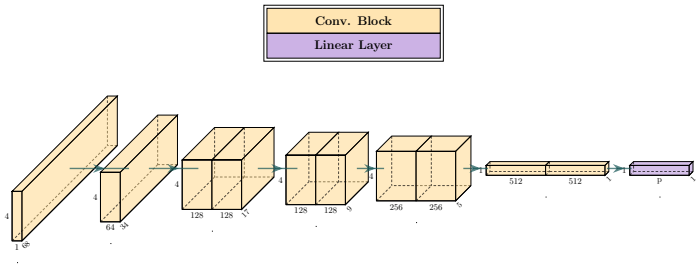


Figure 13. Schematic representation of the NIO-BranchNet architecture used for the Calderón problem with Trigononmetric coefficients and for the Inverse Wave Scattering with Helmholtz Equation

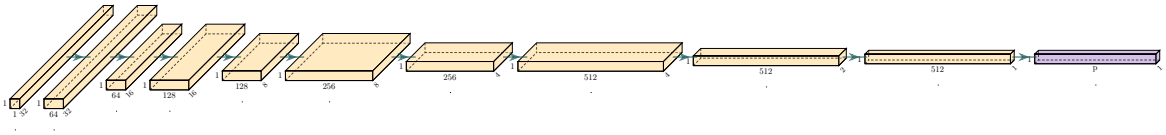


Figure 14. Schematic representation of the NIO-BranchNet architecture used for the optical Imaging with Radiative transport Equation and Calderón Problem with Heart&Lungs phantom.

$\mathbb{R}^n$ 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, $c = 32$ for the seismic imaging and $c = 64$ for all the other benchmarks.

Hence, 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 for both the branch and the trunk.

D.1.4. Fourier Neural Operator

We use the implementation of the FNO model provided by the authors of (Li et al., 2021a). Specifically, 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 $1 < \ell < T$.

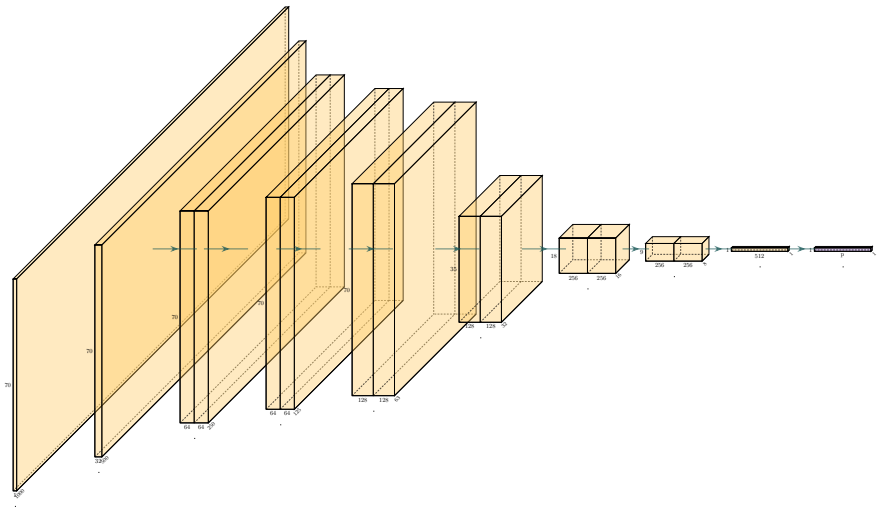


Figure 15. Schematic representation of the NIO-BranchNet architecture used for the seismic imaging.

D.1.5. NEURAL INVERSE OPERATOR

In all numerical experiments, the proposed architecture is constructed by combining the DeepONet and Fourier Neural Operator by means of the lifting operator R defined in equation (27).

The implementation of DeepONet follows the same description as outlined in Section D.1.3. However, the branch structure differs from the *encoder* structure depicted in figures 9, 10, 11, and 12. In the proposed NIO architecture, the channel mixing is performed downstream through the operator $\mathcal{M}$. Specific details regarding the branch-net architectures used in NIO for the benchmark problems can be found in Figures 13, 14, and 15. For instance, for the seismic imaging, the branch architecture is the same as the *encoder* shown in Figure 12, but only a single input channel is used instead of five, following the same rationale as mentioned above. Overall, 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.

D.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 (250 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 two different data transformations to preprocess the data:

1. *MinMax*. This transformation involves scaling both the inputs and outputs to a range between -1 and 1:

$$\widetilde{f} = 2\frac{f - m}{M - m} - 1, \tag{36}$$

   where M and m are the maximum and the minimum value of f across all the *training samples*.

2. *log-MinMax*. This transformation is specifically used for Seismic Imaging problems. The input data are transformed according to the following equation:

$$\widetilde{f} = \log\left(|f|\right) \text{sign}\left(f\right), \tag{37}$$

   and then, the obtained input and output scaled between -1 and 1.

All the parameters mentioned above, including the type of data transformation (*Identity*, *MinMax*, *log-MinMax*, 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. Overall, 50 hyperparameter configurations are tested for NIO and 30 for the baselines. 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 3, 4, and 5. The FCNN hyperparameters reported in the table for the seismic imaging problem are those used in (Deng et al., 2021).

	Calderón Problem Trigonometric	**Calderón Problem Heart&Lungs**	**Inverse Wave Scattering**	**Radiative Transport**	**Seismic Imaging CurveVel - A**	**Seismic Imaging Style - A**
DONet	4.6M 9.07M	4.6M 9.07M	9.54M 14.01M	9.54M 14.01M	12.01M 15.85M	12.01M 15.85M
FCNN	1.07M 68.32M	1.07M 68.32M	4.31M 275.37M	2.48M 39.53M	24.4M 24.4M	24.4M 24.4M
NIO	7.95M 27.79M	10.6M 50.76M	7.95M 27.79M	9.6M 10.3M	13.07M 32.91M	13.07M 32.91M

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 the main text.

D.2.1. Sensitivity to Initialization of the Trainable Parameters

Next, we consider the best performing NIO model corresponding to the hyperparameters reported in Table 5. We then train the model again using multiple initializations of the model parameters, and each model is trained a total of 10 times. Table 6 presents the means and standard deviations of the 10 test median errors for various benchmark experiments.

From the results shown in Table 6, we observe that NIO demonstrates remarkable robustness with respect to random initializations, exhibiting a significantly low standard deviation to mean ratio for the chosen benchmarks. However, it should be noted that the Calderón problem with trigonometric functions is an exception to this trend, as it appears to be quite sensitive to different initializations of the model parameters.

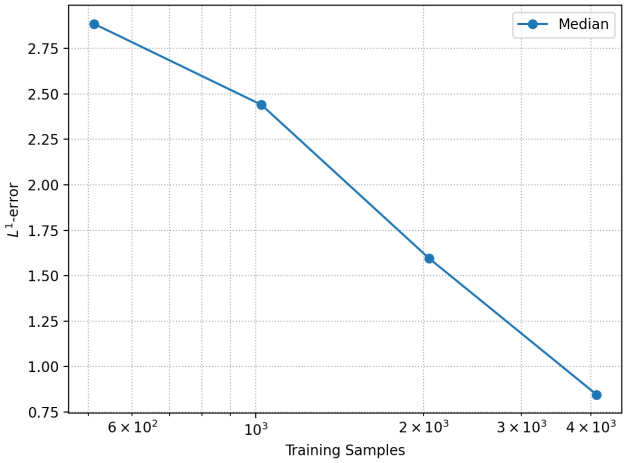
D.2.2. Sensitivity to the Number of Training Samples

Once again, we revisit the best-performing NIO architectures for different benchmark problems and focus on varying the number of training samples and retraining the selected models accordingly. We then plot in Figure16 the median of the testing error as we change the cardinality of the training set. We observe that even for 512 training samples, the accuracy of the models, except for the Inverse Wave Scattering, remains very satisfactory. It should be observed that the number of training samples used in the benchmarks is very small compared to the state-of-the-art data-driven models used in this context.

