Annals of Mathematical Sciences and Applications
Volume 7, Number 1, 23–48, 2022

Solving inverse wave scattering with deep learning

Yuwei Fan and Lexing Ying

This paper proposes a neural network approach for solving two classical problems in the two-dimensional inverse wave scattering: far field pattern problem and seismic imaging. The mathematical problem of inverse wave scattering is to recover the scatterer field of a medium based on the boundary measurement of the scattered wave from the medium, which is high-dimensional and nonlinear. For the far field pattern problem under the circular experimental setup, a perturbative analysis shows that the forward map can be approximated by a vectorized convolution operator in the angular direction. Motivated by this and filtered back-projection, we propose an effective neural network architecture for the inverse map using the recently introduced BCR-Net along with the standard convolution layers. Analogously for the seismic imaging problem, we propose a similar neural network architecture under the rectangular domain setup with a depth-dependent background velocity. Numerical results demonstrate the efficiency of the proposed neural networks.

AMS 2000 subject classifications: Primary 65N21, 74J25, 86A22.
Keywords and phrases: Inverse problems, deep learning, inverse scattering, seismic imaging.

1. Introduction

Inverse wave scattering is the problem of determining the intrinsic property of an object based on the data collected from the object scatters incoming waves under the illumination of an incident wave, which can be acoustic, electromagnetic, or elastic. In most cases, inverse wave scattering is non-intrusive to the object under study and therefore it has a wide range of applications including radar imaging [8], sonar imaging [32], seismic exploration [65], geophysics exploration [64], and medicine imaging [35] and so on.

arXiv: 1911.13202

Background. We focus on the time harmonic acoustic inverse scattering in two dimensions. Let Ω be a compact domain of interest. The inhomogeneous media scattering problem at a fixed frequency ω is modeled by the Helmholtz equation

$$Lu := \left(-\Delta - \frac{\omega^2}{c^2(x)} \right) u, \tag{1}$$

where $c(x)$ is the unknown velocity field. Assume that there exists a known background velocity $c_0(x)$ such that $c(x)$ is identical to $c_0(x)$ outside the domain Ω. By introducing the *scatterer* $\eta(x)$:

$$\eta(x) = \frac{\omega^2}{c(x)^2} - \frac{\omega^2}{c_0(x)^2} \tag{2}$$

compactly supported in Ω, one can equivalently work with $\eta(x)$ instead of $c(x)$. Note that in this definition $\eta(x)$ scales quadratically with the frequency ω. However, as ω is assumed to be fixed throughout this paper, this scaling does not affect our discussion.

In order to recover the unknown $\eta(\cdot)$, a typical setup of an experiment is as follows. For each *source* s from a source set S, one specifies an incoming wave (typically either a plane wave or a point source) and propagates the wave to the scatterer $\eta(\cdot)$. The scattered wave field $u^s(x)$ is then recorded at each *receiver* r from a receiver set R (typically placed at the domain boundary or infinity). The whole dataset, indexed by both the source s and the receiver r, is denoted by $\{d(s,r)\}_{s\in S, r\in R}$. The forward problem is to compute $d(s,r)$ given $\eta(x)$. The inverse scattering problem is to recover $\eta(x)$ given $d(s,r)$,

Both the forward and inverse problems are computationally quite challenging, and in the past several decades a lot of research has been devoted to their numerical solution [21, 22]. For the forward problem, the time-harmonic Helmholtz equation, especially in the high-frequency regime $\omega \gg 1$, is hard to solve mainly due to two reasons: (1) the Helmholtz operator has a large number of both positive and negative eigenvalues, with some close to zero; (2) a large number of degrees of freedom are required for discretization due to the Nyquist sampling rate. In recent years, quite a few methods have been developed for rapid solutions of Helmholtz operator [29, 18, 20, 19, 51]. The inverse problem is quite more difficult for numerical solution, due to the nonlinearity of the problem. For the methods based on optimization, as the loss landscape is highly non-convex (for example, the cycle skipping problem in seismic imaging [55]), the optimization can get stuck at a local minimum with rather large loss value. Other popular methods include the factorization method and the linear sampling method [44, 11, 9].

A deep learning approach. Deep learning (DL) has recently become the state-of-the-art approach in many areas of machine learning and artificial intelligence, including computer vision, image processing, and speech recognition [36, 46, 31, 54, 49, 62, 48, 61]. From a technical point of view, this success can be attributed to several key developments: neural networks (NNs) as a flexible framework for representing high-dimensional functions and maps, simple algorithms such as back-propagation (BP) and stochastic gradient descent (SGD) for tuning the model parameters, efficient general software packages such as Tensorflow and Pytorch, and unprecedented computing power provided by GPUs and TPUs.

More recently, deep neural networks (DNNs) have been increasingly used in scientific computing and computational engineering, particularly for PDE-related problems [41, 6, 33, 25, 3, 58, 47, 28]. One direction focuses on the low-dimensional parameterized PDE problems by representing the nonlinear map from the high-dimensional parameters of the PDE solution using DNNs [52, 34, 41, 25, 24, 23, 50, 4]. A second direction aims to use DNNs as an ansatz for high-dimensional PDEs [60, 10, 33, 42, 17] since DNNs offer a powerful tool for approximating high-dimensional functions and densities [15].

Related to the first direction mentioned above, DNNs have been widely applied to inverse problems [43, 37, 40, 2, 53, 63, 26, 27, 59]. For the forward problem, as applying neural networks to input data can be carried out rapidly due to novel software and hardware architectures, the forward solution can be significantly accelerated once the forward map is represented with a DNN. For the inverse problem, two critical computational issues are the choices of the solution algorithm and the regularization term. DNNs can help on both aspects: first, concerning the solution algorithm, due to its flexibility in representing high-dimensional functions, DNN can potentially be used to approximate the full inverse map, thus avoiding the iterative solution process; second, concerning the regularization term, DNNs often can automatically extract features from the data and offer a data-driven regularization prior.

This paper applies the deep learning approach to inverse wave scattering by representing the whole inverse map using neural networks. Two cases are considered here: (1) far field pattern and (2) seismic imaging. In our relatively simple setups, the main difference between the two is the source and receiver configurations: in the far field pattern, the sources are plane waves and the receivers are regarded as placed at infinity; in the seismic imaging, both the sources and receivers are placed at the top surface of the survey domain.

In each case, the starting point is a perturbative analysis of the forward map, which reveals that the forward map contains a vectorized one-dimensional convolution, after appropriate reparameterization of $\eta(x)$ and $d(s,r)$. This observation suggests to represent the forward map from η to d by a *one-dimensional* convolution neural network (with multiple channels). Following the idea of the filtered back-projection method [56], the inverse map can then be approximated by the adjoint map followed by a pseudo-differential filtering step. This suggests an architecture for the inverse map by reversing the forward map network followed by a simple two-dimensional convolution neural network. For the test problems being considered, the resulting neural networks have a relatively small number of parameters, thanks to the convolutional structure. This small number of parameters allows for more accurate and rapid training, even with a somewhat limited dataset.

Organization. This rest of the paper is organized as follows. Section 2 discusses the far field pattern problem. Section 3 considers the seismic imaging problems. Section 4 concludes some discussions for future work.

2. Far field pattern

2.1. Mathematics analysis

In the far field pattern case, the background velocity field $c_0(x)$ is constant, and without loss of generality, equal to one. We introduce the base operator $L_0 = -\Delta - \omega^2/c_0^2 = -\Delta - \omega^2$ and write L in a perturbative way as

$$L = L_0 - \eta. \tag{3}$$

The sources are parameterized by $s \in S = [0, 2\pi)$. For each source s, the incoming wave is a plane wave $e^{i\omega\widehat{s}\cdot x}$ with the unit direction given by $\widehat{s} = (\cos(s), \sin(s)) \in \mathbb{S}^1$. The scattered wave $u^s(x)$ satisfies the following equation

$$(L_0 - \eta)(e^{i\omega\widehat{s}\cdot x} + u^s(x)) = 0, \tag{4}$$

along with the Sommerfeld radiation boundary condition at infinity [14]. The receivers are also indexed by $r \in R = [0, 2\pi)$. The far field pattern at the unit direction $\widehat{r} = (\cos(r), \sin(r)) \in \mathbb{S}^1$ is defined as

$$\widehat{u}^s(r) \equiv \lim_{\rho\to\infty} \sqrt{\rho} e^{-i\omega\rho} u^s(\rho \cdot \widehat{r}).$$

The recorded data is the set of far field pattern from all incoming directions: $d(s, r) \equiv \widehat{u}^s(r)$ for $r \in R$ and $s \in S$.

