Multi-Frequency Progressive Refinement for Learned Inverse Scattering

Owen Melia [1] **Olivia Tsang** [1] **Vasileios Charisopoulos** [2] **Yuehaw Khoo** [3] **Jeremy Hoskins** [3] **Rebecca Willett** [1 2 3]

Abstract

Interpreting scattered acoustic and electromagnetic wave patterns is a computational task that enables remote imaging in a number of important applications, including medical imaging, geophysical exploration, sonar and radar detection, and nondestructive testing of materials. However, accurately and stably recovering an inhomogeneous medium from far-field scattered wave measurements is a computationally difficult problem, due to the nonlinear and non-local nature of the forward scattering process. We design a neural network, called Multi-Frequency Inverse Scattering Network with Refinement (MFISNet-Refinement), and a training method to approximate the inverse map from far-field scattered wave measurements at multiple frequencies. Our method is inspired by the recursive linearization method — a commonly used technique for stably inverting scattered wavefield data — that progressively refines the estimate with higher frequency content. MFISNet-Refinement outperforms existing methods in regimes with high-contrast, heterogeneous large objects, and inhomogeneous unknown backgrounds.

1. Introduction

Wave scattering is an important imaging technology with applications in medical and seismic imaging, sonar and radar detection, and nondestructive testing of materials. In this setting, a known source transmits incident waves through a penetrable medium, and due to an inhomogeneity in the spatial region of interest, the incident waves are scattered. Several receivers measure the scattered wave field at distant locations. We are interested in the inverse wave scattering problem: given a set of scattered wave field measurements, we want to recover the inhomogeneity in the spatial region of interest that produced the measurements. In this paper, we focus on the inverse wave scattering problem with unknown medium and full-aperture measurements at multiple incident wave frequencies. This problem is characterized by a highly nonlinear forward measurement operator, making the recovery of the scattering potential challenging. We propose a machine learning solution to this problem: given a training set of pairs of scattering potentials and scattered wavefield measurements, we seek to approximate the inversion map with a deep neural network that predicts a scattering potential from scattered wavefield measurements at multiple frequencies. We design a new training method and a new neural network architecture to achieve this goal.

The aforementioned inverse wave scattering problem has been widely studied. While the measurement operator is known to be injective when there are infinitely many sensors positioned in a ring around the scattering potential (Colton & Kress, 2018), computational approaches must always operate in the ill-posed case where finite receivers are present. Thus, past research has focused on optimization approaches to solving the inverse problem. Simple gradient-based optimization approaches to this problem face two major difficulties: computing a gradient requires solving a partial differential equation (PDE), which can be computationally expensive; additionally, the nonlinearity of the forward model induces a non-convex objective function. Therefore, convergence of local search methods such as gradient descent is not guaranteed without careful initialization.

To remedy this, many machine learning approaches have proposed constructing data-driven approximations of the inverse map. These methods often use general-purpose architectures (Wei & Chen, 2019; Chen et al., 2020) which ignore the field's deep understanding of the forward scattering map. Recently, Khoo & Ying (2019); Fan & Ying (2022); Li et al. (2022) proposed to leverage knowledge of the forward map by making a global linearization assumption about the relationship between the scattering potential and the measurements. A natural alternative is integrating machine-learning models into iterative reconstruction methods, which are computationally demanding relative to a global linear approximation but can provide higher quality reconstructions.

[1]Department of Computer Science, University of Chicago, US [2]Data Science Institute, University of Chicago, US [3]Computational and Applied Mathematics, Department of Statistics, University of Chicago, US. Correspondence to: <{meliao, oortsang, vchariso, ykhoo, jeremyhoskins, willett}@uchicago.edu>.

ICML 2024 AI for Science Workshop, Vienna, Austria. 2024.
Copyright 2024 by the author(s).

A standard approach, which has been successful in the strongly nonlinear scattering regime, is to use data collected at multiple incoming wave frequencies. Recursive linearization methods (Chen, 1995; Bao & Liu, 2003; Borges et al., 2017) use multi-frequency measurements to solve a sequence of sub-problems, starting at the lowest frequency to provide an initial estimate of the scattering potential and refining that estimate at progressively higher frequencies using warm-started local search methods. Algorithms in this family offer two benefits: first, they alleviate the need for careful initialization since the loss landscape of the lowest frequency sub-problem is typically well-behaved; second, they greatly reduce the number of PDE solves by relying on first-order approximations of the forward model that are relatively inexpensive to invert. However, these methods require measurements at a large number of incident wave frequencies and still involve solving large-scale PDEs and least-squares problems for each frequency. This requires, for example, multiple CPU core-hours to recover a single image, even with a state-of-the-art PDE solver (Borges et al., 2017).

In light of these advances, we propose a new architecture and training method inspired by the recursive linearization algorithm. Our approach is based on a residual update architecture and training method that ensures specific network blocks solve specific sub-problems.

1.1. Contributions & paper outline

In Section 2, we formally define the inverse scattering problem and the machine learning objective. We present standard results about inverse scattering and survey related work in Section 3. In Section 4, we review the recursive linearization algorithm and introduce our method, MFISNet-Refinement. Finally, we present a numerical evaluation of our method in Section 5. Our main contributions can be summarized as follows:

1. We introduce "MFISNet-Refinement", short for "Multi-Frequency Inverse Scattering Network with Refinement", a neural network architecture and training method that is inspired by recursive linearization algorithms (Chen, 1995; Bao & Liu, 2003; Borges et al., 2017). (Section 4)

2. We show that our network achieves lower errors than single-frequency methods (Fan & Ying, 2022) and multi-frequency methods (Li et al., 2022) in a high-contrast, noiseless, full-aperture setting. (Section 5.1)

3. We consider alternative training strategies and find that the proposed progressive training strategy made possible by the MFISNet-Refinement architecture is important to its performance. (Section 5.2)

2. Problem Setup and Notation

The forward model for our imaging setup is implicitly defined by a PDE problem involving the Helmholtz equation. Let $x \in \mathbb{R}^2$ be the spatial variable. Suppose $u_{\text{in}}(x; s) = e^{ikx\cdot s}$ is an incoming plane wave with direction $s \in \mathbb{S}^1$, wavelength λ, and angular wavenumber $k = 2\pi/\lambda$. We normalize the problem's units so this wave travels at speed $c_0 \equiv 1$ in free space. The incoming wave interacts with a real-valued scattering potential $q(x)$ to produce an additive perturbation, called the scattered wave field $u_{\text{scat}}[q](x; s)$. We define $q(x) = c_0^2/c^2(x) - 1$ where $c(x)$ is the wave speed at x. The total wave field $u[q](x; s) = u_{\text{scat}}[q](x; s) + u_{\text{in}}(x; s)$ solves the following inhomogeneous Helmholtz equation:

$$\Delta u[q](x; s) + k^2(1 + q(x))u[q](x; s) = 0, \; x \in \mathbb{R}^2, \quad \text{(1a)}$$

subject to the Sommerfield radiation boundary condition:

$$\frac{\partial u_{\text{scat}}[q](x; s)}{\partial \|x\|} - iku_{\text{scat}}[q](x; s) = o(\|x\|^{-1/2}), \quad \text{(1b)}$$

as $\|x\| \to \infty$.