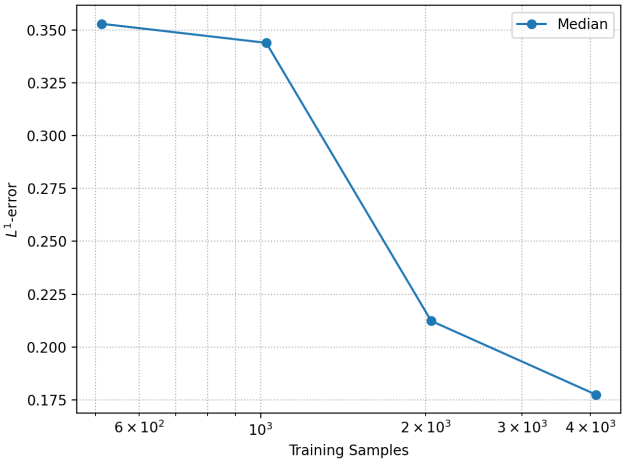
E. Further Experimental Results

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

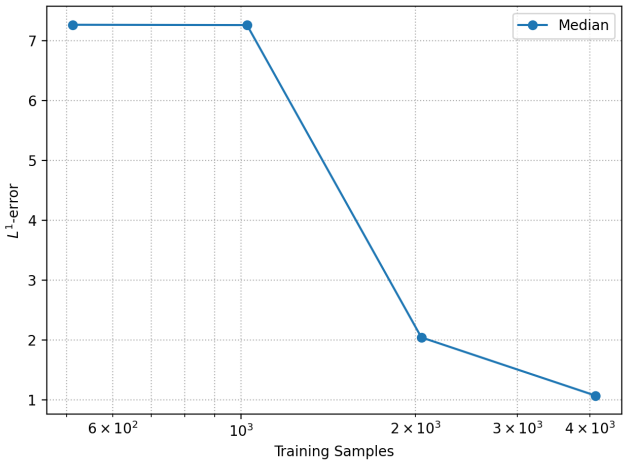
We start by elaborating on the results obtained on the benchmarks and presented in Table 1. In Figure 17, we show two randomly drawn test samples for the Calderón Problem for inferring conductivity with trigonometric coefficients by EIT. For both these test samples, we see that NIO (and FCNN) can accurately approximate the ground truth without any visible artifacts. This observation correlates with very small test errors with NIO. At least for these two samples, there appears to be



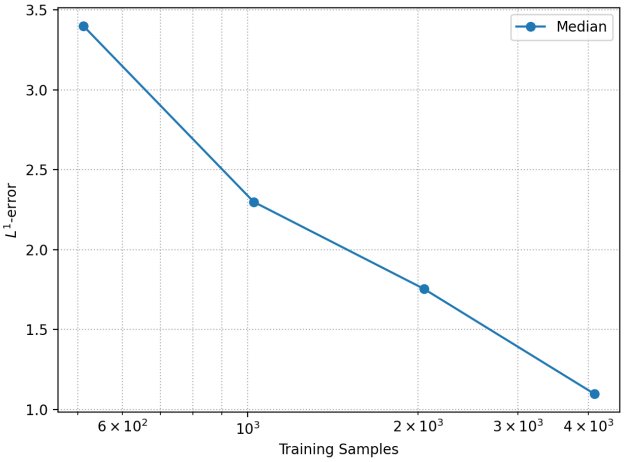
(a) Calderón Problem Trigonometric



(b) Calderón Problem Heart&Lungs



(c) Inverse Wave Scattering



(d) Radiative transport

Figure 16. Median of the L^1-error computed over testing samples VS number of training samples for different benchmarks with NIO.

	η	γ	w	Data Trans	p	L_t	d	Trainable Params
Calderón Problem Trigonometric	0.001	1.0	0.0	Identity	25	8	200	4.84M
Calderón Problem Heart&Lungs	0.001	1.0	0.0	MinMax	100	15	500	13.1M
Inverse Wave Scattering	0.001	1.0	1e-06	Identity	1000	12	500	8.32M
Radiative transport	0.001	1.0	0.0	MinMax	100	15	500	13.1M
Seismic Imaging CurveVel - A	0.001	0.98	1e-06	log-MinMax	400	12	500	15.09M
Seismic Imaging Style - A	0.001	0.98	1e-06	log-MinMax	400	12	500	15.09M

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

	η	γ	w	Data Trans	c	Trainable Params
Calderón Problem Trigonometric	0.001	1.0	0.0	MinMax	16	1.07M
Calderón Problem Heart&Lungs	0.001	1.0	0.0	MinMax	128	275.37M
Inverse Wave Scattering	0.001	1.0	0.0	MinMax	128	68.32M
Radiative transport	0.001	1.0	1e-06	MinMax	16	2.48M
Seismic Imaging CurveVel - A	0.001	1	1e-04	log-MinMax	64	24.4M
Seismic Imaging Style - A	0.001	1	1e-04	log-MinMax	64	24.4M

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

little visible difference between NIO and FCNN. Nevertheless, the results from Table 1 demonstrate that NIO outperforms FCNN considerably on this problem by almost halving the test error.

Next, in Figure 26, we focus on the discontinuous heart-lungs Phantom inferred with EIT. Also, in this case, there is no visual difference between the NIO and FCNN, which are both very accurate in reconstructing the ground truth, and this is indeed consistent with the very low generalization error achieved by both models.

In Figure 19, 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 NIO 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 second sample, the reconstruction with NIO and FCNN are noticeable differences. In particular, FCNN cannot reconstruct the small rectangular scatterer (at the top right of the square domain), whereas NIO can reconstruct it. This possibly explains why NIO is significantly more accurate (see Table 1) for this experiment in reconstructing scatterers.

In Figure 20, 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 NIO 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 20 Right), we see that FCNN gets the correct location but the wrong magnitude of the discontinuity, whereas NIO can approximate both accurately, probably accounting for the significant gain in accuracy on this problem (see Table 1 of main text).

In Figures 21 and 22, 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

	η	γ	w	Data Trans	p	L_t	d	k	d_v	L	Trainable Params
Calderón Problem Trigonometric	0.001	1.0	1e-06	Identity	100	8	100	25	32	4	12.06M
Calderón Problem Heart&Lungs	0.001	1.0	1e-06	MinMax	100	8	100	25	32	4	14.74M
Inverse Wave Scattering	0.001	0.98	0.0	Identity	100	8	200	16	64	4	15.57M
Radiative transport	0.001	0.98	1e-06	MinMax	400	4	100	32	64	4	10.3M
Seismic Imaging CurveVel - A	0.001	0.98	1e-06	MinMax	25	4	200	16	32	3	16.49M
Seismic Imaging Style - A	0.001	0.98	1e-06	log-MinMax	100	8	200	16	64	2	16.49M

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

Calderón Problem Trigonometric	**Calderón Problem Heart&Lungs**	**Inverse Wave Scattering**	**Radiative Transport**
$1.03 \pm 0.45\%$	$0.19 \pm 0.03\%$	$1.08 \pm 0.16\%$	$1.14 \pm 0.18\%$

Table 6. Means $\pm$ standard deviations of relative median L^1 test errors computed over 10 different NIO retrainings for different benchmarks.

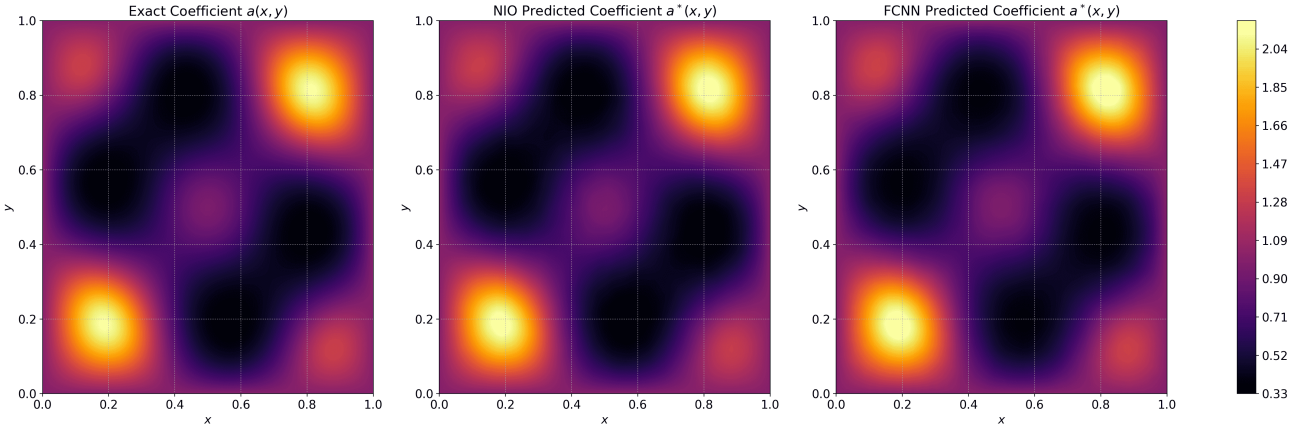
et al., 2021)), respectively. Both figures show that NIO and FCNN reconstruct the coefficient reasonably accurately, although slight differences exist between the models. Nevertheless, coupled with quantitative results from Table 1, we can conclude that NIO is at least on par with FCNN, which was shown to be one of the state-of-the-art models in this context in (Deng et al., 2021).

E.2. Robustness of Reconstruction to Λ_a-Discretizazion.