In order to understand better the relationship between $\eta(x)$ and $d(s, r)$, we perform a perturbative analysis for small η. Expanding (4) leads to

$$(L_0 u^s)(x) = \eta(x) e^{i\omega\widehat{s}\cdot x} + \dots,$$

where $\dots$ stands for higher order terms in η. Letting $G_0 = L_0^{-1}$ be the Green's functions of the free-space Helmholtz operator L_0, we get

$$u^s(y) = \int G_0(y - x)\eta(x) e^{i\omega\widehat{s}\cdot x}\,\mathrm{d}x + \dots$$

Using the expansion at infinity

$$G_0(z) \approx \frac{1}{\sqrt{|z|}} \left(e^{i\omega|z|} + o(1)\right),$$

we arrive at

(5)

$$\begin{aligned}\widehat{u}^s(r) = \lim_{\rho\to\infty} \sqrt{\rho} e^{-i\omega\rho} u^s(\rho \cdot \widehat{r}) &\approx \lim_{\rho\to\infty} \int \frac{1}{\sqrt{\rho}} e^{i\omega(\rho - \widehat{r}\cdot x)} \sqrt{\rho} e^{-i\omega\rho} \eta(x) e^{i\omega\widehat{s}\cdot x}\,\mathrm{d}x \\ &= \int e^{-i\omega(\widehat{r}-\widehat{s})\cdot x} \eta(x)\,\mathrm{d}x \equiv d_1(s, r),\end{aligned}$$

where the notation $d_1(s, r)$ stands for the first order approximation to $d(s, r)$ in η.

2.1.1. Problem setup. For the far field pattern problem, we are free to treat the domain Ω as the unit disk centered at origin (by appropriate rescaling and translation), as illustrated in Fig. 1. In a common setting, the sources and receivers are uniformly sampled in $\mathbb{S}^1$, and $s = \frac{2\pi j}{N_s}$, $j = 0, \dots, N_s - 1$ and $r = \frac{2\pi k}{N_r}$, $k = 0, \dots, N_r - 1$, where $N_s = N_r$ in the current setup.

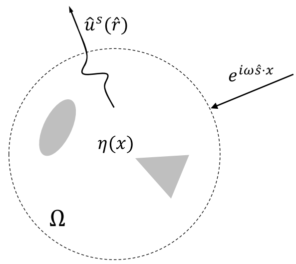


Figure 1: Illustration of the incoming and outgoing waves for a far field pattern problem. The scatterer $\eta(x)$ is compactly supported in the domain Ω. The incoming plane wave points at direction $\widehat{s} = (\cos(s), \sin(s))$. The far field pattern is sampled at each receiver direction $\widehat{r} = (\cos(r), \sin(r))$.

2.1.2. Forward map. Since the domain Ω is the unit disk, it is convenient to work with the problem in the polar coordinates. Let $x = (\rho\cos(\theta), \rho\sin(\theta))$, where $\rho \in [0, 1]$ is the radial coordinate and $\theta \in [0, 2\pi)$ is the angular one. Due to the circular tomography geometry that $r, s \in [0, 2\pi)$, it is convenient to reparameterize the measurement data by a change of variables

$$m = \frac{r+s}{2}, h = \frac{r-s}{2}, \quad r = m+h, s = m-h, \tag{6}$$

where all the variables m, h, r, s are understood modulus 2π. Figure 2 presents an example of the scatterer field $\eta(x)$ and the measurement data $d(s, r)$ in the original and transformed coordinates.

With a bit abuse of notation, we can redefine the measurement data

$$d(m, h) \equiv d(s, r)|_{s=m-h, r=m+h}, \tag{7}$$

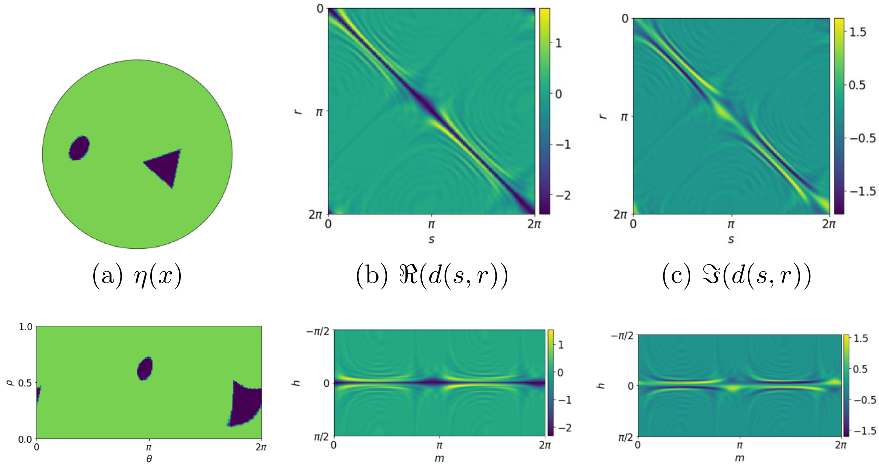
and so does $d_1(m, h)$. At the same time, we redefine

$$\eta(\theta, \rho) = \eta(\rho\cos(\theta), \rho\sin(\theta)$$

in the polar coordinates. Since the first order approximation $d_1(m, h)$ is linearly dependent on $\eta(\theta, \rho)$, there exists a kernel distribution $K(m, h, \theta, \rho)$ such that

$$d_1(m, h) = \int_0^1 \int_0^{2\pi} K(m, h, \theta, \rho)\eta(\theta, \rho)\, \mathrm{d}\rho\, \mathrm{d}\theta. \tag{8}$$

Since the domain is the unit disk centered at origin and the background velocity field $c_0 = 1$ is constant, the whole problem is equivariant to rotation.



(a) $\eta(x)$ (b) $\Re(d(s,r))$ (c) $\Im(d(s,r))$

(d) $\eta(\theta,\rho)$ (e) $\Re(d(m,h))$ (f) $\Im(d(m,h))$

Figure 2: Visualization of the scatterer field η and the measurement data d. The upper figures are the scatterer and the real and imaginary part of the measurement data d, respectively. The lower-left figure is the scatterer in the polar coordinates and the lower-right two figures are the real and imaginary part of the measurement data after change of variable.

In this case, the system can be dramatically simplified due to the following proposition.

Proposition 1. *There exists a function $\kappa(h,\rho,\cdot)$ periodic in the last parameter such that $K(m,h,\theta,\rho)=\kappa(h,\rho,m-\theta)$ or equivalently,*

$$d_1(m,h)=\int_0^1\int_0^{2\pi}\kappa(h,\rho,m-\theta)\eta(\rho,\theta)\,\mathrm{d}\theta\,\mathrm{d}\rho. \tag{9}$$

Proof. A simple calculation shows that the phase $(\widehat{r}-\widehat{s})\cdot x$ now becomes

$$(\widehat{r}-\widehat{s})\cdot x=(\widehat{(m+h)}-\widehat{(m-h)})\cdot(\rho\cos(\theta),\rho\sin(\theta))=2\rho\sin(h)\sin(\theta-m).$$

Therefore, (5) turns to

$$d_1(m,h)=\int_0^1\int_0^{2\pi}\left(\rho e^{2i\rho\omega\sin h\sin(m-\theta)}\right)\eta(\theta,\rho)\,\mathrm{d}\theta\,\mathrm{d}\rho.$$

By introducing $\kappa(h,\rho,y)=\rho e^{2i\rho\omega\sin h\sin(y)}$, we complete the proof. $\square$

Proposition 1 shows that K acts on η in the angular direction by a convolution, which allows us to evaluate the map $\eta(\theta,\rho) \to d_1(m,h)$ by a family of 1D convolutions, parameterized ρ and h.