We assume $q(x)$ is supported on a square domain $\Omega = [-0.5, 0.5]^2$, and we work with $q \in \mathbb{R}^{N_q \times N_q}$, the discretization of $q(x)$ onto a regular grid with (N_q, N_q) grid points. We place the receivers equally spaced around a large ring of radius $R \gg 1$, centered at the origin. We identify individual receivers by their unit-vector directions $r_j \in \mathbb{S}^1$. We compute the solution for a set of N_r receiver directions and N_s incoming wave directions s_ℓ equally-spaced about the unit circle. This results in a set of (N_r, N_s) observations, $\{u_{\text{scat}}[q](Rr_j; s_\ell)\}_{j,\ell \in [N_r]\times[N_s]}$, which we arrange in a data array $d_k \in \mathbb{C}^{N_r \times N_s}$. We call the mapping from q to d_k the forward model with incoming wave frequency k:

$$(d_k)_{j,\ell} = \mathcal{F}_k[q]_{j,\ell} \equiv u_{\text{scat}}[q](Rr_j; s_\ell) \quad (2)$$

Because we are interested in multi-frequency algorithms, we are interested in observations of the forward model evaluated on the same q but with a set of incoming wave frequencies $[k_1, \ldots, k_{N_k}]$. In particular, our goal is to approximate the following mapping:

$$\left[\mathcal{F}_{k_1}[q], \ldots, \mathcal{F}_{k_{N_k}}[q]\right] \mapsto q. \quad (3)$$

Our goal is to train a neural network g_θ with parameters θ to approximate the mapping: $g_\theta(\mathcal{F}_{k_1}[q], \ldots, \mathcal{F}_{k_{N_k}}[q]) \approx q$. Given a distribution $\mathcal{D}$ over scattering potentials q, we draw a training set of n independent samples from $\mathcal{D}$ to generate data to train the neural network. After evaluating the forward model nN_k times, we have a training set

$$\mathcal{D}_n := \left\{(q^{(j)}, \mathcal{F}_{k_1}[q^{(j)}], \ldots, \mathcal{F}_{k_{N_k}}[q^{(j)}])\right\}_{j=1}^{n} \quad (4)$$

We evaluate networks by measuring the relative ℓ_2 error:

$$\mathbb{E}_{q\sim\mathcal{D}}\left[\frac{\|g_\theta\left(\mathcal{F}_{k_1}[q],\ldots,\mathcal{F}_{k_{N_k}}[q]\right)-q\|_2}{\|q\|_2}\right] \tag{5}$$

In practice, we approximate the expected relative ℓ_2 error in (5) by an empirical mean over a held-out test set of 1,000 samples drawn independently from $\mathcal{D}$.

3. Background and Related Work

3.1. Background

In this section, we review standard results about $\mathcal{F}_k$ relevant to our study. In particular, we focus on a linear approximation of $\mathcal{F}_k$ that gives insights into the inverse scattering problem.

The first result is that $\mathcal{F}_k$ becomes more nonlinear as the magnitude of the scatterer $\|q\|$ or the wavenumber k increases. Indeed, the solution to (1a)-(1b) can be equivalently defined as the solution to the Lippmann-Schwinger integral equation:

$$\begin{aligned} &u_{\text{scat}}[q](x;s) \\ &= k^2\int_\Omega G_k(\|x-x'\|)q(x')u_{\text{in}}(x';s)\mathrm{d}x' \\ &+ k^2\int_\Omega G_k(\|x-x'\|)q(x')u_{\text{scat}}[q](x';s)\mathrm{d}x' \end{aligned} \tag{6}$$

where G_k is the Green's function for the homogeneous Helmholtz operator. This recursive equation provides a nonlinear map from $q(x)$ to $u_{\text{scat}}[q](\cdot;s)$ and therefore $\mathcal{F}_k[q]$.

One way to view this nonlinearity is to interpret the Lippmann-Schwinger equation as a power series in $q(x)$ by iteratively substituting the value of $u_{\text{scat}}[q](x)$ into its appearance on the right-hand side of (6). For example, performing this substitution once yields a linear and a quadratic term in $q(x)$ that are independent of $u_{\text{scat}}[q]$, as well as a "remainder" term involving the unknown $u_{\text{scat}}[q]$ that accounts for higher-order terms. This power series does not converge for general $q(x)$, but it helps illustrate which parts of the problem drive the nonlinearity of the operator $\mathcal{F}_k$: as $\|q\|$ or k grow, the size of these nonlinear terms will also grow, and as a result $\mathcal{F}_k$ becomes highly nonlinear.

The next result is that, under a linear approximation, the far-field measurements are diffraction-limited and can only capture frequency components of q up to $2k$. Equivalently, the measurements depend on q to a spatial resolution of $\lambda/2$. We consider the first-order *Born approximation* (Born et al., 1999), which approximates (6) by dropping the $u_{\text{scat}}[q](x';s)$ term from the right-hand side. This is further simplified with an approximation of the Green's function in the far-field limit (Born et al., 1999), yielding

$$d_k(r,s)\approx k^2\int_\Omega e^{-ik(r-s)\cdot x'}q(x')\mathrm{d}x'. \tag{7}$$

We will refer to this linear approximation of the map from $q(x)$ to $d_k(r,s)$ as F_k. Note that F_kq is proportional to the Fourier Transform of q evaluated at frequency vectors of the form $k(r-s)$. Since $r,s\in\mathbb{S}^1$ range over the unit circle, the frequency vectors $k(r-s)$ take on values throughout a disk with radius $2k$ centered at the origin. Thus, evaluations of the linearized forward model F_kq only contain the low-frequency components of q, while high-frequency components are in the kernel of F_k (Chen, 1995).

Owing to its simplicity, (7) is often used as inspiration for the design of neural network architectures approximating the inverse map $d_k\mapsto q$. The networks emulate the *filtered back-projection (FBP) method* (Natterer, 2001), which produces an estimate $\hat{q}$ of the scattering potential q as

$$\hat{q}=(F_k^*F_k+\mu I)^{-1}F_k^*d_k, \tag{8}$$

where F_k^* is the adjoint of F_k. The operator $(F_k^*F_k+\mu I)^{-1}$ can be implemented as a two-dimensional spatial convolution (Khoo & Ying, 2019; Fan & Ying, 2022; Li et al., 2022), while novel network architectures have been proposed to emulate $F_k^*\in\mathbb{C}^{N_q^2\times N_rN_s}$ in a parameter-efficient manner. In particular, Fan & Ying (2022) and Zhang et al. (2023) suggest leveraging the rotational equivariance of the forward model to emulate F_k^* with one-dimensional convolutions, after applying a far-field scaling and an appropriate coordinate transformation in (Fan & Ying, 2022, Equation 6)). We refer to the network described by Fan & Ying (2022) as FYNet.