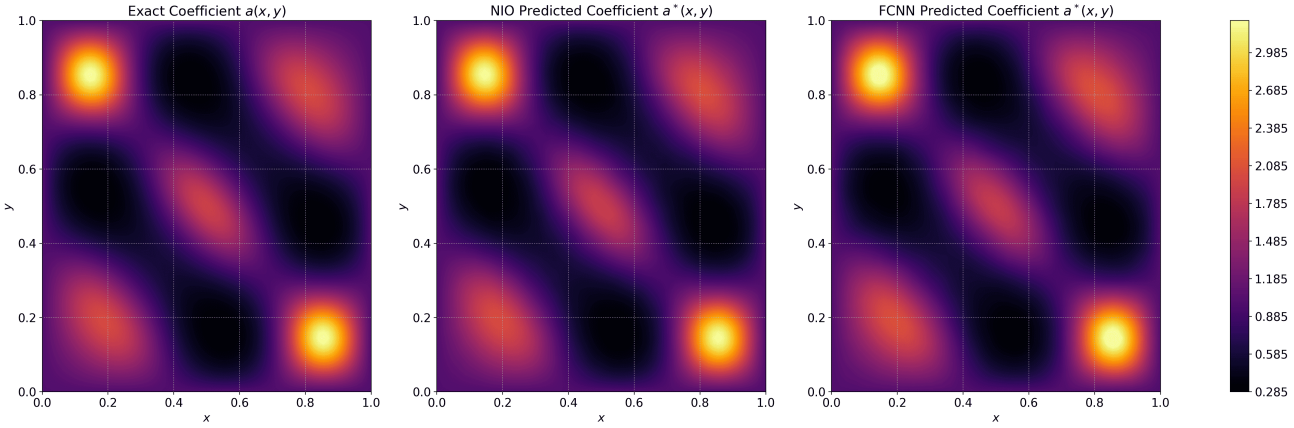
As outlined in Section 3.4, one crucial property that the model should exhibit is robustness to the number of samples $\widetilde{L}$ used to approximate the pushforward measure μ_Ψ. To assess this, we conduct two different experiments.

In the first experiment, we consider the test set used to compute the errors reported in the main table 1 and consisting of input-output pairs $(\{\Psi_\ell\}_{\ell=1}^L, a)$. For each model (NIO and baselines) and each benchmark, we construct a new test set by picking at random $\widetilde{L}$ samples from $\{\Psi_\ell\}_{\ell=1}^L$, $\widetilde{L} \leq L$ and compute the corresponding testing error. As a remark, it should be noted that NIO can be evaluated directly for any input $\{\Psi_k\}_{k=1}^{\widetilde{L}}$, without no change in the architecture. On the other hand, in order to even evaluate the baselines, interpolation (we chose to use the nearest interpolation) must be used to obtain inputs consisting of exactly L samples. In Figure 23, we plot the median L^1-error obtained as a function of the number of samples $\widetilde{L}$ for different models and benchmarks. We observe that the performance of NIO remains invariant with respect to $\widetilde{L}$, with the testing error only showing a *slight* increase as $\widetilde{L}$ decreases, which is typically expected since the approximation of the measure μ_Ψ by the empirical distribution becomes less accurate as the number of samples decreases. In contrast, the performance of the baselines deteriorates even when the number of samples $\widetilde{L}$ is extremely close to the training set size. This demonstrates that the baselines are *not invariant* to permutations of the samples $\{\Psi_\ell\}_{\ell=1}^L$.

Next, we consider the Calderón problem with trigonometric functions and the inverse wave scattering problem. We generate a new testing set from scratch, consisting of input-output pairs $(\{\Psi_\ell\}_{\ell=1}^L, a)$, where $L = 100$ (compared to the $L = 20$ samples used for model training). We then conduct the same experiments as before and present the results in Figure 24. These results further reinforce the fact that NIO exhibits invariance with respect to the discretization of the input measure μ_Ψ.



(a) Test Sample 1



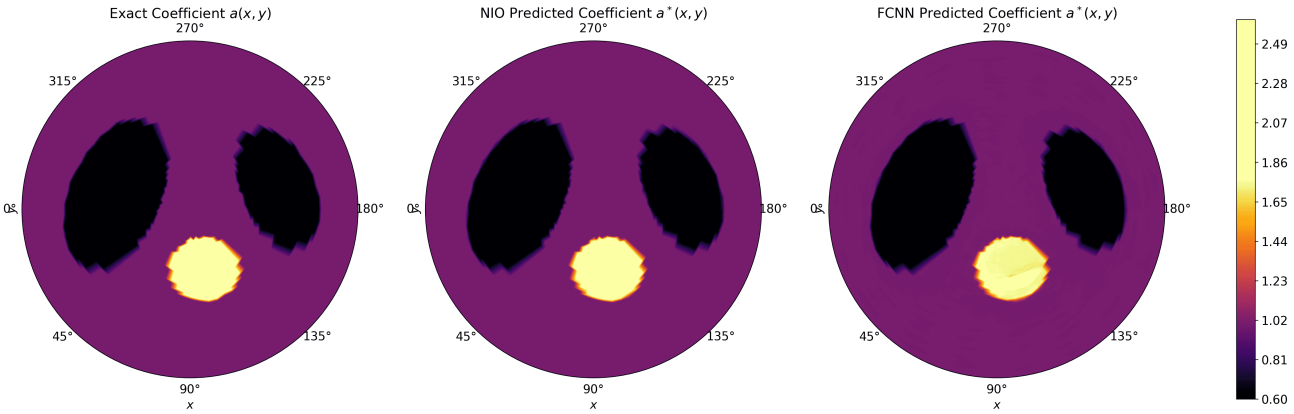
(b) Test Sample 2

Figure 17. Exact and predicted coefficients for two different test samples (Rows) for the Calderón problem with Trigonometric coefficients. Left Column: Ground Truth. Middle Column: NIO reconstruction. Right Column: FCNN Reconstruction.

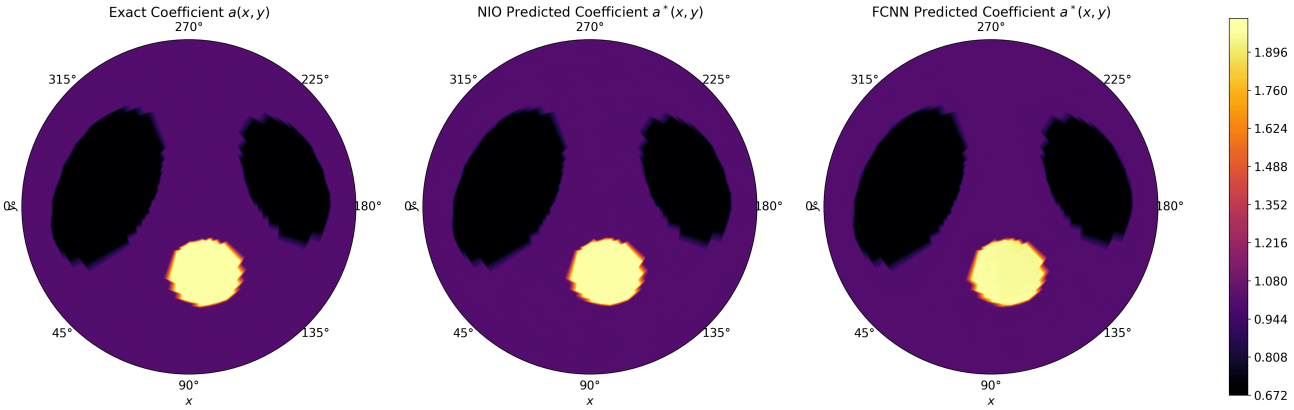
E.3. Robustness of Reconstruction to Noise

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 NIO (and competing models) to noise, we take all the benchmark test problems reported in Table 1 of the main text and add 1% noise to the inputs to each model at test time. Table 8 presents the resulting test errors. This table shows that NIO (as well as DOnet and FCNN) is very robust to this measurement noise.

Furthermore, upon closer examination of the results, it is evident that the models displaying the highest robustness with respect to additional noise are those trained using the *log-MinMax* data scaling transformation. To validate this observation, we consider the inverse wave scattering problem and train NIO with the hyperparameters reported in Table 5, but employing the *log-MinMax* scaling of the data. Additionally, instead of monitoring the validation error computed on the noiseless data, we monitor the validation error computed on data corrupted by 10% noise and interrupt the training based on this metric. The final median testing error on 1%-noisy data is 1.64%, two times lower than the value reported in Table 8. Moreover, the testing error on the noise-free data only marginally increased to 1.61%. These findings suggest that utilizing *log-MinMax* scaling and potentially monitoring the validation error on the corrupted data can significantly enhance the model's robustness to noise, with minimal loss in performance on the noise-free data.



(a) Test Sample 1



(b) Test Sample 2

Figure 18. Exact and predicted coefficients for two different test samples (Rows) for the Calderón problem with Heart&Lungs Phantom. Left Column: Ground Truth. Middle Column: NIO reconstruction. Right Column: FCNN Reconstruction.