Discretization. Until now, the discussion is in continuous space. The numerical discretization and numerical method for the Helmholtz equation will be presented in the numerical section. With a slight abuse of notation, the same symbols will be used to denote the discretization version of the continuous data and kernels. Then the discretization version of Equations (5) and (9) takes the form

$$d(m,h) \approx \sum_{\rho,\theta} K(m,h,\theta,\rho)\eta(\theta,\rho) = \sum_{\rho} \left(\kappa(h,\rho,\cdot) * \eta(\cdot,\rho)\right)(m). \tag{10}$$

2.2. Neural network

Forward map. The perturbative analysis shows that, when η is sufficiently small, the forward map $\eta(\theta,\rho) \to d(m,h)$ can be approximated by (10). In terms of the NN architecture, for small η, the forward map (10) can be approximated by a one-dimensional (non-local) convolution layer on the angular direction by taking h and ρ as the channels. For larger η, this linear approximation is no longer accurate. In order to address the nonlinear case, we propose to increase the number of convolution layers and include nonlinear activations for the neural network of (10).

Algorithm 1 Neural network architecture for the forward map $\eta \to d$.

Require: c, $N_{\text{cnn}} \in \mathbb{N}^+$, $\eta \in \mathbb{R}^{N_\theta \times N_\rho}$
Ensure: $d \in \mathbb{R}^{N_s \times N_h}$
1: $\xi = \mathsf{Conv1d}[c, 1, \mathsf{id}](\eta)$ with ρ as the channel direction ▷ Resampling η to fit for BCR-Net
2: $\zeta = \mathsf{BCR\text{-}Net}[c, N_{\text{cnn}}](\xi)$ ▷ Use BCR-Net to implement the convolutional neural network.
3: $d = \mathsf{Conv1d}[N_h, 1, \mathsf{id}](\zeta)$ ▷ Reconstruct the result d from the output of BCR-Net.
4: return d

In the convolutional neural network, the h and ρ directions are taken as channels direction. The number of channels, denoted by c, is quite problem-dependent and will be discussed in the numerical section. In the angular direction, since the convolution between η and d is global, in order to represent global interactions the window size of the convolution layer w must satisfy the constraint

$$wN_{\text{cnn}} \geq N_\theta, \tag{11}$$

where N_{cnn} is the number of layers and N_θ is number of discretization points on the angular direction. A simple calculation shows that the number of parameters of the neural network is $O(wN_{\text{cnn}}c^2) \sim O(N_\theta c^2)$. The recently proposed BCR-Net [23] has been demonstrated to require fewer number of parameters and provide good efficiency for such interactions. Therefore, we replace the convolution layers with the BCR-Net in our architecture. The resulting neural network architecture for the forward map is summarized in Algorithm 1 with an estimate of $O(c^2 \log(N_\theta) N_{\text{cnn}})$ parameters. The components of Algorithm 1 are detailed below.

- $\xi = \mathsf{Conv1d}[c, w, \phi](\eta)$ mapping $\eta \in \mathbb{R}^{N_\theta \times N_\rho}$ to $\xi \in \mathbb{R}^{N_\theta \times c}$ is the one-dimensional convolution layer with window size w, channel number c, activation function ϕ and period padding on the first direction.
- BCR-Net is motivated by the data-sparse nonstandard wavelet representation of the linear operators [7]. It processes the information at different scale separately and each scale can be understood as a *local* convolutional neural network. The one-dimensional $\zeta = \mathsf{BCR\text{-}Net}[c, N_{\text{cnn}}](\xi)$ maps $\xi \in \mathbb{R}^{N_\theta \times c}$ to $\zeta \in \mathbb{R}^{N_\theta \times c}$ where the number of channels and layers in the local convolutional neural network in each scale are c and n_{cnn}, respectively. The readers are referred to [23] for more details on the BCR-Net.

Inverse map. As we have seen, if η is sufficiently small, the forward map can be approximated by $d \approx K\eta$, the operator notation of the discretization (10). Here η is a vector indexed by (θ, ρ), d is a vector indexed by (m, h), and K is a matrix with row indexed by (m, h) and column indexed by (θ, ρ).

The filtered back-projection method [56] suggests the following formula to recover η:

$$\eta \approx (K^\mathsf{T} K + \epsilon I)^{-1} K^\mathsf{T} d, \tag{12}$$

where ϵ is a regularization parameter. The first piece $K^\mathsf{T} d$ can also be written as a family of convolutions as well

$$(K^\mathsf{T} d)(\theta, \rho) = \sum_h (\kappa(h, \rho, \cdot) * d(\cdot, h))(\theta). \tag{13}$$

The application of K^T to d can be approximated with a neural network similar to the one for K in Algorithm 1, by reversing the order. The second piece $(K^\mathsf{T} K + \epsilon I)^{-1}$ is a pseudo-differential operator in the (θ, ρ) space and it is implemented with several two-dimensional convolutional layers for

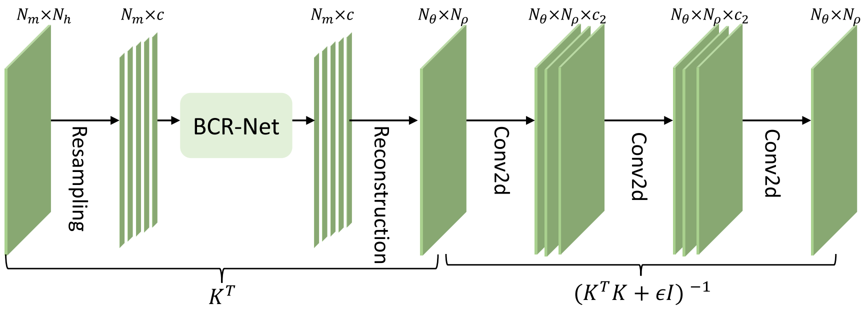


Figure 3: Neural network architecture for the inverse map of far field pattern problem.

simplicity. Putting two pieces together, the resulting architecture for the inverse map is summarized in Algorithm 2 and illustrated in Fig. 3. Here, $\mathsf{Conv2d}[c_2, w, \phi]$ used in Algorithm 2 is a two-dimensional convolution layer with window size w, channel number c_2, activation function ϕ and periodic padding on the first direction and zero padding on the second direction. The selection of the hyper-parameters in Algorithm 2 will be discussed in the numerical section.

Algorithm 2 Neural network architecture for the inverse problem $d \to \eta$.

Require: $c, c_2, w, N_{\text{cnn}}, N_{\text{cnn2}} \in \mathbb{N}^+$, $d \in \mathbb{R}^{N_s \times N_h}$
Ensure: $\eta \in \mathbb{R}^{N_\theta \times N_\rho}$
  # *Application of* K^T *to* d
1: $\zeta = \mathsf{Conv1d}[c, 1, \mathsf{id}](d)$ with h as the channel direction
2: $\xi = \mathsf{BCR\text{-}Net}[c, N_{\text{cnn}}](\zeta)$
3: $\xi^{(0)} = \mathsf{Conv1d}[N_\rho, 1, \mathsf{id}](\xi)$
  # *Application of* $(K^\mathsf{T}K + \epsilon I)^{-1}$
4: **for** k from 1 to $N_{\text{cnn2}} - 1$ **do**
5:   $\xi^{(k)} = \mathsf{Conv2d}[c_2, w, \mathsf{ReLU}](\xi^{(k-1)})$
6: **end for**
7: $\eta = \mathsf{Conv2d}[1, w, \mathsf{id}](\xi^{(N_{\text{cnn2}}-1)})$
8: return d

2.3. Numerical examples

This section report the numerical setup and results of the proposed neural network architecture in Algorithm 2 for the inverse map $d \to \eta$.