3.2. Related Work

Deep learning has revolutionized linear inverse problems in imaging, advancing methods for superresolution, inpainting, deblurring, and medical imaging. Many of these advances stem from methods combining deep neural networks with optimization algorithms. For example, the *deep unrolling* paradigm (Monga et al., 2021) performs a fixed number of steps of an iterative algorithm and replaces certain operations with learnable mappings, which are parameterized by neural networks whose weights are learned from data. Components that remain fixed throughout training may reflect prior knowledge of problem parameters, such as explicit knowledge of the forward measurement model. In this setting, the network is usually trained end-to-end by minimizing the Euclidean distance between the network outputs and the true data. Another paradigm, called *plug-and-play denoising* (Venkatakrishnan et al., 2013), suggests that general image denoisers can be used in place of proximal operators for regularization functions, an important subroutine

in many optimization routines for linear inverse problems in imaging. In this setting, neural network blocks are often trained to solve a different task, such as denoising corrupted signals, and then used inside the inversion algorithm. Ongie et al. (2020) provides a review of deep learning for inverse problems in imaging.

Several works in the wave scattering literature attempt to solve the inverse scattering problem by augmenting an optimization algorithm with components learned from data. At inference time, these methods require running an iterative optimization algorithm. Kamilov et al. (2017) develop a plug-and-play algorithm for inverse scattering, and show that various off-the-shelf denoisers can be applied as proximal operators. Zhao et al. (2023) use an encoder network paired with a network emulating the forward model and suggest optimizing a latent representation of the scattering potential using stochastic gradient descent. When only phaseless measurements of the scattered wave field u_{scat} are available, Deshmukh et al. (2022) propose a network unrolling proximal gradient descent, where the proximal operator is a neural network learned from data. For the inverse obstacle scattering problem in two dimensions, Zhou et al. (2023) propose using a fully-connected neural network to warm-start a Gauss-Newton algorithm. Ding et al. (2022) train a neural network to approximately invert a forward scattering process depending on temporal data, and use this approximate inverse as a nonlinear preconditioner for a nonlinear least squares optimization routine.

Other methods propose to learn the inverse map directly from data. Recently, neural networks that are approximately invariant to discretization size have been proposed as methods of learning maps between general function spaces (Li et al., 2021b; Lu et al., 2021) and these general-purpose networks have been applied to inverse scattering (Ong et al., 2022). Other networks have been designed to invert the forward scattering model in particular; see Chen et al. (2020) for a broad review of such approaches. One particularly successful approach has been to emulate the FBP method (Khoo & Ying, 2019; Fan & Ying, 2022). In our work, we consider how to combine multiple such network blocks to invert the multi-frequency forward map in (3). One way of combining these blocks is to learn each adjoint operator $F_{k_t}^*$ as a separate neural network block and combine data to jointly emulate the learned filtering operators $(F_{k_t}^* F_{k_t} + \mu I)^{-1}$. This strategy is employed in Zhang et al. (2023). Another strategy is to use the Wide-Band Butterfly Network (Li et al., 2022; 2021a), which hierarchically merges information from different frequencies in the network block emulating $F_{k_t}^*$.

Finally, we note in passing that the design of our method is inspired by *homotopy methods* (Watson & Haftka, 1989). These methods solve a sequence of sub-problems of increasing difficulty, gradually transforming a simple (but uninformative) optimization problem to the optimization problem of interest and using solutions to a given sub-problem to warm-start local search methods for subsequent sub-problems. Such a sequence can be constructed explicitly (*e.g.*, by varying regularization levels) or implicitly; for example, *curriculum learning* (Bengio et al., 2009) progressively adjusts the training data distribution from "easy" to "hard" samples and has been used to train physics-informed neural networks in challenging problem settings (Krishnapriyan et al., 2021; Huang et al., 2022).

4. Recursive Linearization and Our Method

We propose a neural network that learns to approximate the multi-frequency inversion map from training data. To design the network and training algorithm, we draw inspiration from the recursive linearization method for inverse scattering, which we briefly review below.

4.1. Recursive Linearization

Recursive linearization is a classical method for solving the inverse scattering problem, introduced by Chen (1995). In spite of the nonlinearity of the true forward scattering model described in (6), recursive linearization breaks the inverse problem into a series of simpler problems, each of which corresponds to a linear inverse problem. In this section we discuss the intuition behind this strategy.

Recall from our discussion in Section 3.1 that the forward map evaluated at low incident wave frequencies k acts approximately like a low-pass filter with cutoff frequency $2k$. At first glance, this suggests that observing $\mathcal{F}_k[q]$ for a high value of k is sufficient for high-resolution recovery of q. However, when viewed from an optimization perspective, it becomes clear that this problem is increasingly challenging for large values of k. For example, one might consider the nonlinear least-squares problem

$$\underset{\hat{q}}{\text{minimize}} \; \|d_k - \mathcal{F}_k[\hat{q}]\|_2^2. \tag{9}$$

To illustrate the challenges for the optimization formulation with increasing values of k, we consider a simple example where q is known to be a Gaussian bump with a given spread parameter and unknown amplitude. Given observations $d_k = \mathcal{F}_k[q]$ and a numerical PDE solver to calculate $\mathcal{F}_k[\cdot]$, one could estimate the amplitude of q by solving a minimization problem similar to (9), but only searching over the unknown amplitude. In Figure 1, we plot this objective as a function of the amplitude of $\hat{q}$ for a range of values of k. For large values of k, the objective function is highly oscillatory and contains many spurious local minima. When optimizing using high-frequency data, convergence to the global optimum is only guaranteed with careful initialization. That is, if we want to use measurements $\mathcal{F}_k[q]$ with

high k to find a high-resolution reconstruction, we need a careful initialization near the ground-truth q.

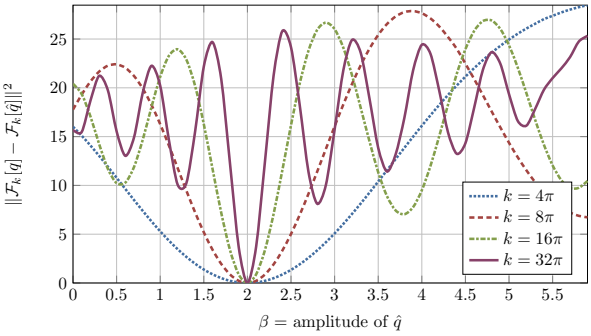


Figure 1: Even in a highly stylized setting, accurately and reliably inverting $\mathcal{F}_k$ is difficult for high frequencies k. Suppose the ground-truth scattering potential is $q(x) = \beta \exp(-\frac{\|x\|^2}{2\sigma^2})$, a Gaussian bump with known spread parameter $\sigma = 0.1$ but unknown amplitude β. We show the optimization landscape that arises from searching over different amplitudes. At low incident wave frequencies, this optimization landscape is smooth and has a large basin of attraction. However, as the incident wave frequency increases, the optimization landscape becomes highly oscillatory, requiring a nearly exact initialization to guarantee convergence to the ground truth. In experimental settings, the parameterization of q is often high-dimensional, which requires higher frequency data to resolve high-frequency information in q. This numerical example was inspired by Bao & Liu (2003).