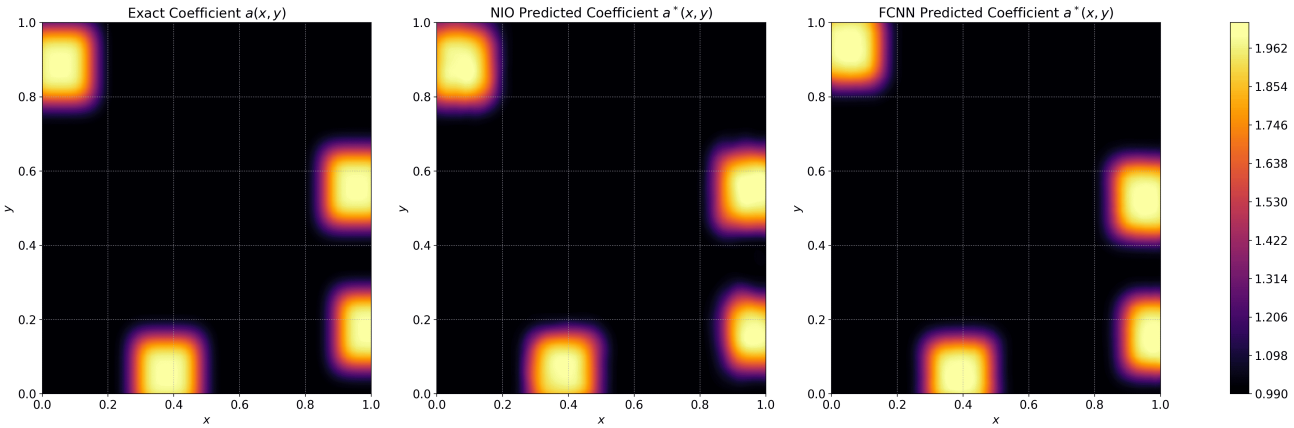
E.4. Robustness of Reconstructions to Varying Grid Sizes.

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 example, where NIO 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 9 to observe that NIO (and the baselines) is robust to varying resolutions.

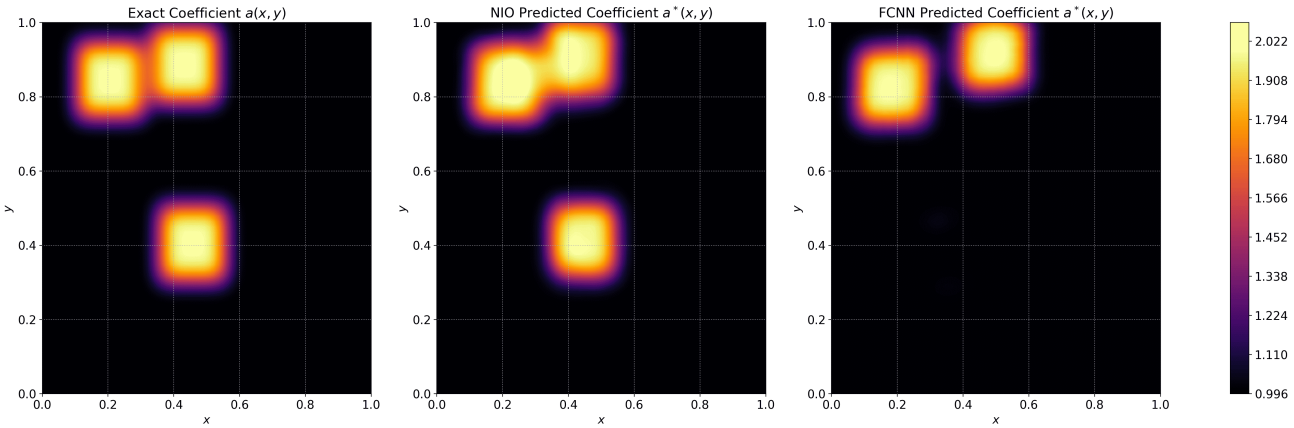
E.5. Robustness of Reconstruction to Random Sensors Location.

While training data typically assumes equidistant placement of sensors along the boundaries of the square domain (as they are synthetically generated using standard numerical methods), real-world scenarios often involve sensors located randomly along the boundaries. Hence, the learning model must exhibit robustness to these random sensor placements.

Our experiments to assess this robustness focus on two specific problems: the Calderón problem with trigonometric function and inverse wave scattering. We perform testing with input data obtained from 200 sensors randomly distributed along the domain boundary. Observe that the training data accounts for 272 sensors. Therefore, before feeding the data to NIO (and baselines), we interpolate it onto the original equispaced set of points. For both problems, we examine the L^1 error, as

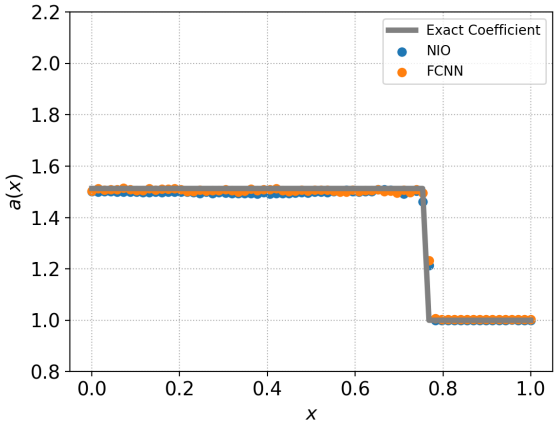


(a) Test Sample 1

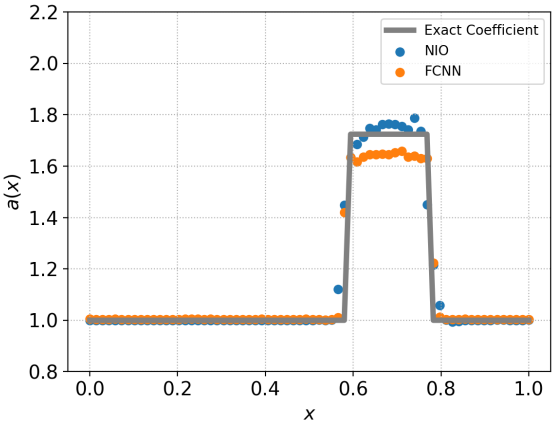


(b) Test Sample 2

Figure 19. Exact and predicted coefficients for two different test samples (Rows) for the Inverse Wave Scattering with Helmholtz Equation. Left Column: Ground Truth. Middle Column: NIO reconstruction. Right Column: FCNN Reconstruction.

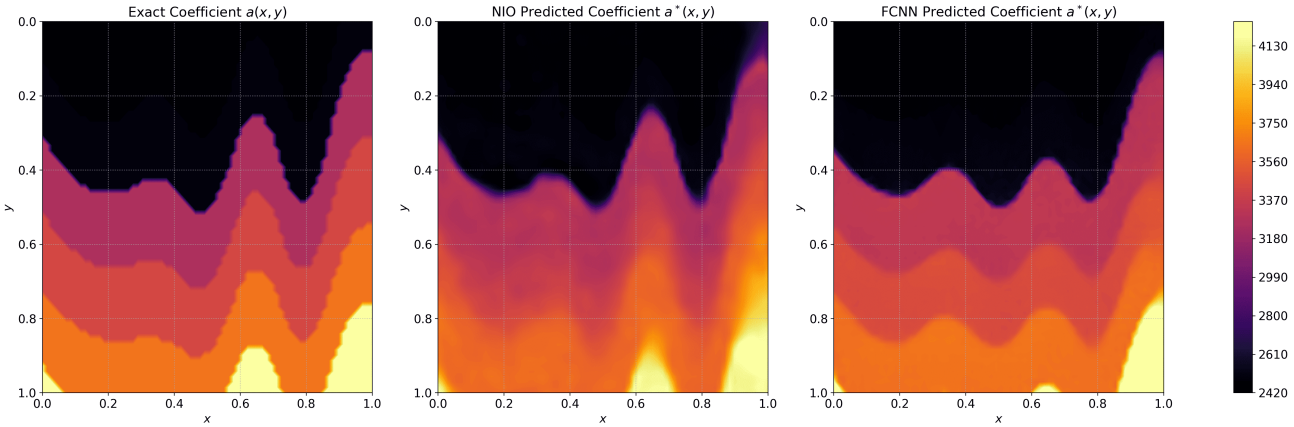


(a) Test Sample 1

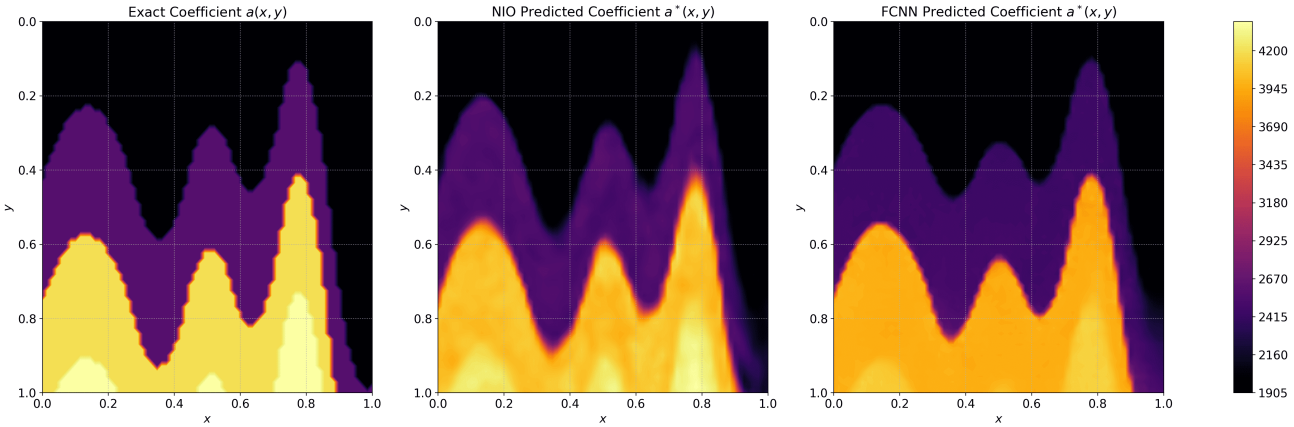


(b) Test Sample 2

Figure 20. Exact and predicted absorption coefficients for two different test samples, obtained with optical imaging for the radiative transport Equation.