2.3.1. Experimental setup. Since the scatterer η is compactly supported in the unit disk Ω, we embed Ω into the square domain $[-1,1]^2$ and solve the Helmholtz equation (1) in the square. In the numerical solution of the Helmholtz equation, we discretize $[-1,1]^2$ with a uniform Cartesian mesh with 192 points (wave frequency is $\omega = 16$ and the number of points in each wavelength is 12) in each direction by a finite difference scheme. The perfectly matched layer [5] is used to deal with the Sommerfeld boundary condition and the solution of the discrete system can be accelerated with appropriate preconditioners (for example, [20]).

In the polar coordinates of Ω, $(\theta, \rho) \in [0, 2\pi) \times [0, 1]$ is partitioned by uniformly Cartesian mesh with 192×96 points, i.e., $N_\theta = 192$ and $N_\rho = 96$. Given the values of η in the Cartesian grid, the values $\eta(\theta, \rho)$ used in Algorithm 2 in the polar coordinates are computed via linear interpolation.

The number of sources and receivers are $N_s = N_r = 192$. The measurement data $d(s, r)$ is generated by solving the Helmholtz equation N_s times with different incident plane wave. For the change of variable of $(s, r) \to (m, h)$, linear interpolation is used to generate the data $d(m, h)$ from $d(s, r)$. In the (m, h) space, $N_m = 192$ for $m \in [0, 2\pi)$ and $N_h = 96$ for $h \in (-\pi/2, \pi/2)$. Since the measurement data is complex, the real and imaginary parts can be treated separately as two channels. In the actual simulation, numerical tests show that the real part is enough to generate good results. Hence, the following results, only the real part of $d(m, h)$ are used as the input of the neural network in Algorithm 2.

The NN in Algorithm 2 is implemented with Keras [12] running on top of TensorFlow [1]. All the parameters of the network are initialized by Xavier initialization [30]. The loss function is the mean squared error and the optimizer is the Nadam [16]. In the training process, the batch size and the learning rate is firstly set as 32 and 10^{-3} respectively, and the NN is trained 100 epochs. We then increase the batch size by a factor 2 till 512 with the learning rate unchanged, and then decreases the learning rate by a factor $10^{1/2}$ to 10^{-5} with the batch size fixed as 512. In each step, the NN is trained with 50 epochs. For the hyper-parameters used in Algorithm 2, $N_{\mathrm{cnn}} = 6$, $N_{\mathrm{cnn2}} = 5$, and $w = 3 \times 3$. The selection of the channel number c will be studied next.

2.3.2. Results. For a fixed η, $d(m, h)$ stands for the *exact* measurement data solved by numerical discretization of (1). The prediction of the NN from $d(m, h)$ is denoted by η^{NN}. The metric for the prediction is the peak

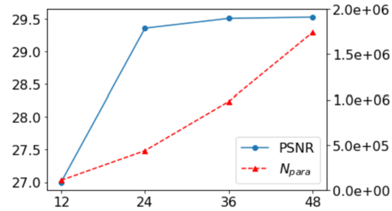


Figure 4: The test PSNR for different channel numbers c for the dataset $N_{\text{shape}} = 4$.

signal-to-noise ratio (PSNR), which is defined as

$$\text{PSNR} = 10 \log_{10}\left(\frac{\text{Max}^2}{\text{MSE}}\right) \tag{14}$$

where $\text{Max} = \max_{ij}(\eta_{ij}) - \min_{ij}(\eta_{ij})$ and $\text{MSE} = \frac{1}{N_\theta N_\rho}\sum_{i,j}|\eta_{i,j} - \eta_{i,j}^{\text{NN}}|^2$. For each experiment, the test PSNR is then obtained by averaging (14) over a given set of test samples. The numerical results presented below are obtained by repeating the training process five times, using different random seeds for the NN initialization.

The numerical experiments focus on the shape reconstruction setting [44, 45, 13], where η are often piecewise constant inclusions. Here, the scatterer field η is assumed to be the sum of N_{shape} piecewise constant shapes. For each shape, it can be either triangle, square or ellipse, its direction is uniformly random over the unit circle, its position is uniformly sampled in the disk, and its inradius is sampled from the uniform distribution $\mathcal{U}(0.1, 0.2)$. When a shape is an ellipse, the width and height are sampled from the uniform distribution $\mathcal{U}(0.1, 0.2)$ and $\mathcal{U}(0.05, 0.1)$. It is also required that each shape lies in the disk and there is no intersection between every two shapes. We generate two dataset for $N_{\text{shape}} = 2$ and $N_{\text{shape}} = 4$, and each has 20,480 samples $\{(\eta_i, d_i)\}$ with 16,384 used for training and the remaining 4,096 for testing.

We first study the choice of channel number c in Algorithm 2. Figure 4 presents the test PSNR and the number of parameters for different channel number c for the dataset $N_{\text{shape}} = 4$. As the channel number c increases, the test PSNR first increases consistently and then saturates. Note that the number of parameters of the neural network is $O(c^2 \log(N_\theta) N_{\text{cnn}})$. The choice of $c = 24$ offers a reasonable balance between accuracy and efficiency and the total number of parameters is 439K.

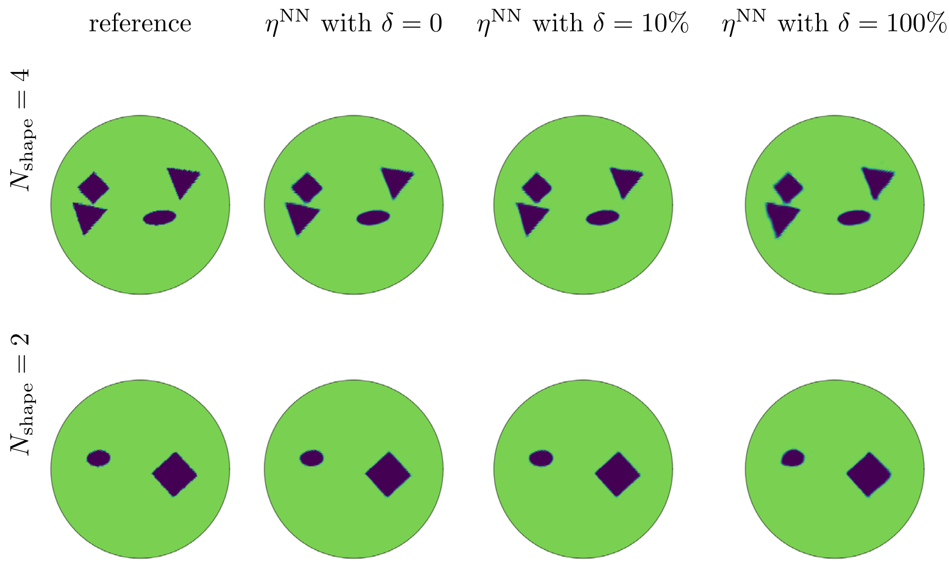


Figure 5: NN prediction for far field pattern of a sample in the test data for $N_{\mathrm{shape}} = 4$ (first row) or $N_{\mathrm{shape}} = 2$ (second row) for different noise level $\delta = 0$, 10% and 100%.

To model the uncertainty in the measurement data, we introduce noises to the measurement data by defining $d_i^\delta \equiv (1+Z_i\delta)d_i$, where Z_i is a Gaussian random variable with zero mean and unit variance and δ controls the signal-to-noise ratio. For each noisy level $\delta = 0$, 10%, 100%, an independent NN is trained and tested with the noisy dataset $\{(d_i^\delta, \eta_i)\}$.

Figure 5 collects, for different noise level $\delta = 0$, 10%, 100%, samples for different $N_{\mathrm{shape}} = 2$, 4. The NN is trained with the datasets generated in the same way as the test data. When there is no noise in the measurement data, the NN consistently gives accurate predictions of the scatterer field η, in the position, shape, and direction of the shapes. In particular, for the case $N_{\mathrm{shape}} = 4$, the square in the left part of the domain is close to a triangle. The NN is able to distinguish the shapes and gives a clear boundary of each. For the small noise levels, for example, $\delta = 10\%$, the boundary of the shapes slightly blurs while the position, direction and shape are still correct. As the noise level δ increases, the boundary of the shapes blurs more, but the position and direction of shape are always correct.