The recursive linearization algorithm leverages this insight by solving a sequence of inversion problems at increasing wave frequencies k. Crucially, each sub-problem uses the output of the previous sub-problem to initialize a new optimization problem. This method was introduced in Chen (1995) and further developed in Bao & Liu (2003); Borges et al. (2017). At iteration t, the algorithm uses the previous estimate $\hat{q}_{k_{t-1}}$ along with a new set of observations d_{k_t}, and it calculates an update δq that minimizes the ℓ_2 distance in measurement space:

$$\underset{\delta q}{\operatorname{argmin}} \; \|d_{k_t} - \mathcal{F}_{k_t}[\hat{q}_{k_{t-1}} + \delta q]\|_2^2 \tag{10}$$

The value of $\hat{q}_{k_{t-1}}$ may make it possible to avoid spurious local minima, but this problem is still difficult since an iterative optimizer would require solving a PDE at each of its iterations. However, $\mathcal{F}_{k_t}[\hat{q}_{k_{t-1}} + \delta q]$ is well-approximated by a first-order Taylor expansion about $\hat{q}_{k_{t-1}}$ when δq is small or when it does not contain low-frequency information (Chen, 1995). This motivates the following surrogate for the optimization problem in (10):

$$\underset{\delta q}{\operatorname{argmin}} \; \|d_{k_t} - \left(\mathcal{F}_{k_t}[\hat{q}_{k_{t-1}}] + D\mathcal{F}_{k_t}[\hat{q}_{k_{t-1}}]\delta q\right)\|_2^2, \tag{11}$$

where $D\mathcal{F}_{k_t}[\hat{q}_{k_{t-1}}]$ denotes the Fréchet derivative of the forward model at $\hat{q}_{k_{t-1}}$. The action of $D\mathcal{F}_{k_t}[\hat{q}_{k_{t-1}}]$ and its adjoint, $D\mathcal{F}^*_{k_t}[\hat{q}_{k_{t-1}}]$, can be computed using the adjoint-state method (Bao & Liu, 2003; Borges et al., 2017). The resulting algorithm is akin to a Gauss-Newton method; critically, each sub-problem of the form shown in (11) is a linear least-squares problem. We outline a sketch of the recursive linearization algorithm in Algorithm 1.

The recursive linearization algorithm is very demanding computationally. Each iteration requires solving N_s large-scale PDEs and a high-dimensional least-squares problem, which quickly creates a large computational burden when producing high-resolution solutions. In a classical setting without machine learning, the frequencies should be spaced close to each other for best results. Chen (1997) uses $k = 1, 2, \ldots, 9$ in their numerical experiments, while Borges et al. (2017) uses $k = 1, 1.25, \ldots, 70$, which they report takes around 40-50 hours per sample to produce a single 241×241 pixel image.

Algorithm 1 Recursive Linearization for Inverse Scattering based on Chen (1995; 1997); Bao & Liu (2003); Borges et al. (2017)

1: **Input:** Multi-frequency data $\{d_{k_1}, d_{k_2}, \ldots, d_{k_{N_k}}\}$
2: $\hat{q}_{k_1} \leftarrow \left(F^*_{k_1} F_{k_1} + \mu I\right)^{-1} F^*_{k_1} d_{k_1}$
3: **for** $t = 2, \ldots, N_k$ **do**
4: Compute $\mathcal{F}_{k_t}[\hat{q}_{k_{t-1}}]$ and $D\mathcal{F}_{k_t}[\hat{q}_{k_{t-1}}]$.
5: Compute δq_{k_t} by solving (11)
6: $\hat{q}_{k_t} \leftarrow \hat{q}_{k_{t-1}} + \delta q_{k_t}$.
7: **end for**
8: **Result:** Final estimate $\hat{q}_{k_{N_k}}$.

Although recursive linearization is computationally expensive as stated, we believe that one of the key features of recursive linearization is the way that it breaks the recovery process into multiple steps, each of which refines the estimate from the previous step using data of a higher frequency. We will refer to this step-wise recovery strategy as progressive refinement.

Progressive refinement facilitates the recovery process, since each step is only responsible for a correction to the estimate of the scattering potential within a frequency band. Focusing on this strategy also allows us to look for machine learning methods that do not explicitly emulate $\mathcal{F}_k[\cdot]$ or $D\mathcal{F}_k[\cdot]$, which are expensive to compute. To this end, we consider a generalization of recursive linearization where we replace lines 4 and 5 in Algorithm 1 with a generic refinement step: for $t = 2, \ldots, N_k$,

$$\delta q_{k_t} = \text{RefinementStep}_{k_t}(\hat{q}_{k_{t-1}}, d_{k_t}), \tag{12}$$

where $\text{RefinementStep}_{k_t}(\hat{q}_{k_{t-1}}, d_{k_t})$ refers to the update cal-

culated for estimate $\hat{q}_{k_{t-1}}$ given data d_{k_t} and can be implemented using a neural network. We will propose and discuss a network architecture in the next section.

4.2. Our Method

We use Algorithm 1 as inspiration for the design of our neural network architecture and training method. In particular, we focus on the following two crucial aspects of Algorithm 1:

Progressive refinement: The algorithm builds intermediate estimates of the scattering potential which are progressively refined with the introduction of new data.

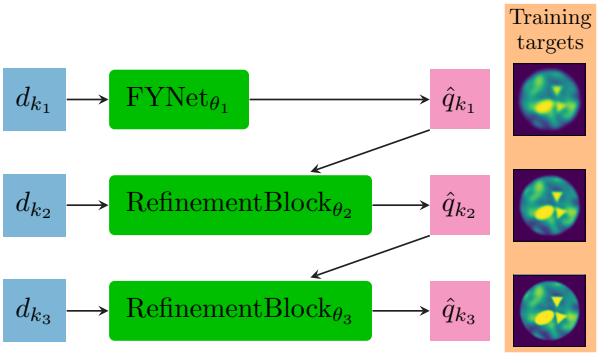
Homotopy through frequency: The iterative refinements from the first step form a homotopy from low to high-frequency measurements. As a result, updates at step t contain high-frequency information relative to k_{t-1}.

To emulate the progressive refinement structure, we propose a network with a residual structure and skip connections. The network comprises multiple blocks, one for each incident wave frequency k_t, $t = 1, \ldots, N_k$. The input to each block is measurement data d_{k_t} collected at a particular incident wave frequency k_t. The input passes through an FYNet block (Fan & Ying, 2022), which approximately inverts the forward model. The output of the FYNet block is then concatenated with the output of the previous block, $\hat{q}_{k_{t-1}}$, and the concatenation is passed to 2D convolutional layers for a second filtering step. Finally, a skip connection adds $\hat{q}_{k_{t-1}}$ to the output of the last convolutional layer of the block, producing the next estimate $\hat{q}_{k_t}$. The network's architecture is shown in Figure 2; we call the resulting network "MFISNet-Refinement." Note that, under this construction, the FYNet blocks could be replaced by any other neural network architecture designed for the single-frequency inverse scattering problem.