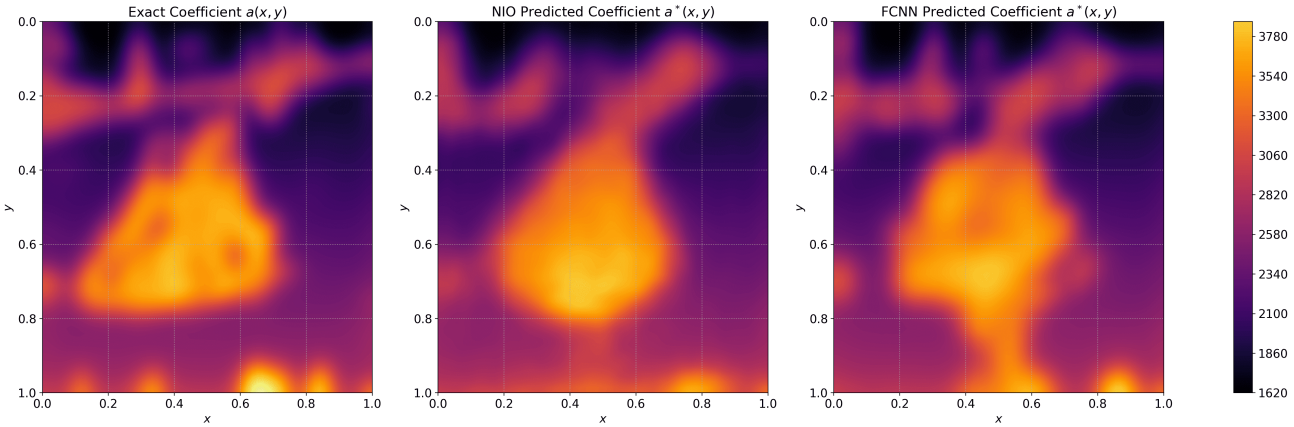


(a) Test Sample 1

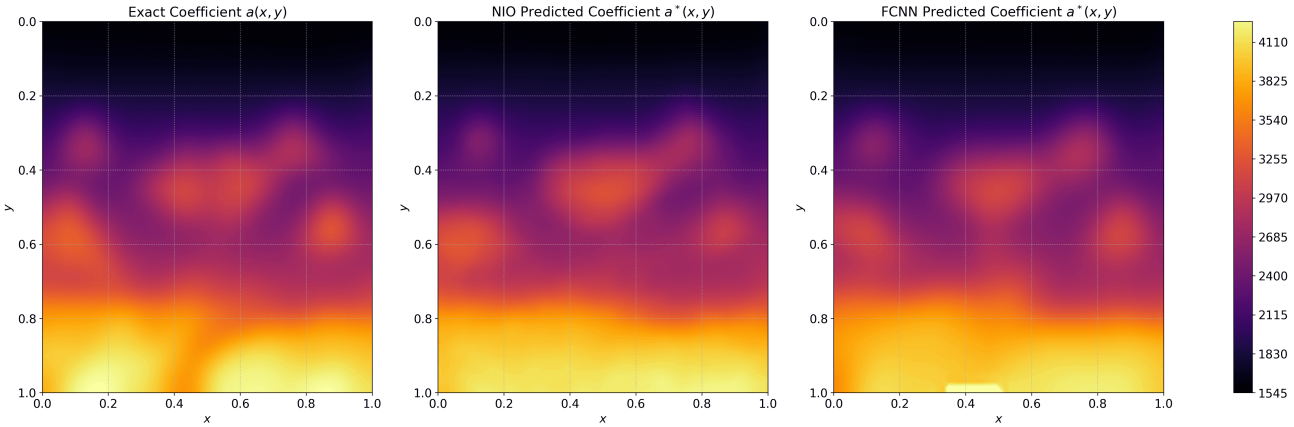


(b) Test Sample 2

Figure 21. 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: NIO reconstruction. Right Column: FCNN Reconstruction.

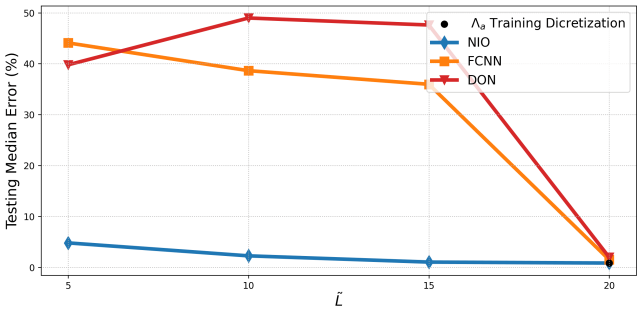


(a) Test Sample 1

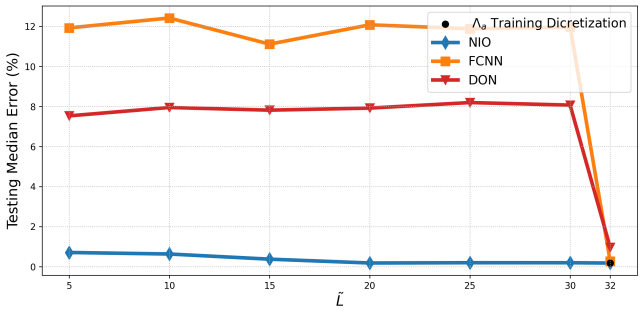


(b) Test Sample 2

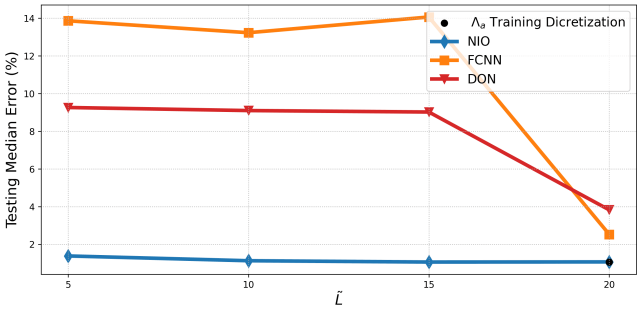
Figure 22. Exact and predicted coefficients for two different test samples (Rows) for the Seismic Imaging with the acoustic wave equation with *Style A* data set. Left Column: Ground Truth. Middle Column: NIO reconstruction. Right Column: FCNN Reconstruction.