The next test is about the generalization of the proposed NN. We first train the NN with one data set ($N_{\mathrm{shape}} = 2$ or 4) with noise level $\delta = 0$, 10% or 100% and test with the other ($N_{\mathrm{shape}} = 4$ or 2) with the same noise level.

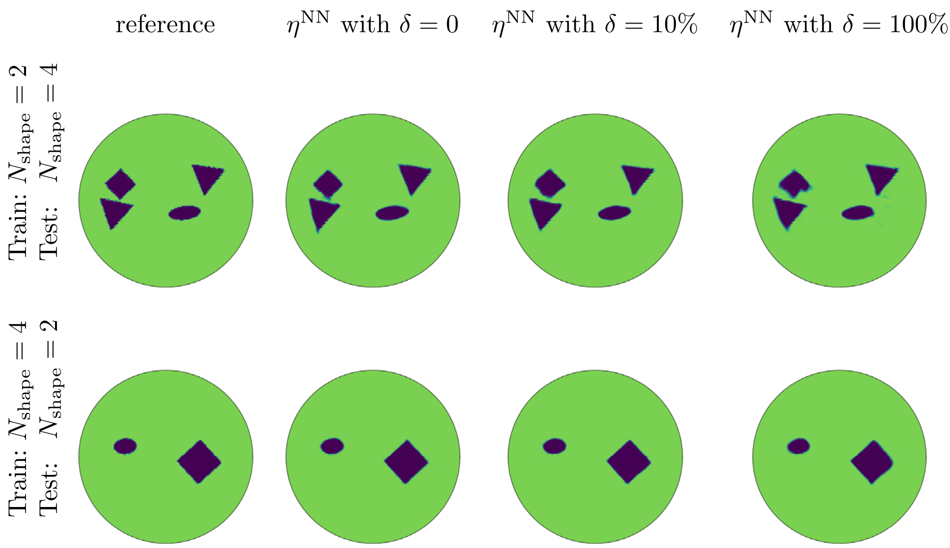


Figure 6: NN generalization test for far field pattern problem. The upper (or lower) figures: the NN is trained by the data of the number of shapes $N_{\mathrm{shape}} = 4$ (or 2) with noise level $\delta = 0$, 10% or 100% and test by the data of $N_{\mathrm{shape}} = 2$ (or 4) with the same noise level.

The results, presented in Fig. 6, indicate that the NN trained by the data with two inclusions is capable of recovering the measurement data of the case with four inclusions, and vice versa. Moreover, the prediction results are comparable with those in Fig. 5. This shows that the trained NN is capable of predicting beyond the training scenario.

3. Seismic imaging

3.1. Mathematics analysis

In the seismic imaging case, Ω is a rectangular domain with Sommerfeld radiation boundary condition specified, as illustrated in Fig. 7. Following [38, 57], we apply periodic boundary condition in the horizontal direction to our problem for simplicity. This setup is also appropriate for studying periodic material, such as phononic crystals [57, 39], etc. After appropriate rescaling, we consider the domain $\Omega = [0,1] \times [0, Z]$, where Z is a fixed constant. Both the sources $S = \{x_s\}$ and the receivers $R = \{x_r\}$ are a set of uniformly sampled points along a horizontal line near the top surface of the domain, and $x_r = (r, Z)$ and $x_s = (s, Z)$, for $r, s \in [0,1]$.

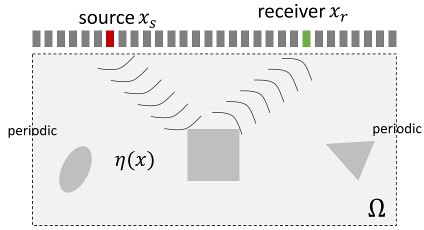


Figure 7: Illustration of a simple seismic imaging setting. The sources and receivers are located near the surface level (top) of the domain Ω. The scatterer field $\eta(x)$ is assumed to be well-separated from the sources and the receivers.

Using the background velocity field $c_0(x)$, we first introduce the background Helmholtz operator $L_0 = -\Delta - \omega^2/c_0(x)^2$. For each source s, we place a delta source at point x_s and solve the the Helmholtz equation (in the differential imaging setting)

$$(L_0 - \eta)(G_0(x, x_s) + u^s(x)) = 0, \tag{15}$$

where $G_0 = L_0^{-1}$ be the Green's functions of the background Helmholtz operator L_0. The solution is recorded at points x_r for $r \in R$ and the whole dataset is $d(s,r) \equiv u^s(x_r)$. In order to understand better the relationship between $\eta(x)$ and $d(s,r)$, let us perform a perturbative analysis for small η. Expanding (15) gives rise to

$$(L_0 u^s)(x) = \eta(x) G_0(x, x_s) + \dots .$$

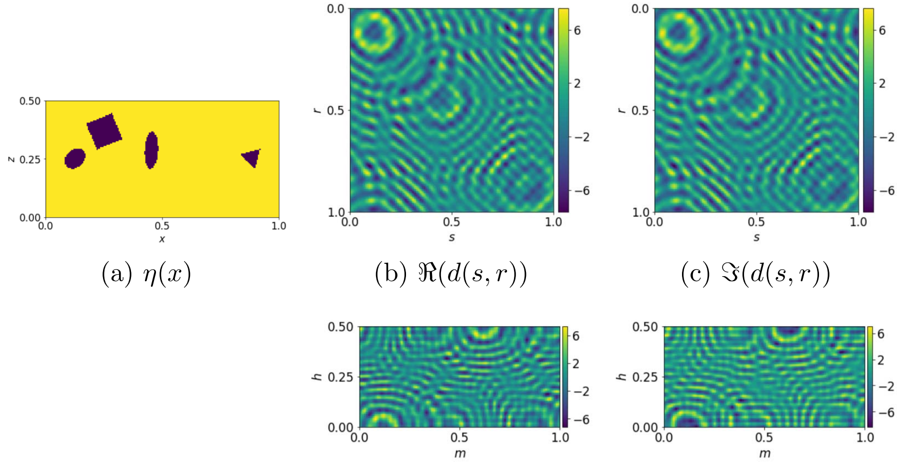
Solving this leads to

$$d(s,r) = u^s(r) = \int G_0(x_r, x) G_0(x, x_s) \eta(x) \,\mathrm{d}x + \dots .$$

Again, we introduce $d_1(s,r) = \int G_0(x_r, x) G_0(x, x_s) \eta(x) \,\mathrm{d}x$ as the leading order linear term in η.

Figure 8 gives an example of the scatterer field and the measurement data. Notice that the strongest signal concentrates at the diagonal of the measurement data $d(s,r)$. Because of the periodicity in the horizontal direction, it is convenient to rotate the measurement data by a change of variables as

$$m = \frac{r+s}{2}, h = \frac{r-s}{2}, \quad \text{or equivalently} \quad r = m + h, s = m - h, \tag{16}$$



(a) $\eta(x)$ (b) $\Re(d(s,r))$ (c) $\Im(d(s,r))$

(d) $\Re(d(m,h))$ (e) $\Im(d(m,h))$

Figure 8: Visualization of the scatterer field η and the measurement data d for the seismic imaging. The upper figures are the scatterer and the real and imaginary part of the measurement data d, respectively. The lower two figures are the real and imaginary part of the measurement data after change of variable.

where all the variables m, h, r, s are understood modulus 1. With a bit abuse of notation, we recast the measurement data

$$d(m,h) \equiv d(s,r)|_{s=m-h, r=m+h}, \tag{17}$$

and so does for $d_1(m,h)$. At the same time, by letting $x = (p,z)$ where p is horizontal component of x and z is the depth component, we write $\eta(p,z) = \eta(x)$. Since $d_1(m,h)$ is linearly dependent on $\eta(p,z)$, there exists a kernel distribution $K(m,h,p,z)$ such that

$$d_1(m,h) = \int_0^Z \int_0^1 K(m,h,p,z)\eta(p,z)\, \mathrm{d}z\, \mathrm{d}p. \tag{18}$$

One of the most common scenario in seismic imaging is that $c_0(x)$ only depends on the depth, i.e., $c_0(p,z) \equiv c_0(z)$. Note that in this scenario the whole problem is equivariant to translation in the horizontal direction. The system can be dramatically simplified due to the following proposition.