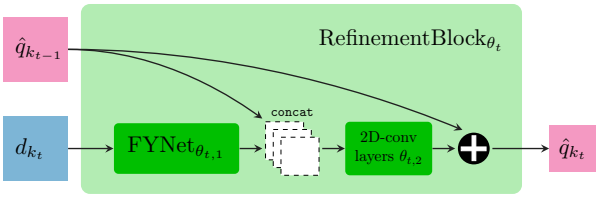
To emulate the homotopy through frequency, we design a training method to ensure each successive block adds higher-frequency information to the estimate of the scattering potential. Under the Born approximation (7), we know d_k contains information about q up to frequency limit $2k$. This suggests that given data d_{k_t}, we should be able to reconstruct the frequency components of q up to $2k_t$. To reflect this, we train the output of block t with the following loss function:

$$L_t(\hat{q}_{k_t}; q) = \mathbb{E}_{q \sim \mathcal{D}_n} \left[\|\hat{q}_{k_t} - \mathsf{LPF}_{2k_t} q\|^2 \right] \tag{13}$$

In (13), LPF_{2k_t} is a low-pass filter with approximate cutoff frequency $2k_t$, implemented as a Gaussian filter to avoid ringing artifacts. To train the network, we first adjust the weights of each block in a sequential fashion and then perform a final training step which fine-tunes all of the blocks jointly; the training procedure is summarized in Algorithm 2.



(a) MFISNet-Refinement Architecture



(b) Refinement Block

Figure 2: Our MFISNet-Refinement architecture is designed to emulate the recursive linearization algorithm. Figure 2a shows that our network proceeds by making an initial low-frequency reconstruction and then making a series of updates given higher-frequency data and an estimate of the scattering potential. The network is trained to match the intermediate reconstructions to low-pass filtered versions of scattering potentials from the training set. One example collection of such filtered scattering potentials is shown. Figure 2b shows that our refinement block is a simple extension of the FYNet architecture from Fan & Ying (2022). By using a skip connection in this block, we ensure the network only needs to predict an update to the estimated scattering potential.

5. Experiments

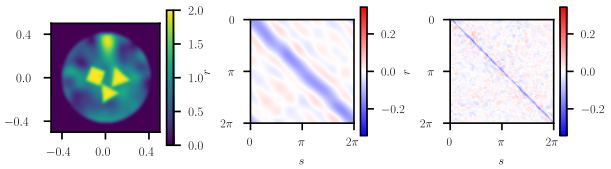
In this section, we describe the setting and results for our numerical investigation of the efficacy of our proposed method. We test the methods on a new distribution of scattering potentials with high contrast and a smoothly-varying unknown background, described in Figure 3a and Appendix B.

5.1. Stabilizing Reconstruction by Adding Frequencies

First, we test whether the intuition built in Section 4 is true in a machine learning context. We test whether machine learning methods that operate on data with multiple incom-

Algorithm 2 Training Procedure

Input: Randomly-initialized neural network parameters $\{\theta_1, \ldots, \theta_{N_k}\}$; Training data samples $\mathcal{D}_m := \left\{(q^{(j)}, d^{(j)}_{k_1}, \ldots, d^{(j)}_{k_{N_k}})\right\}_{j=1}^{n}$.
for $t = 1, ..., N_k$ **do**
  Set θ_t as trainable, and freeze all other weights.
  if $t < N_k$ **then**
    Train θ_t by optimizing L_t. { Equation (13) }
  else
    Train θ_t by optimizing $\|\hat{q}_{k_{N_k}} - q\|_2^2$.
  end if
end for
Set all weights as trainable.
Train all weights by optimizing $\|\hat{q}_{k_{N_k}} - q\|_2^2$.
Result: Trained neural network parameters $\{\theta_1, ..., \theta_{N_k}\}$.



(a) Scattering potential q **(b)** Measured d_k, $k = 4\pi$ **(c)** Measured d_k, $k = 32\pi$

Figure 3: Figure 3a shows a typical example from our distribution of scattering potentials, drawn from the test set. Our distribution of scattering potentials has a random low-frequency background field, occluded by piecewise constant geometric shapes. We reconstruct these scattering potentials on a regular grid with $(N_q, N_q) = (192, 192)$ pixels. Figures 3b and 3c show the output of the forward model applied to this scattering potential. The real part of u_{scat} is shown, which we sample on a grid of size $(N_r, N_s) = (192, 192)$ equally-spaced directions on the unit circle.

ing wave frequencies are more accurate and stable than single-frequency machine learning methods. To make this comparison fair, we create a sequence of training datasets with number of incident wave frequencies $N_k \in \{1, 2, 3\}$ and keep the amount of training data, nN_k, constant for each dataset by suitably adjusting the number of training samples n. We wish to compare our method, MFISNet-Refinement, with other methods of learning an inverse to the single-frequency or multi-frequency forward map. We give details about the FYNet and Wide-Band Butterfly Network architectures in Appendix A.1.

For the $N_k = 1$ dataset, we train an FYNet model, and for $N_k > 1$, we train our model MFISNet-Refinement, as well as the Wide-Band Butterfly Network (Li et al., 2022). For each model, we train for a fixed number of epochs and choose the model weights at the epoch at which a validation set of size $n/10$ is minimized. We also use the validation set to search over various hyperparameters, such as the size of 1D and 2D convolutional kernels, the number of channels in the convolutional layers, and optimization hyperparameters, such as step size and weight decay (Appendix C.)

We present the quantitative results of this experiment in Table 1; see also Figures 4 and 6 for a qualitative comparison on representative test samples. The relatively poor performance of FYNet confirms our belief that we are in a challenging nonlinear problem regime. As more frequencies are added, the multi-frequency methods improve. All of tested methods are uniformly outperformed by our method, MFISNet-Refinement, at all values of N_k. The Wide-Band Butterfly Network underperforms the other methods. We hypothesize that the weaker performance of the Wide-Band Butterfly Network may be attributed to several factors: first, the butterfly factorization was inspired by analysis in a weak (linear) scattering regime, but our experiments are in a strong (nonlinear) scattering regime. Additionally, the Wide-Band Butterfly Network was previously tested in low-contrast settings with sub-wavelength scatterers and a known background, while we are in an experimental setting with high contrast and an unknown, inhomogeneous background.

N_k	$[k_1, k_2, ...]$	n	**Method Name**	**Relative ℓ_2 Error**
1	$[32\pi]$	$10,000$	FYNet	0.261 ± 0.036
2	$[16\pi, 32\pi]$	$5,000$	MFISNet-Refinement (Ours)	0.154 ± 0.034
3	$[8\pi, 16\pi, 32\pi]$	$3,333$	Wide-Band Butterfly Network	0.160 ± 0.037
			MFISNet-Refinement (Ours)	$\mathbf{0.098 \pm 0.020}$

Table 1: When holding the number of forward model evaluations $= nN_k$ constant, methods trained on more frequencies outperform methods with fewer frequencies. The final column reports the relative ℓ_2 error mean $\pm$ one standard deviation computed over $1,000$ held-out test samples. The lowest mean for each incident frequency set is marked in boldface font.

5.2. Investigating the Training Method

We hypothesize that our method is successful because it effectively breaks the reconstruction problem into a sequence of simpler frequency-dependent refinement steps. This emulates the sequential, frequency-dependent structure of the recursive linearization method. A natural question is whether the success of our method is primarily driven by emulating sequential refinement steps or can be attributed to the residual architecture alone. To test this question, we investigate two different changes to our training method which remove the progressive refinement structure. We describe each adjustment below.