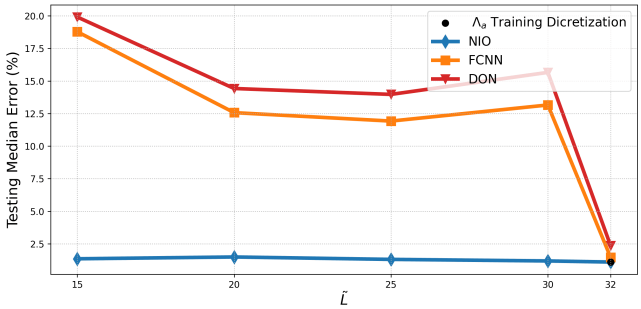
(a) Calderón Problem Trigonometric



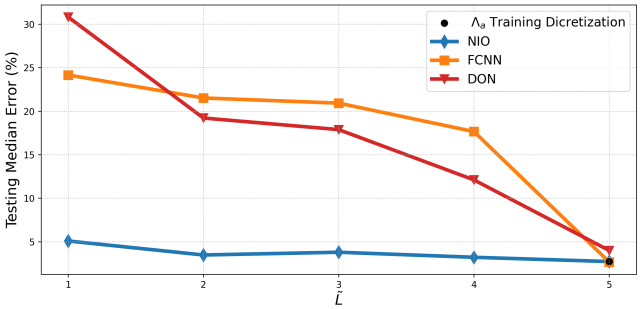
(b) Calderón Problem Heart&Lungs



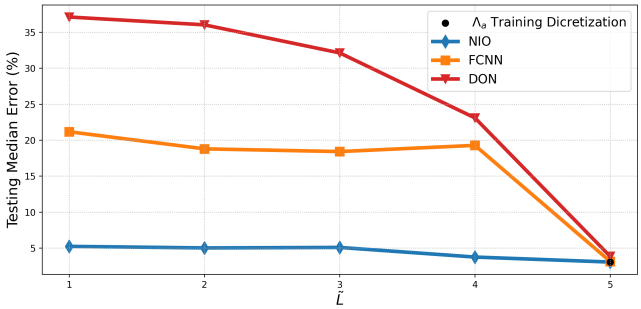
(c) Inverse Wave Scattering



(d) Radiative transport

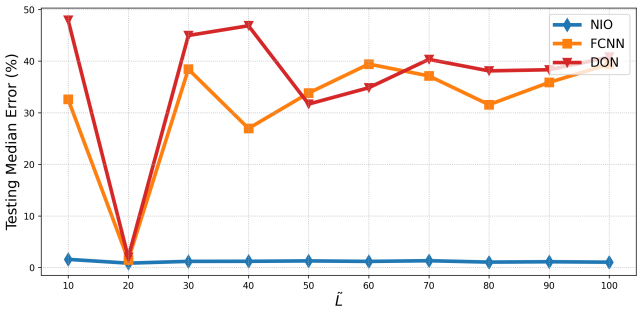


(e) Seismic Imaging CurveVel - A

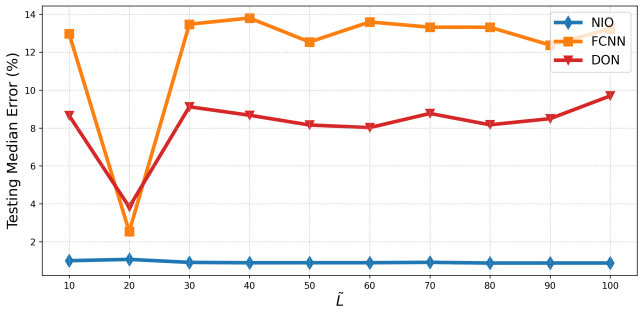


(f) Seismic Imaging StyleVel - A

Figure 23. Median of the L^1-error computed over testing samples $(\{\Psi_k\}_{k=1}^{\widetilde{L}}, a)$ VS $\widetilde{L}$ for different benchmarks with different models ($\widetilde{L} < L$).



(a) Calderón Problem Trigonometric



(b) Inverse Wave Scattering

Figure 24. Median of the L^1-error computed over testing samples $(\{\Psi_k\}_{k=1}^{\widetilde{L}}, a)$ VS $\widetilde{L}$ for different benchmarks with different models ($\widetilde{L}$ spans the entire range 10-100).

	DONet		FCNN		NIO	
	$L^1 \downarrow$	$L^2 \downarrow$	$L^1 \downarrow$	$L^2 \downarrow$	$L^1 \downarrow$	$L^2 \downarrow$
Calderón Problem Trigonometric	1.3% 3.19%	1.58% 3.75%	1.08% 2.24%	1.33% 2.7%	0.56% 1.21%	0.7% 1.47%
Calderón Problem Heart&Lungs	0.87% 1.42%	3.32% 4.11%	0.25% 0.44%	1.37% 3.5%	0.16% 0.2%	0.88% 1.53%
Inverse Wave Scattering	2.28% 5.55%	4.37% 11.0%	0.84% 4.98%	2.08% 13.47%	0.3% 2.15%	1.08% 5.92%
Radiative transport	1.56% 3.38%	2.72% 6.5%	1.0% 2.26%	2.02% 6.0%	0.74% 1.63%	1.82% 4.86%
Seismic Imaging CurveVel - A	3.25% 4.93%	4.79% 7.4%	2.01% 3.53%	3.98% 6.52%	2.16% 3.36%	3.83% 5.81%
Seismic Imaging Style - A	2.97% 4.88%	4.02% 6.61%	2.39% 4.1%	3.49% 6.09%	2.4% 3.83%	3.4% 5.6%

Table 7. 0.25 and 0.75 quantiles of the relative L^1-error and L^2-error computed over testing samples for different benchmarks and models.

	DONet	FCNN	NIO
Calderón Problem Trigonometric	2.02%	1.51%	0.91%
Calderón Problem Heart&Lungs	0.95%	0.27%	0.18%
Inverse Wave Scattering	3.83%	2.54%	3.72%
Radiative transport	2.38%	1.47%	1.1%
Seismic Imaging CurveVel - A	3.98%	2.65%	2.73%
Seismic Imaging Style - A	3.82%	3.13%	3.09%

Table 8. Median of the relative L^1-error computed over 1%-noisy testing samples for different benchmarks with different models.

presented in Table 10.

The results show that the L^1-error increases only to 1.18% and 1.43%, compared to the original setup where the errors were 0.86% and 1.11%, respectively. These findings underscore the NIO model's ability to maintain robust performance even when the boundary sensors are placed at different locations.

E.6. Out-of-Distribution Reconstruction.

In addition to *in-distribution* testing, we also consider an *out-of-distribution* testing task. This will enable us to evaluate the ability of the models to *generalize* to inputs (and outputs) that possess different features from the training ones.

First, we considered the Calderón Problem (Trigonometric) benchmark. The coefficients in the training distribution were sampled from the exponential of a sum of sines, with up to 4 frequency modes (up to 8π). We now test with the following distributions:

- Distribution A: $a(x, y) = \exp\left(\sum_{k=1}^{6} c_k \sin(k\pi x)\sin(k\pi y)/k^{\frac{3}{2}}\right)$
- Distribution B: $a(x, y) = \exp\left(\sum_{k=1}^{6} c_k \sin(k\pi x)\sin(k\pi y)/k\right)$

Here, c_k is a uniformly distributed random variable in the range $[0, 1]^m$. The coefficients include up to 6 frequency modes, with different decays of the higher order modes (1.5 and 1). Thus, at test time, the model now has to infer data with

	DONet		FCNN		NIO	
Resolution	50×50	100×100	50×50	100×100	50×50	100×100
Inverse Wave Scattering	3.74%	3.63%	1.81%	1.66%	0.93%	0.95%

Table 9. Relative median L^1-error computed over testing samples generated at different resolutions (grid sizes).

	DONet	FCNN	NIO
	$L^1 \downarrow$	$L^1 \downarrow$	$L^1 \downarrow$
Calderón Problem Trigonometric	3.39%	1.88%	1.18%
Inverse Wave Scattering	3.84%	2.53%	1.43%

Table 10. Relative median L^1-error computed over testing samples with random location of the boundary measurements with different models and different benchmarks.

significantly higher frequencies than the training data.

Next, we consider the Calderon problem of the Heart and Lungs. Here, the training distribution was based on a normally distributed perturbation of the Heart and Lungs Phantom, with the amplitude of the perturbation being at most 8% of the Phantom values (See **SM** 4). At test time, we now consider perturbations with amplitudes being 12% of the Phantom values, thus sampling from a different distribution. For instance, this higher amplitude could model individuals with some diseases. Note that even higher variations are probably unrealistic as this problem models body organs and one has to restrict to some biological constraints.

As a third benchmark, we consider the inverse scattering problem. Here, the training distribution was of a coefficient that consisted of between 1-4 scatterers, *of identical shape*, whose locations were randomly chosen (See **SM** 5). We chose to test this model now on two different test distributions

- Distribution A: $a(x, y) = \sum_{k=1}^{5} \exp\big(- c(x - c_{1,k})^4 - c(y - c_{2,k})^4\big)$. In this case, we use a family of coefficients with a fixed number of inclusions equal to five.
- Distribution B: $a(x, y) = \sum_{k=1}^{m} \exp\big(- b_k^4(x - c_{1,k})^4 - b_k^4(y - c_{2,k})^4\big)$, with $b_k \sim \mathcal{U}[5, 15]$, $m = \mod(\overline{m})$, $m \sim \mathcal{U}([1, 4])$ and $\{(c_{1,k}, c_{2,k})\} \sim \mathcal{U}([0, 1]^{m\times 2})$. This corresponds to a medium with one to four scatterers with varying shapes.

The relative median L^1 error for different models and different out-of-distribution testing is reported in Table 11. We observe that NIO generalizes well to unseen data, with test errors increasing at most by approximately a factor of 4, and still outperforms the baselines in all cases.

E.7. Ablation Studies.