Proposition 2. *There exists a function $\kappa(h, z, \cdot)$ periodic in the last parameter such that $K(m, h, p, z) = \kappa(h, z, m-p)$ or equivalently,*

$$d_1(m,h) = \int_0^Z \int_0^1 \kappa(h,z,m-p)\eta(p,z)\,\mathrm{d}p\,\mathrm{d}z. \tag{19}$$

Proof. Because of $c_0(p,z) = c_0(z)$ and the periodic boundary condition in the horizontal direction, the Green's function of the background Helmholtz operator G_0 is translation invariant on the horizontal direction, i.e., there exists a $g_0(\cdot,\cdot,\cdot)$ such that $G((x_1,x_2),(y_1,y_2)) = g(x_1-y_1, x_2, y_2)$. Therefore,

$$\begin{aligned} d_1(m,h) &= \int_0^Z \int_0^1 G_0((m+h,Z),(p,z))G_0((p,z),(m-h,Z))\eta(p,z)\,\mathrm{d}z\,\mathrm{d}p \\ &= \int_0^Z \int_0^1 g_0(m-p+h,Z,z)g_0(p-m+h,z,Z)\eta(p,z)\,\mathrm{d}z\,\mathrm{d}p. \end{aligned}$$

By introducing $\kappa(h,z,y) = g_0(y+h,Z,z)g_0(-y+h,z,Z)$, we complete the proof. $\square$

To discrete the problem, the scatterer $\eta(p,z)$ will be represented on a uniform mesh of $[0,1]\times[0,Z]$. With a slight abuse of notation, we shall use the same symbols to denote the discretization version of the continuous kernels and variables. The discrete version of (19) then becomes

$$d(m,h) \approx \sum_z \left(\kappa(h,z,\cdot) * \eta(\cdot,z)\right)(m). \tag{20}$$

3.2. Neural network and numerical examples

3.2.1. Neural network. Note that the key of the neural network architecture in Algorithm 2 for the far field pattern case is the convolution form in the angular direction in Proposition 1. For the seismic imaging case, Proposition 2 is the counterpart of Proposition 1. Since the argument in Section 2.2 remains valid for seismic imaging, the neural network architecture for seismic imaging is the same as that in Algorithm 2. However, the hyper-parameters are problem-dependent.

3.2.2. Experimental setup. In the experiment $Z = 1/2$ and the domain $\Omega = [0,1]\times[0,Z]$ is discretized with a uniform Cartesian mesh with 192×96 points with the wave frequency $\omega = 16$. The remaining setup of the numerical solution of the Helmholtz equation is same as that for the far field

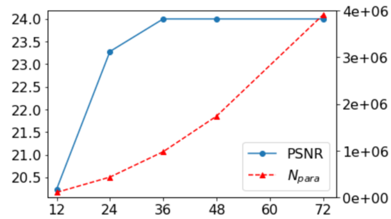
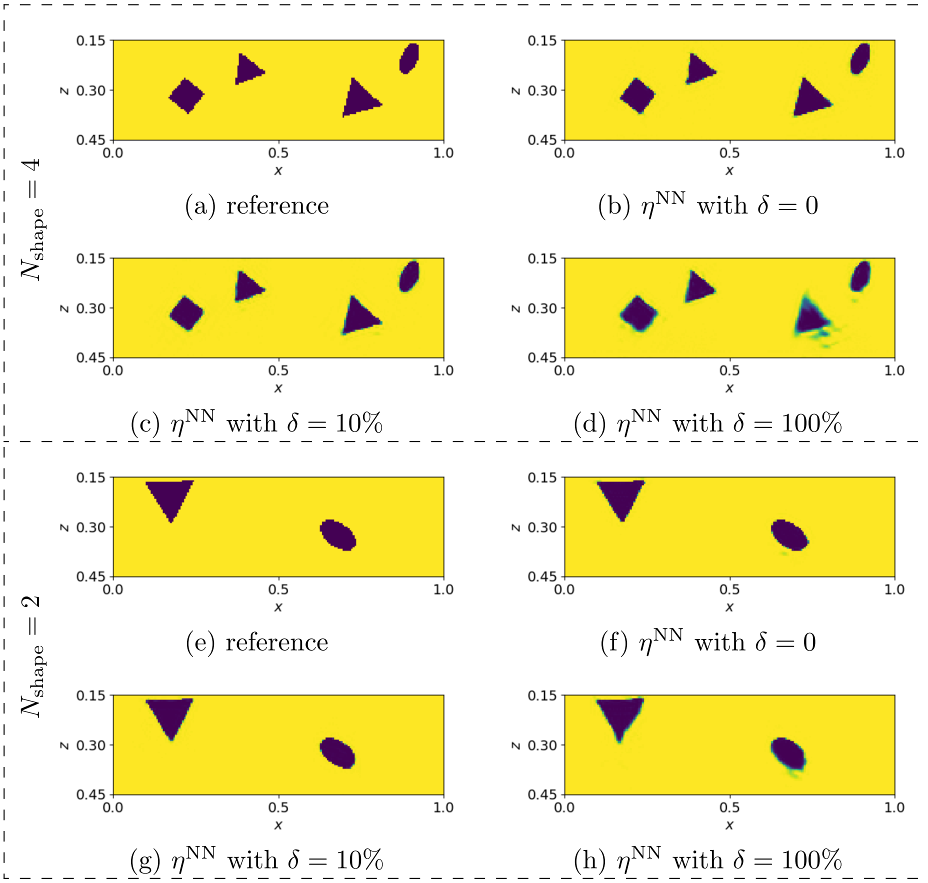


Figure 9: The test PSNR for seismic imaging for different channel numbers c with $N_{\text{shape}} = 4$.

pattern problem. For the measurement, we also set the number of sources and receivers as $N_s = N_r = 192$. The measurement data $d(s, r)$ is generated by solving the Helmholtz equation N_s times by placing a delta function on each source point. For the change of variable of $(s, r) \to (m, h)$, linear interpolation is used for generating the data $d(m, h)$ from $d(s, r)$, with $N_m = 192$ for $m \in [0, 1)$ and $N_h = 96$ for $h \in (0, 1/2)$. In the actual simulation, we use both the real and imaginary part and concentrate them on the h direction as the input.

3.2.3. Results. The numerical experiments here focus on the shape reconstruction setting, where η are piecewise constant inclusions. Here, the scatterer field η is assumed to be the sum of N_{shape} piecewise constant shapes. For each shape, it can be either triangle, square or ellipse, the orientation is uniformly random over the unit circle, the position is uniformly sampled in the $[0, 1] \times [0.2, 0.4]$, and the circumradius is sampled from the uniform distribution $\mathcal{U}(0.1, 0.2)$. If the shape is ellipse, its width and height are sampled from the uniform distribution $\mathcal{U}(0.08, 0.16)$ and $\mathcal{U}(0.04, 0.8)$. It is also required that there is no intersection between any two shapes. We generate two datasets with $N_{\text{shape}} = 2$ and $N_{\text{shape}} = 4$ and each has 20,480 samples $\{(\eta_i, d_i)\}$ with 16,384 used for training and the remaining 4,096 reserved for testing.

The first study is about the choice of the channel number c in Algorithm 2. Figure 9 presents the test PSNR and the number of parameters, for different channel number c on the dataset $N_{\text{shape}} = 4$. Similar to the far field pattern problem, as the channel number c increases, the test PSNR first consistently increases and then saturates. Notice that the number of parameters of the neural network is $O(c^2 \log(N_\theta) N_{\text{cnn}})$. The choice of $c = 36$ is a reasonable balance between accuracy and efficiency and the total number of parameters is 981K.