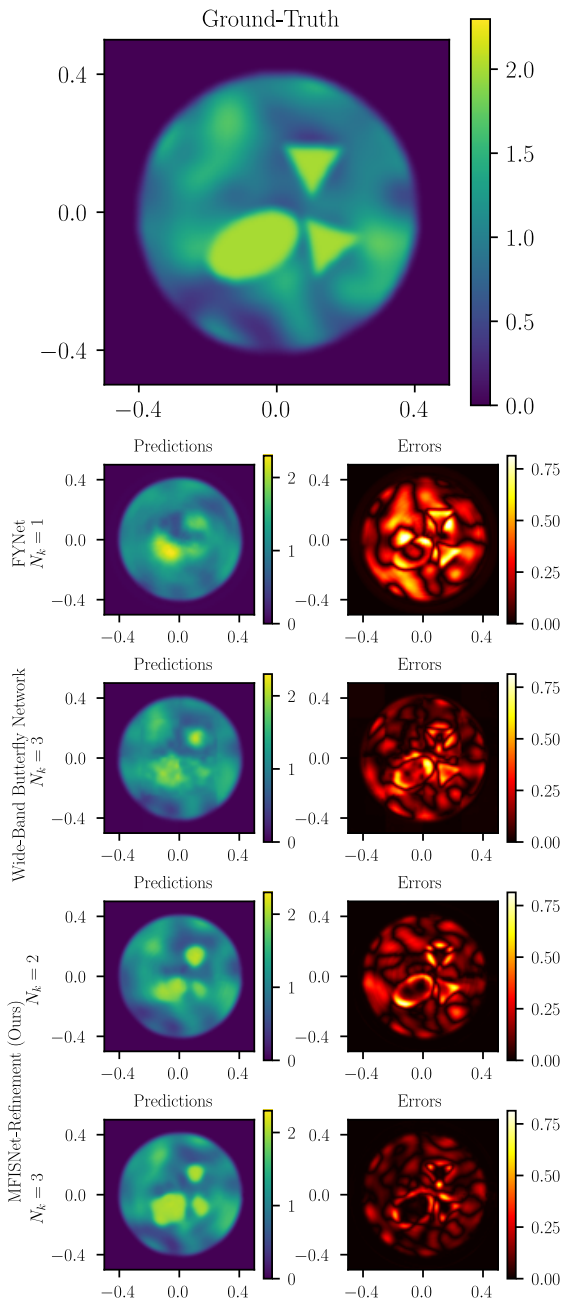


Figure 4: Sample predictions from models trained on different datasets. Predictions and errors on one held-out test sample are shown for single-frequency methods (FYNet) and multi-frequency methods (Wide-Band Butterfly Network and MFISNet-Refinement). The first row shows the ground-truth scattering potential. See Figure 6 in the appendix for outputs on additional test samples.

No Homotopy through Frequency Rather than sequentially training each network block after the previous blocks have been optimized, we jointly train all of the blocks by optimizing the loss function

$$\|\hat{q}_{k_{N_k}} - q\|^2 + \sum_{t=1}^{N_k-1} \gamma^{N_k-t} \|\hat{q}_{k_t} - \mathsf{LPF}_{2k_t} q\|^2 \qquad (14)$$

Here γ is a hyperparameter which controls the relative importance of the different loss terms. We tuned over a few choices of γ; see Table 5 for details.

No Progressive Refinement Instead of using the intermediate loss terms $\|\hat{q}_{k_t} - \mathsf{LPF}_{2k_t} q\|^2$, designed to promote specific network blocks learning different parts of the reconstruction, we train the network by only optimizing the final loss term $\|\hat{q}_{k_{N_k}} - q\|^2$. This is the standard training loss used in most other works, including Khoo & Ying (2019); Fan & Ying (2022); Li et al. (2022).

Training Method	$[k_1, k_2, k_3]$	Relative L2 Error
No Progressive Refinement	$[8\pi, 16\pi, 32\pi]$	0.110 ± 0.021
No Homotopy through Frequency	$[8\pi, 16\pi, 32\pi]$	0.103 ± 0.021
Our Method	$[8\pi, 16\pi, 32\pi]$	$\mathbf{0.098 \pm 0.020}$

Table 2: Our training method, described in Algorithm 2, produces better results than the other methods, which were designed to remove parts of the recursive linearization structure. We describe these alternate training methods in Section 5.2.

6. Conclusion

This paper investigates the use of multi-frequency data in deep learning approaches to the inverse medium scattering problem in a highly nonlinear, full-aperture regime. We review standard optimization results for this problem, identify recursive linearization as an algorithm particularly well-suited for this problem, and use this insight to design a neural network architecture and training method. Our experimental evaluations suggest that our method outperforms existing approaches that combine multi-frequency data across a wide range of data settings.

Our work leaves open important questions about machine learning in different multi-frequency data settings, which we defer to future work. The first setting is seismic imaging, where sources and receivers are located on one side of the scattering potential, resulting in limited-aperture measurements. Another setting to consider is full-aperture measurements, with a small fixed or frequency-dependent number of source and receiver directions, as N_s and N_r drive real-world costs when implementing an imaging system. Finally, we suggest a distribution of scattering potentials with an unknown, smoothly varying background occluded by strongly scattering shapes. We note that an important open problem is to learn to *segment* the reconstruction into disjoint regions, containing only background or only the strong scatterers.

Author Contributions

- **Owen Melia:** Conceptualization; Data curation; Methodology; Software; Writing - original draft.
- **Olivia Tsang:** Conceptualization; Data curation; Methodology; Software; Writing - original draft.
- **Vasileios Charisopoulos:** Conceptualization; Methodology; Supervision; Writing - review & editing.
- **Yuehaw Khoo:** Conceptualization; Methodology; Supervision; Writing - review & editing.
- **Jeremy Hoskins:** Conceptualization; Methodology; Supervision; Writing - review & editing.
- **Rebecca Willett:** Conceptualization; Methodology; Supervision; Funding acquisition; Project administration; Writing - review & editing.

Acknowledgements

The authors would like to thank Borong Zhang for sharing the implementation of the Wide-Band Butterfly Network. OM, OT, VC, and RW gratefully acknowledge the support of AFOSR FA9550-18-1-0166 and NSF DMS-2023109. RW and YK gratefully acknowledge the support of DOE DE-SC0022232. The team gratefully acknowledges the support of the Margot and Tom Pritzker Foundation.

References

Bao, G. and Liu, J. Numerical Solution of Inverse Scattering Problems with Multi-experimental Limited Aperture Data. *SIAM Journal on Scientific Computing*, 25(3):1102–1117, January 2003. ISSN 1064-8275. doi:10.1137/S1064827502409705. URL https://epubs.siam.org/doi/abs/10.1137/S1064827502409705. Publisher: Society for Industrial and Applied Mathematics.

Bengio, Y., Louradour, J., Collobert, R., and Weston, J. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pp. 41–48, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605585161. doi:10.1145/1553374.1553380. URL https://doi.org/10.1145/1553374.1553380.

Borges, C., Gillman, A., and Greengard, L. High resolution inverse scattering in two dimensions using recursive linearization. *SIAM Journal on Imaging Sciences*, 10(2): 641–664, 2017. doi:10.1137/16M1093562.

Born, M., Wolf, E., and Bhatia, A. B. *Principles of optics: electromagnetic theory of propagation, interference and diffraction of light*. Cambridge University Press, Cambridge [England] ; New York, 7th (expanded) edition, 1999. ISBN 978-0-521-64222-4.