We conduct two ablation studies focusing on two key elements of the Neural Inverse Operator. Firstly, we focus on the architecture shown in Figure 1, where we remove the nonlinear part $\mathcal{M}$ of FNO and set $d_v = 1$. In this case, we have:

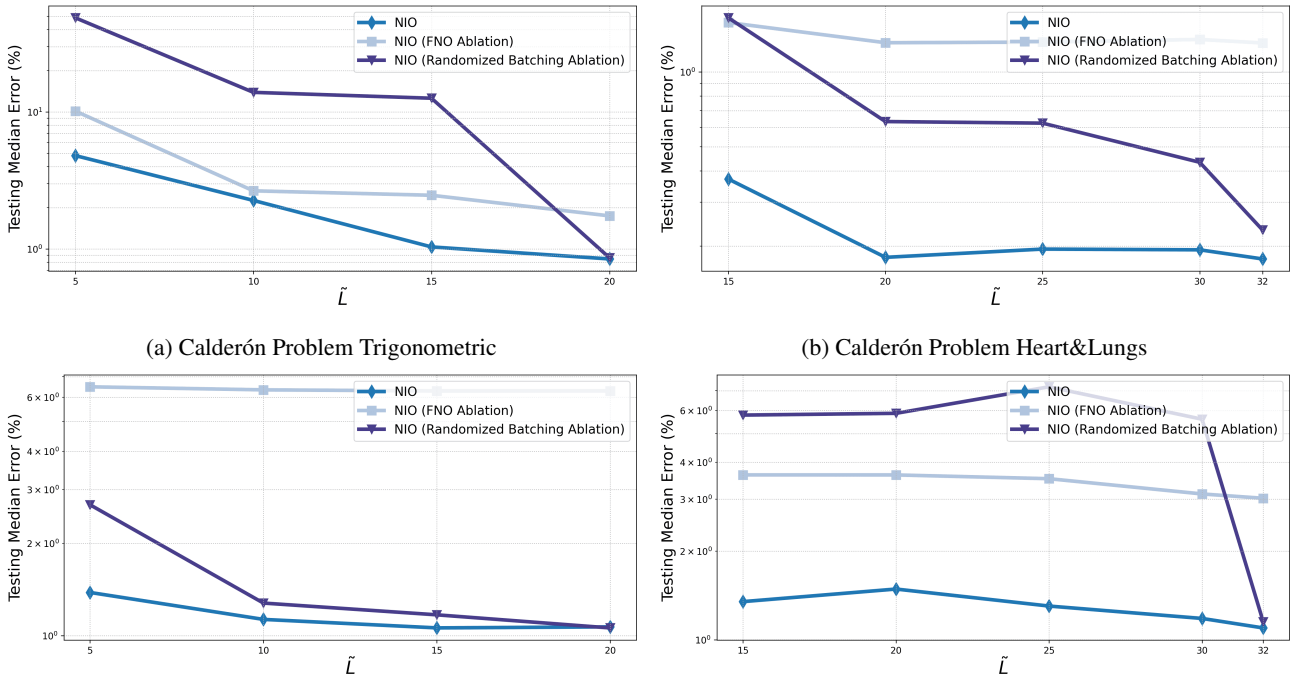
$$a^*(z) = R(f_1, \dots, f_L, z) = \frac{D}{L} \sum_{\ell=1}^{L} f_\ell + Ez, \tag{38}$$

where E and D are real-valued parameters. With these experiments, we aim to assess if the channels' mixing realized with $\mathcal{M}$ could improve the model's performance. Secondly, we examine the influence of randomized batching by training NIO without including the algorithm.

In order to maintain consistency, we use the same hyperparameter configurations for the ablation models as those of the best-performing NIO models (refer to Table 5 for the specific values).

	DONet	FCNN	NIO
	$L^1 \downarrow$	$L^1 \downarrow$	$L^1 \downarrow$
Calderón Problem Trigonometric Distribution A	1.37%	1.27%	1.2%
Calderón Problem Trigonometric Distribution B	1.62%	1.28%	0.91%
Calderón Problem Heart&Lungs	1.05%	0.28%	0.19%
Inverse Wave Scattering Distribution A	4.61%	3.84%	3.0%
Inverse Wave Scattering Distribution B	8.61%	8.98%	4.54%

Table 11. Relative median L^1-error computed over out-of-distribution test samples with different models and different benchmarks.



(a) Calderón Problem Trigonometric

(b) Calderón Problem Heart&Lungs

(c) Inverse Wave Scattering

(d) Radiative Transport

Figure 25. Median of the L^1-error computed over testing samples $(\{\Psi_k\}_{k=1}^{\widetilde{L}}, a)$, VS $\widetilde{L}$ for different benchmarks with different models (NIO and ablations).

The ablation study is being carried out for the Calderón problem with trigonometric function and heart&lungs Phantom, inverse wave scattering, and radiative transport. We consider the same experimental setup outlined in Section E.2 and report the corresponding results in Figure 25. The figure shows that the nonlinear term $\mathcal{M}$ and randomized batching significantly improve the model's performance. For the Calderón problem with trigonometric coefficients, removing $\mathcal{M}$ resulted in a nearly 2-times increase of the generalization error. On the other hand, for the remaining problems, the improvement is considerably more relevant, up to a factor of six for the inverse wave scattering.

It should be noted that in all experiments, only between $L = 20$ and $L = 32$ boundary measurements are used for training. In this scenario, the limited measurement data represents a bottleneck in accurately reconstructing the target coefficient. With a larger number, the nonlinear part may lead to an even greater reduction in the generalization error.

Regarding randomized batching, the improvements for the Calderón problem and radiative transport are substantial, ranging from 5 to 10 times compared to the ablated version. On the other hand, the improvements are more modest for the Inverse Wave Scattering, and the algorithm does not appear to considerably enhance generalization. However, these results are particularly impressive on account of the virtually zero cost associated with the randomized batching algorithm.

E.8. Comparison Standard Numerical Methods for Inverse Problems

In this section, we compare the performance of the proposed architecture, in terms of accuracy and inference time, with standard numerical methods, particularly PDE-constrained optimization techniques.

E.8.1. Calderón Problem Heart&Lungs

We begin by considering the Calderón problem for the discontinuous heart-lungs Phantom and conduct a comparative analysis between the proposed approach and the well-known D-bar method (Muller & Siltanen, 2012), which is commonly employed in the context of EIT. In Figure 4, we present the ground truth along with the NIO reconstruction for a randomly selected set of test samples. For comparison, we also include the reconstruction obtained using the D-bar method. As the figure shows, NIO 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 0.15% test error with NIO. 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 NIO is only 0.1 seconds (on CPU). 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 NIO, compared to traditional direct methods.

E.8.2. Inverse Wave Scattering

Next, we investigate the Inverse Wave Scattering problem and compare the NIO reconstruction with the results obtained through PDE-constrained optimization. We examine an out-of-distribution (Distribution A) test sample to accomplish this.

For PDE-constrained optimization, we employ a feed-forward neural network with trainable parameters θ that parameterizes the coefficient a. The neural network architecture follows the form specified in Equation 32, with L layers, d neurons, and activation function σ. The model parameters θ are initialized randomly, and the Helmholtz equation (9) is numerically solved for $L = 20$ realizations of the boundary conditions g_ℓ, where $\ell = 1, ..., L$ and the L^2-loss

$$J(\theta) = \sum_{\ell=1}^{L} ||\Psi_\ell - \widetilde{\Psi}_\ell||_2^2 \tag{39}$$

computed. Here, $\widetilde{\Psi}_\ell$, $\ell = 1, ..., L$ denotes the target data. The trainable network parameters are updated using LBFG in the direction of the gradient of $J(\theta)$. The process is iterated until convergence (1000 iterations, corresponding to 1000×20 PDE solves).

Observe that the hyperparameters of the network are chosen through a random search by picking the one minimizing $J(\theta)$.

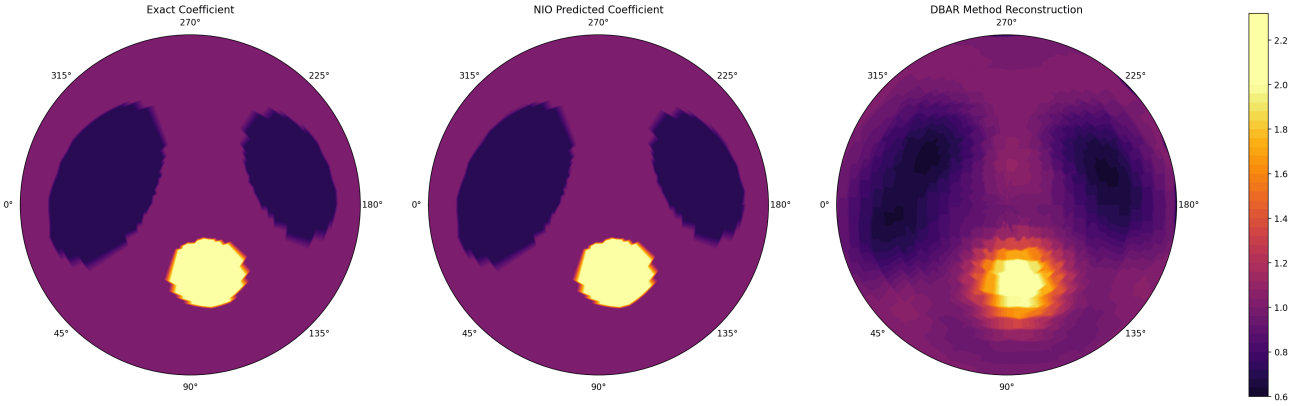
The reconstructed coefficient obtained through PDE-constrained optimization, along with the NIO prediction obtained by directly feeding the model with $\widetilde{\Psi}_\ell$ for $\ell = 1, \ldots, L$, are depicted in the right panel of Figure 2. The NIO reconstruction is significantly more accurate than the one obtained with PDE-constrained optimization, further reinforced by the corresponding L^1-error, equal to 2.3% and 11.1%, respectively. The corresponding total time required to reconstruct the coefficient, amounts to less than 1 second (on CPU) for NIO and 8.5 hours for the traditional method. It is worth noting that the finite difference (FD) solver employed for solving the equation is implemented on GPU within the PyTorch framework. By solving the PDE in parallel for $L = 20$ boundary measurements on 20 different GPUs, the computation time could potentially be reduced to 30 minutes, which is still three orders of magnitude slower than NIO's inference.