(a) reference (b) η^{NN} with $\delta = 0$ (c) η^{NN} with $\delta = 10\%$ (d) η^{NN} with $\delta = 100\%$

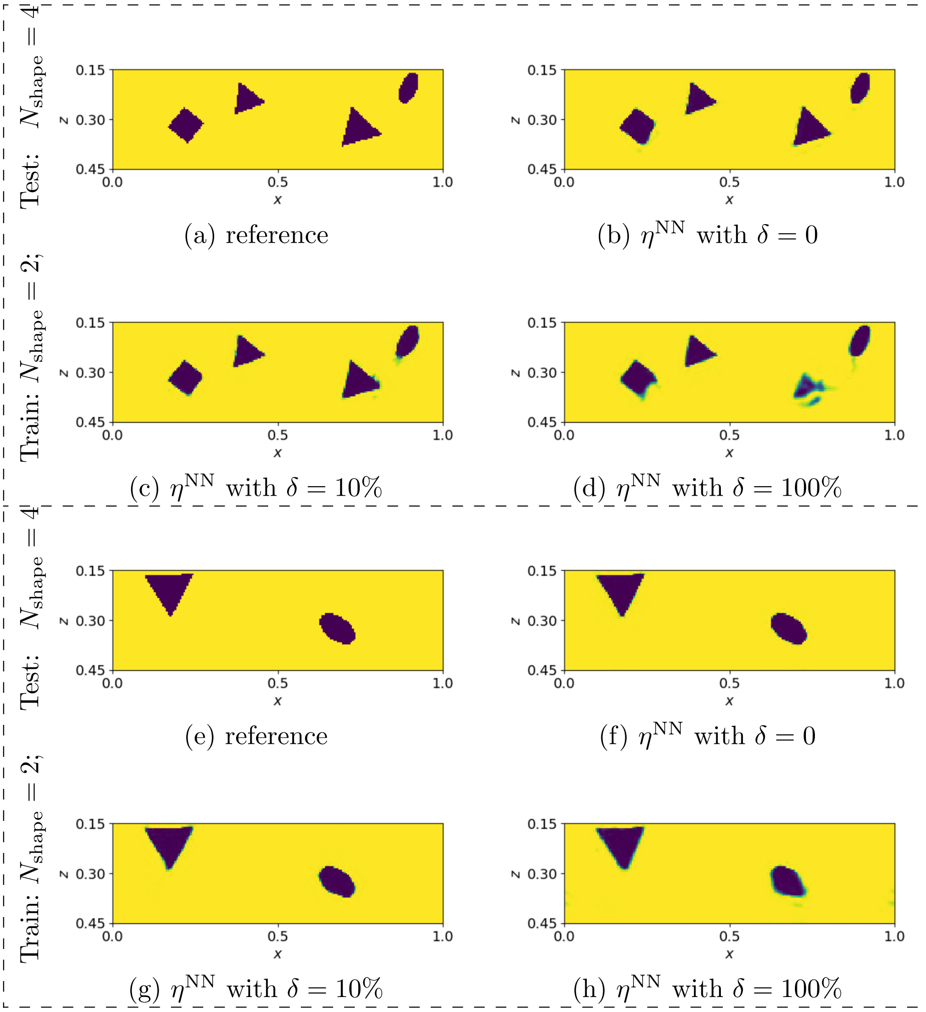
(e) reference (f) η^{NN} with $\delta = 0$ (g) η^{NN} with $\delta = 10\%$ (h) η^{NN} with $\delta = 100\%$

Figure 10: NN prediction for seismic imaging of a sample in the test data for $N_{\text{shape}} = 4$ (first two rows) or $N_{\text{shape}} = 2$ (last two rows) for different noise level $\delta = 0$, 10% and 100%.

To model the uncertainty in the measurement data, the same method as the far field pattern problem is used to add noises to the measurement data. Figure 10 collects, for different noise level $\delta = 0$, 10%, 100%, samples for $N_{\text{shape}} = 2$ and 4, and Fig. 11 presents the generalization test of the proposed NN by training and testing on different datasets..

4. Discussions

This paper presents a neural network approach for the two typical problems of the inverse scattering: far field pattern and seismic imaging. The



(a) reference (b) η^{NN} with $\delta = 0$

(c) η^{NN} with $\delta = 10\%$ (d) η^{NN} with $\delta = 100\%$

(e) reference (f) η^{NN} with $\delta = 0$

(g) η^{NN} with $\delta = 10\%$ (h) η^{NN} with $\delta = 100\%$

Figure 11: NN generalization test for seismic imaging. The upper (or lower) figures: the NN is trained by the data of the number of shapes $N_{\text{shape}} = 4$ (or 2) with noise level $\delta = 0$, 10% or 100% and test by the data of $N_{\text{shape}} = 2$ (or 4) with the same noise level.

approach uses the NN to approximate the whole inverse map from the measurement data to the scatterer field, inspired by the perturbative analysis that indicates that the linearized forward map can be represented by a one-dimensional convolution with multiple channels. The analysis in this paper can also be extended to the three-dimensional scattering problems. The analysis of seismic imaging can be easily extended to non-periodic boundary conditions by replacing the periodic padding in Algorithm 2 with zero padding.

Acknowledgements

The work of Y.F. and L.Y. is partially supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Scientific Discovery through Advanced Computing (SciDAC) program. The work of L.Y. is also partially supported by the National Science Foundation under award DMS-1818449.

References

[1] M. Abadi et al. Tensorflow: A system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016.

[2] J. Adler and O. Öktem. Solving ill-posed inverse problems using iterative deep neural networks. *Inverse Problems*, 33(12):124007, 2017. MR3729789

[3] M. Araya-Polo, J. Jennings, A. Adler, and T. Dahlke. Deep-learning tomography. *The Leading Edge*, 37(1):58–66, 2018.

[4] L. Bar and N. Sochen. Unsupervised deep learning algorithm for PDE-based forward and inverse problems. *arXiv preprint arXiv:1904.05417*, 2019.

[5] J.-P. Berenger. Perfectly matched layer for the fdtd solution of wave-structure interaction problems. *IEEE Transactions on Antennas and Propagation*, 44(1):110–117, 1996.

[6] J. Berg and K. Nyström. A unified deep artificial neural network approach to partial differential equations in complex geometries. *Neurocomputing*, 317:28–41, 2018. MR3796894

[7] G. Beylkin, R. Coifman, and V. Rokhlin. Fast wavelet transforms and numerical algorithms I. *Communications on Pure and Applied Mathematics*, 44(2):141–183, 1991. MR1085827

[8] B. Borden. Mathematical problems in radar inverse scattering. *Inverse Problems*, 18(1):R1, 2001. MR1893582

[9] F. Cakoni, D. Colton, and P. Monk. *The linear sampling method in inverse electromagnetic scattering*, volume 80. SIAM, 2011. MR2731610

[10] G. Carleo and M. Troyer. Solving the quantum many-body problem with artificial neural networks. *Science*, 355(6325):602–606, 2017. MR3642415

[11] M. Cheney. The linear sampling method and the music algorithm. *Inverse problems*, 17(4):591, 2001. MR1861470

[12] F. Chollet et al. Keras. https://keras.io, 2015.

[13] D. Colton and R. Kress. Looking back on inverse scattering theory. *SIAM Review*, 60(4):779–807, 2018. MR3873017

[14] D. L. Colton, R. Kress, and R. Kress. *Inverse acoustic and electromagnetic scattering theory*, volume 93. Springer, 1998. MR2986407

[15] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989. MR1015670

[16] T. Dozat. Incorporating Nesterov momentum into adam. *International Conference on Learning Representations*, 2016.