Chen, X., Wei, Z., Li, M., and Rocca, P. A review of deep learning approaches for inverse scattering problems (Invited review). *Progress In Electromagnetics Research*, 167:67–81, 2020. ISSN 1559-8985. doi:10.2528/PIER20030705. URL http://www.jpier.org/PIER/pier.php?paper=20030705.

Chen, Y. Recursive Linearization for Inverse Scattering. Technical Report YALEU/DCS/RR-1088, Yale University, October 1995.

Chen, Y. Inverse scattering via Heisenberg's uncertainty principle. *Inverse Problems*, 13(2):253, April 1997. ISSN 0266-5611. doi:10.1088/0266-5611/13/2/005. URL https://dx.doi.org/10.1088/0266-5611/13/2/005.

Colton, D. and Kress, R. Looking Back on Inverse Scattering Theory. *SIAM Review*, 60(4):779–807, January 2018. ISSN 0036-1445, 1095-7200. doi:10.1137/17M1144763. URL https://epubs.siam.org/doi/10.1137/17M1144763.

Deshmukh, S., Dubey, A., and Murch, R. Unrolled Optimization With Deep Learning-Based Priors for Phaseless Inverse Scattering Problems. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–14, 2022. ISSN 1558-0644. doi:10.1109/TGRS.2022.3214495. Conference Name: IEEE Transactions on Geoscience and Remote Sensing.

Ding, W., Ren, K., and Zhang, L. Coupling Deep Learning with Full Waveform Inversion, March 2022. URL http://arxiv.org/abs/2203.01799. arXiv:2203.01799 [cs, math].

Fan, Y. and Ying, L. Solving inverse wave scattering with deep learning. *Annals of Mathematical Sciences and Applications*, 7(1):23–48, 2022.

Huang, Y., Hao, W., and Lin, G. Hompinns: Homotopy physics-informed neural networks for learning multiple solutions of nonlinear elliptic differential equations. *Computers & Mathematics with Applications*, 121:62–73, 2022. ISSN 0898-1221. doi:https://doi.org/10.1016/j.camwa.2022.07.002. URL https://www.sciencedirect.com/science/article/pii/S0898122122002851.

Kamilov, U. S., Mansour, H., and Wohlberg, B. A Plug-and-Play Priors Approach for Solving Nonlinear Imaging Inverse Problems. *IEEE Signal Processing Let-*

ters, 24(12):1872–1876, December 2017. ISSN 1558-2361. doi:10.1109/LSP.2017.2763583. Conference Name: IEEE Signal Processing Letters.

Khoo, Y. and Ying, L. SwitchNet: A Neural Network Model for Forward and Inverse Scattering Problems. *SIAM Journal on Scientific Computing*, 41(5):A3182–A3201, January 2019. ISSN 1064-8275. doi:10.1137/18M1222399. URL https://epubs.siam.org/doi/abs/10.1137/18M1222399. Publisher: Society for Industrial and Applied Mathematics.

Krishnapriyan, A., Gholami, A., Zhe, S., Kirby, R., and Mahoney, M. W. Characterizing possible failure modes in physics-informed neural networks. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P. S., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 26548–26560. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/df438e5206f31600e6ae4af72f2725f1-Paper.pdf.

Li, M., Demanet, L., and Zepeda-Núñez, L. Accurate and Robust Deep Learning Framework for Solving Wave-Based Inverse Problems in the Super-Resolution Regime, June 2021a. URL http://arxiv.org/abs/2106.01143. arXiv:2106.01143 [cs, math, stat].

Li, M., Demanet, L., and Zepeda-Núñez, L. Wide-Band Butterfly Network: Stable and Efficient Inversion Via Multi-Frequency Neural Networks. *Multiscale Modeling & Simulation*, 20(4):1191–1227, December 2022. ISSN 1540-3459. doi:10.1137/20M1383276. URL https://epubs.siam.org/doi/abs/10.1137/20M1383276. Publisher: Society for Industrial and Applied Mathematics.

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*, 2021b. URL https://openreview.net/forum?id=c8P9NQVtmnO.

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, March 2021. ISSN 2522-5839. doi:10.1038/s42256-021-00302-5.

Monga, V., Li, Y., and Eldar, Y. C. Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing. *IEEE Signal Processing Magazine*, 38(2):18–44, 2021.

Natterer, F. *The Mathematics of Computerized Tomography*. Society for Industrial and Applied Mathematics, 2001. doi:10.1137/1.9780898719284.

Ong, Y. Z., Shen, Z., and Yang, H. Integral autoencoder network for discretization-invariant learning. *Journal of Machine Learning Research*, 23(286):1–45, 2022. ISSN 1533-7928.

Ongie, G., Jalal, A., Metzler, C. A., Baraniuk, R. G., Dimakis, A. G., and Willett, R. Deep Learning Techniques for Inverse Problems in Imaging. *IEEE Journal on Selected Areas in Information Theory*, 1(1):39–56, May 2020. ISSN 2641-8770. doi:10.1109/JSAIT.2020.2991563. Conference Name: IEEE Journal on Selected Areas in Information Theory.

Potapczynski, A., Finzi, M., Pleiss, G., and Wilson, A. G. CoLA: Exploiting Compositional Structure for Automatic and Efficient Numerical Linear Algebra. *arXiv preprint arXiv:2309.03060*, 2023.

Saad, Y. and Schultz, M. H. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7(3):856–869, 1986. doi:10.1137/0907058. URL https://doi.org/10.1137/0907058.

Venkatakrishnan, S. V., Bouman, C. A., and Wohlberg, B. Plug-and-play priors for model based reconstruction. In *2013 IEEE Global Conference on Signal and Information Processing*, pp. 945–948, 2013. doi:10.1109/GlobalSIP.2013.6737048.

Watson, L. T. and Haftka, R. T. Modern homotopy methods in optimization. *Computer Methods in Applied Mechanics and Engineering*, 74(3):289–305, September 1989. ISSN 0045-7825. doi:10.1016/0045-7825(89)90053-4. URL https://www.sciencedirect.com/science/article/pii/0045782589900534.

Wei, Z. and Chen, X. Deep-learning schemes for full-wave nonlinear inverse scattering problems. *IEEE Transactions on Geoscience and Remote Sensing*, 57(4):1849–1860, April 2019. ISSN 1558-0644. doi:10.1109/TGRS.2018.2869221.

Zhang, B., Zepeda-Núñez, L., and Li, Q. Solving the Wide-band Inverse Scattering Problem via Equivariant Neural Networks, October 2023. URL http://arxiv.org/abs/2212.06068. arXiv:2212.06068 [cs, math].

Zhao, Q., Ma, Y., Boufounos, P., Nabi, S., and Mansour, H. Deep born operator learning for reflection tomographic imaging. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, 2023. doi:10.1109/ICASSP49357.2023.10095494.

Zhou, M., Han, J., Rachh, M., and Borges, C. A neural network warm-start approach for the inverse acoustic obstacle scattering problem. *Journal of Computational Physics*, 490:112341, 2023. ISSN 0021-9991. doi:https://doi.org/10.1016/j.jcp.2023.112341. URL https://www.sciencedirect.com/science/article/pii/S0021999123004369.