Finally, an alternative approach would be to replace the forward solver with a learning model instead of directly learning the inverse map. Although the surrogate model might achieve low errors in learning the forward map (compared to the inverse map), the inference time for the model could be approximately 0.1 seconds (on a CPU). If we require 1000 calls to the model for the optimization process, this will result in an overall inference time that is two orders of magnitude longer than NIO. Moreover, the reconstruction accuracy cannot surpass that obtained using the FD forward solver, which is already four times worse than the accuracy achieved with NIO.

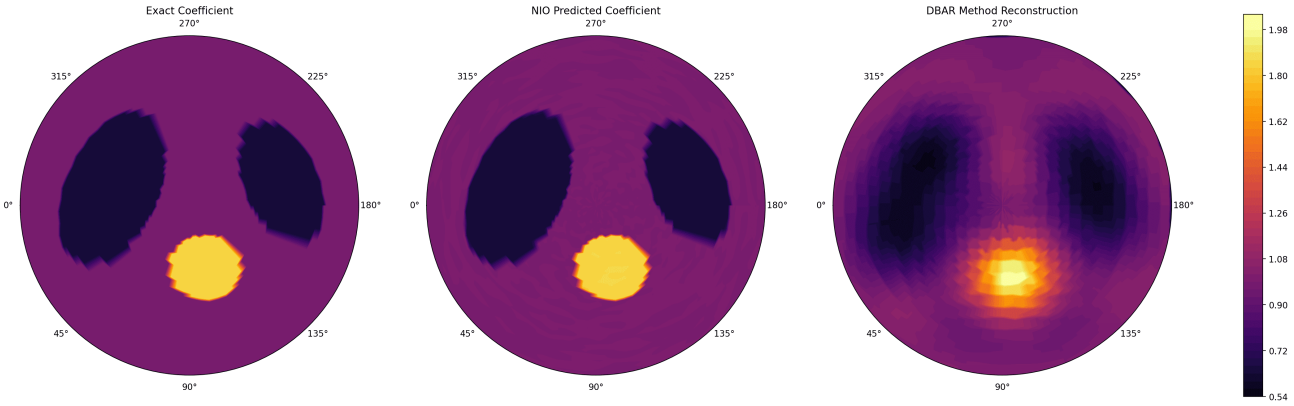
E.8.3. SEISMIC IMAGING

We also tested the PDE-constrained optimization method in the context of Seismic imaging with the CurveVel-A dataset. Unlike the NIO approach, in the PDE-constrained optimization approach, we first discretize the velocity coefficient $a(z)$ using a piecewise-constant parameterization, denoting the parameters as θ. We then solve the wave equation (14) using the finite difference method. The method is thus naturally dependent on the discretization of $a(z)$ and the PDE solver used to evaluate the source-to-receiver operator. We use the same experimental setup as provided in (Deng et al., 2021). A similar squared L^2 loss to (39) is used as the objective function to measure the mismatch between the observed data and the simulated receiver data based on the current prediction of $a(z)$. We also solve a corresponding adjoint wave equation whose solution is used to compute the gradient of the objective function $J(\theta)$ with respect to θ. We use the standard gradient descent method to perform the optimization. Each gradient evaluation requires two time-dependent PDE solves, which is the most expensive part of this PDE-constrained optimization approach.

The method leads to errors (in L^1) 2.5-4 times larger than CNIO for the same test sample while taking approximately 30 minutes of run-time on an 8-core M1-chip CPU. Figure 27 depicts the results for two test samples. The L^1-error achieved with NIO is 2.03% and 4.84% for samples 1 and 2, respectively, whereas the ones obtained with PDE-constrained optimization are 5.05% and 16.9%.

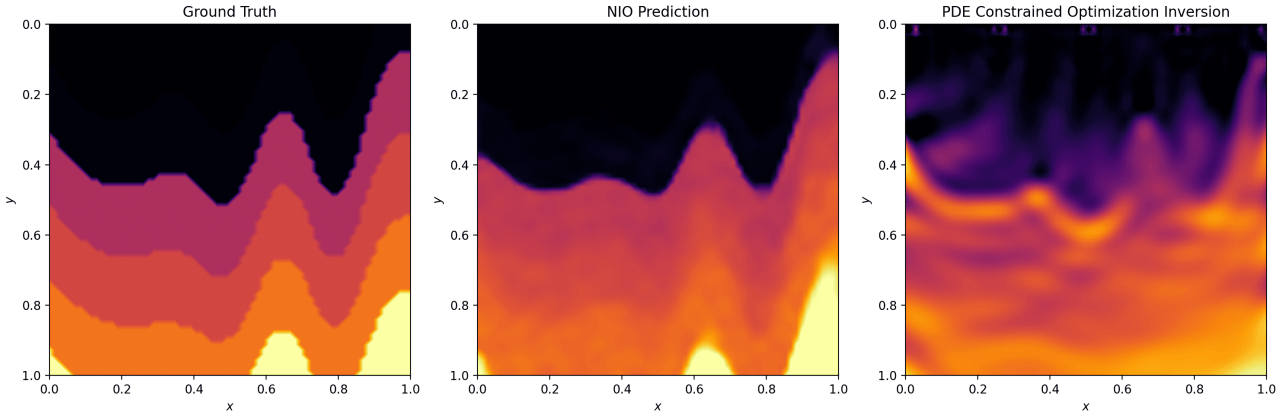


(a) Test Sample 1

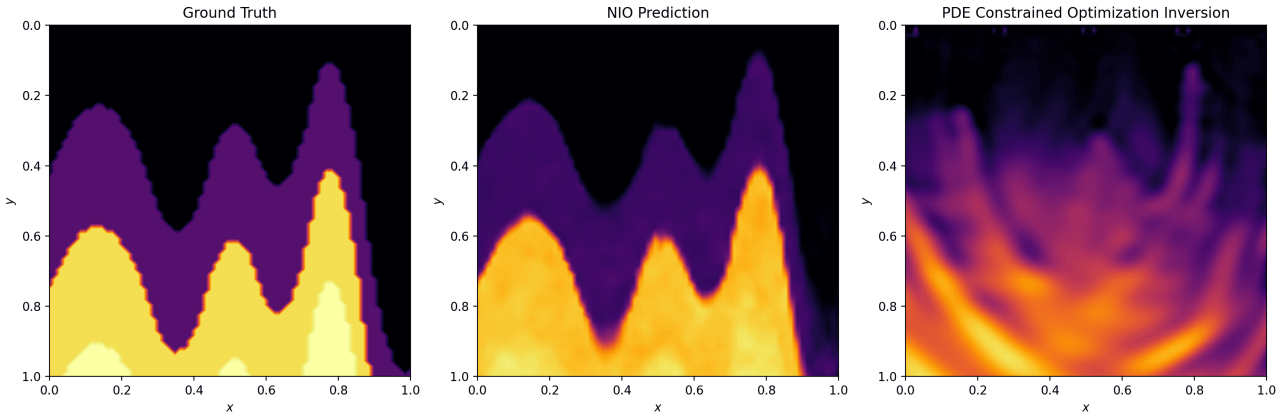


(b) Test Sample 2

Figure 26. Exact and predicted coefficients for two different test samples (Rows) for the Calderón problem with Trigononmetric - coefficients. Left Column: Ground Truth. Middle Column: NIO reconstruction. Right Column: Reconstruction with the D-bar Direct method of (Muller & Siltanen, 2012).



(a) Test Sample 1



(b) Test Sample 2

Figure 27. Exact and predicted coefficients for two different test samples (Rows) for the Curve Vel family. Left Column: Ground Truth. Middle Column: NIO reconstruction. Right Column: Reconstruction with the PDE-constrained optimization method.