[17] W. E and B. Yu. The deep Ritz method: A deep learning-based numerical algorithm for solving variational problems. *Communications in Mathematics and Statistics*, 6(1):1–12, 2018. MR3767958

[18] B. Engquist and L. Ying. Fast directional algorithms for the helmholtz kernel. *Journal of Computational and Applied Mathematics*, 234(6):1851–1859, 2010. MR2644180

[19] B. Engquist and L. Ying. Sweeping preconditioner for the helmholtz equation: hierarchical matrix representation. *Communications on Pure and Applied Mathematics*, 64(5):697–735, 2011. MR2789492

[20] B. Engquist and L. Ying. Sweeping preconditioner for the helmholtz equation: moving perfectly matched layers. *Multiscale Modeling & Simulation*, 9(2):686–710, 2011. MR2818416

[21] Y. A. Erlangga. Advances in iterative methods and preconditioners for the helmholtz equation. *Archives of Computational Methods in Engineering*, 15(1):37–66, 2008. MR2387515

[22] O. G. Ernst and M. J. Gander. Why it is difficult to solve helmholtz problems with classical iterative methods. In *Numerical analysis of multiscale problems*, pages 325–363. Springer, 2012. MR3050918

[23] Y. Fan, C. O. Bohorquez, and L. Ying. BCR-Net: a neural network

based on the nonstandard wavelet form. *Journal of Computational Physics*, 384:1–15, 2019. MR3915332

[24] Y. Fan, J. Feliu-Fabà, L. Lin, L. Ying, and L. Zepeda-Núñez. A multiscale neural network based on hierarchical nested bases. *Research in the Mathematical Sciences*, 6(2):21, 2019. MR3921459

[25] Y. Fan, L. Lin, L. Ying, and L. Zepeda-Núñez. A multiscale neural network based on hierarchical matrices. *arXiv preprint arXiv:1807.01883*, 2018. MR4033684

[26] Y. Fan and L. Ying. Solving electrical impedance tomography with deep learning. *arXiv preprint arXiv:1906.03944*, 2019. MR4044725

[27] Y. Fan and L. Ying. Solving optical tomography with deep learning. *arXiv preprint arXiv:1910.04756*, 2019. MR4044725

[28] J. Feliu-Faba, Y. Fan, and L. Ying. Meta-learning pseudo-differential operators with deep neural networks. *arXiv preprint arXiv:1906.06782*, 2019. MR4065434

[29] M. J. Gander and F. Nataf. Ailu for helmholtz problems: a new preconditioner based on the analytic parabolic factorization. *Journal of Computational Acoustics*, 9(04):1499–1506, 2001. MR1881800

[30] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.

[31] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016. MR3617773

[32] C. Greene, P. Wiebe, J. Burczynski, and M. Youngbluth. Acoustical detection of high-density krill demersal layers in the submarine canyons off georges bank. *Science*, 241(4863):359–361, 1988.

[33] J. Han, A. Jentzen, and W. E. Solving high-dimensional partial differential equations using deep learning. *Proceedings of the National Academy of Sciences*, 115(34):8505–8510, 2018. MR3847747

[34] J. Han, L. Zhang, R. Car, and W. E. Deep potential: A general representation of a many-body potential energy surface. *Communications in Computational Physics*, 23(3):629–639, 2018. MR3869647

[35] T. Henriksson, N. Joachimowicz, C. Conessa, and J.-C. Bolomey. Quantitative microwave imaging for breast cancer detection using a planar

2.45 ghz system. *IEEE Transactions on Instrumentation and Measurement*, 59(10):2691–2699, 2010.

[36] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

[37] S. R. H. Hoole. Artificial neural networks in the solution of inverse electromagnetic field problems. *IEEE Transactions on Magnetics*, 29(2):1931–1934, 1993.

[38] T. Hulme, A. Haines, and J. Yu. General elastic wave scattering problems using an impedance operator approach-ii. two-dimensional isotropic validation and examples. *Geophysical Journal International*, 159(2):658–666, 2004.

[39] M. I. Hussein, M. J. Leamy, and M. Ruzzene. Dynamics of phononic materials and structures: Historical origins, recent progress, and future outlook. *Applied Mechanics Reviews*, 66(4):040802, 2014.

[40] H. Kabir, Y. Wang, M. Yu, and Q.-J. Zhang. Neural network inverse modeling and applications to microwave filter design. *IEEE Transactions on Microwave Theory and Techniques*, 56(4):867–879, 2008.

[41] Y. Khoo, J. Lu, and L. Ying. Solving parametric PDE problems with artificial neural networks. *arXiv preprint arXiv:1707.03351*, 2017. MR4253972

[42] Y. Khoo, J. Lu, and L. Ying. Solving for high-dimensional committor functions using artificial neural networks. *Research in the Mathematical Sciences*, 6(1):1, 2019. MR3887217

[43] Y. Khoo and L. Ying. SwitchNet: a neural network model for forward and inverse scattering problems. *arXiv preprint arXiv:1810.09675*, 2018. MR4018415

[44] A. Kirsch. Factorization of the far-field operator for the inhomogeneous medium case and an application in inverse scattering theory. *Inverse Problems*, 15(2):413, 1999. MR1684466

[45] A. Kirsch and N. Grinberg. *The factorization method for inverse problems*, volume 36. Oxford University Press, 2008. MR2378253

[46] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *Proceedings of the 25th*

International Conference on Neural Information Processing Systems - Volume 1, NIPS'12, pages 1097–1105, USA, 2012. Curran Associates Inc.

[47] G. Kutyniok, P. Petersen, M. Raslan, and R. Schneider. A theoretical analysis of deep neural networks and parametric PDEs. *arXiv preprint arXiv:1904.00377*, 2019.

[48] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(436), 2015.

[49] M. K. K. Leung, H. Y. Xiong, L. J. Lee, and B. J. Frey. Deep learning of the tissue-regulated splicing code. *Bioinformatics*, 30(12):i121–i129, 2014.

[50] Y. Li, J. Lu, and A. Mao. Variational training of neural network approximations of solution maps for physical models. *arXiv preprint arXiv:1905.02789*, 2019. MR4074519

[51] F. Liu and L. Ying. Recursive sweeping preconditioner for the three-dimensional helmholtz equation. *SIAM Journal on Scientific Computing*, 38(2):A814–A832, 2016. MR3473601

[52] Z. Long, Y. Lu, X. Ma, and B. Dong. PDE-net: Learning PDEs from data. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 3208–3216, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

[53] A. Lucas, M. Iliadis, R. Molina, and A. K. Katsaggelos. Using deep neural networks for inverse problems in imaging: beyond analytical methods. *IEEE Signal Processing Magazine*, 35(1):20–36, 2018.

[54] J. Ma, R. P. Sheridan, A. Liaw, G. E. Dahl, and V. Svetnik. Deep neural nets as a method for quantitative structure-activity relationships. *Journal of Chemical Information and Modeling*, 55(2):263–274, 2015.

[55] P. Mora. Nonlinear two-dimensional elastic inversion of multioffset seismic data. *Geophysics*, 52(9):1211–1228, 1987.

[56] S. J. Norton and M. Linzer. Ultrasonic reflectivity tomography: reconstruction with circular transducer arrays. *Ultrasonic Imaging*, 1(2):154–184, 1979.

[57] A. Palermo, S. Krödel, A. Marzani, and C. Daraio. Engineered metabarrier as shield from seismic surface waves. *Scientific Reports*, 6:39356, 2016.

[58] M. Raissi and G. E. Karniadakis. Hidden physics models: Machine learning of nonlinear partial differential equations. *Journal of Computational Physics*, 357:125 – 141, 2018. MR3759415

[59] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019. MR3881695

[60] K. Rudd and S. Ferrari. A constrained integration (CINT) approach to solving partial differential equations using artificial neural networks. *Neurocomputing*, 155:277–285, 2015. MR3211506

[61] J. Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, 2015.

[62] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc., 2014.

[63] C. Tan, S. Lv, F. Dong, and M. Takei. Image reconstruction based on convolutional neural network for electrical resistance tomography. *IEEE Sensors Journal*, 19(1):196–204, 2018.

[64] D. Verschuur and A. Berkhout. Estimation of multiple scattering by iterative inversion, part ii: Practical aspects and examples. *Geophysics*, 62(5):1596–1611, 1997.

[65] A. B. Weglein, F. V. Araújo, P. M. Carvalho, R. H. Stolt, K. H. Matson, R. T. Coates, D. Corrigan, D. J. Foster, S. A. Shaw, and H. Zhang. Inverse scattering series and seismic exploration. *Inverse Problems*, 19(6):R27, 2003. MR2036527

Yuwei Fan
Stanford University, Stanford, CA 94305
USA
E-mail address: ywfan1989@gmail.com

Lexing Ying
Stanford University, Stanford, CA 94305
USA
E-mail address: lexing@stanford.edu

Received December 29, 2021