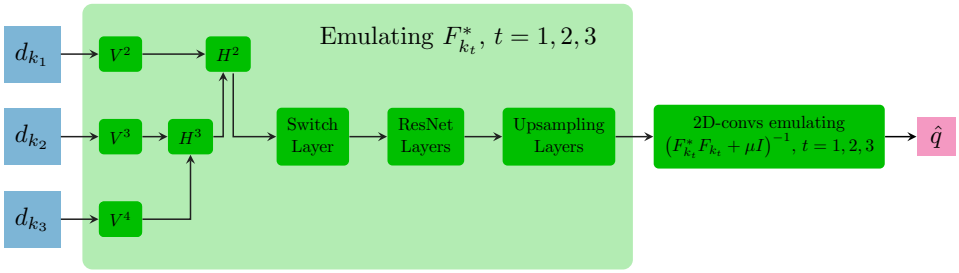


Figure 5: A diagram of Wide-Band Butterfly Network, adapted from Li et al. (2021a; 2022). All blocks in dark green contain trainable parameters.

A. Architecture Details

A.1. FYNet

We implement the inversion network described in (Fan & Ying, 2022) and call it FYNet. This method suggests applying the far-field scaling and a transformation from the receiver and source direction coordinates (r, s) to a new set of variables as summarized in (Fan & Ying, 2022, Equation (6)), which we perform using bicubic interpolation. We split the complex-valued input into real and imaginary parts along a channel dimension. To implement the action of the adjoint operator F_k^*, we use a composition of three 1-dimensional convolutional layers. We implement the convolutional layers as learnable in the Fourier domain because the expression for F_k^* derived in (Fan & Ying, 2022) is local in frequency, but not space. To implement the filtering operator $(F_k^* F_k + \mu I)^{-1}$, we use a composition of three 2-dimensional convolutional layers. All layers but the last one use ReLU activations. This network outputs an estimate of the scattering potential on a regular polar grid, in coordinates (ρ, ϕ). Following (Fan & Ying, 2022), we train the network by minimizing the difference between predictions and targets on the polar grid. We transform to Cartesian coordinates for visualization and computing final test statistics.

A.2. Wide-Band Butterfly Network

The Wide-Band Butterfly Network is introduced and defined in Li et al. (2022; 2021a). This architecture jointly parameterizes the adjoint operators $F_{k_1}^*, \ldots, F_{k_{N_k}}^*$, but it leverages the complementary low-rank property of $F_{k_t}^*$ (Khoo & Ying, 2019) to hierarchically merge the data using a butterfly network. For this network, we use code provided by the authors.[1] The reference implementation is limited to using data at three incident frequencies, so we only include results in this setting.

B. Data Generation

B.1. Distribution of Scattering Potentials

We define a distribution of scattering potentials $\mathcal{D}$ which has nonzero spatial support on the disk of radius 0.4, with a smoothly varying random background occluded by three randomly placed and randomly sized piecewise-constant shapes. We normalize the scattering potential so the background has minimum and maximum values 0 and 2 respectively, and we normalize the piecewise-constant shapes to have value 2. Figure 3a shows one such scattering potential from our distribution.

Notably, the contrast of these scattering potentials $\|q\|_\infty = 2$, which is much larger than the contrast used in distributions to evaluate other machine learning methods in the shape reconstruction regime Fan & Ying (2022); Li et al. (2022). The high-contrast regime is an important experimental setting because it ensures the nonlinearity of the forward model, which is the difficult and interesting problem setting, is captured. The non-constant background also adds to the difficulty of the problem by increasing $\|q\|_2$, which adds to the nonlinearity of the forward model. It also adds much more entropy to $\mathcal{D}$. We use this model to reflect experimental conditions in imaging tasks, wherein backgrounds are rarely known, constant, or homogeneous.

To generate samples from $\mathcal{D}$, our distribution of scattering potentials, we draw a random smoothly-varying background and three shapes with random sizes, positions, and rotations. This section provides details about the generation of these scattering objects.

[1] https://github.com/borongzhang/ISP_baseline

The random low-frequency backgrounds were generated by drawing random Fourier coefficients and filtering out the high frequencies using $\mathsf{LPF}_{4.0}$. The resulting background was transformed to Cartesian coordinates, and then shifted and scaled so the maximum value was 2.0 and the minimum value was 0.0. The background was truncated to 0.0 outside of the disk of radius 0.4. Three shapes were randomly chosen among equilaterial triangles, squares, and ellipses. The three shapes had randomly-chosen centers and rotations, constrained to be non-overlapping and fit inside the disk of radius 0.4. The side lengths of the squares and triangles were uniformly sampled from $[0.1, 0.15]$. The major axis lengths of the ellipses were uniformly sampled from $[0.1, 0.15]$, and the minor axis lengths were uniformly sampled from $[0.05, 0.1]$. Finally, $\mathsf{LPF}_{32\pi}$ was applied to the scattering potential.

B.2. Implementation of $\mathcal{F}$

To implement the forward model, we implement a numerical PDE solver to compute solutions of (1a)-(1b). We implement this by discretizing the scattering domain Ω with a (N_q, N_q) regular grid, with $N_q = 192$. We transform (1a)-(1b) into the Lippmann-Schwinger integral equation and recast the latter as a sparse linear system, which we solve with an iterative method to approximate convergence. This formulation allows us to compute the solution u_{scat} on a large, distant ring placed at radius $R = 100$. We compute the solution at $N_r = 192$ equally-spaced positions on this ring, and we repeat this process for $N_s = 192$ equally-spaced source directions. We use a `Pytorch` implementation of GMRES (Saad & Schultz, 1986) through the `CoLA` library (Potapczynski et al., 2023) to facilitate batched computation over multiple source directions, fast sparse linear system solves, and GPU acceleration; altogether, evaluating the full forward model for a given q only takes about 5 seconds using our implementation, using one NVIDIA® A40 GPU. Figures 3b and 3c show the solution u_{scat} produced by our implementation for high and low incident wave frequencies.

We use bicubic interpolation to implement the data transformation described in (Fan & Ying, 2022, Equation (6)), resulting in a transformed grid of size $(N_m, N_h) = (192, 96)$ for the input data. The FYNet blocks reconstruct images on a regular polar grid with $(N_\rho, N_\phi) = (96, 192)$ pixels, and the radial dimension of our polar grid extends to $\rho_{\max} = 0.5$. Finally, we use bicubic interpolation to transform the model's outputs to the (N_q, N_q) Cartesian grid for final visualization and error measurement.

C. Hyperparameter Search

For our hyperparameter searches, we trained models on a grid of hyperparameters and evaluated them on a validation set every 5 epochs. We found the epoch and hyperparameter setting which produced the lowest error on the validation set, and used those model weights for final evaluation on a held-out test set.

C.1. MFISNet-Refinement and FYNet

We train all models using the Adam algorithm, with a batch size of 16 samples. The FYNet and MFISNet-Refinement models tested have 3 1D convolutional layers followed by 3 2D convolutional layers; ReLU activations are used between layers. We use 1D and 2D convolutional kernels with 24 channels following (Fan & Ying, 2022);. To train FYNet and MFISNet-Refinement, we search over a grid of architecture and optimization hyperparameters. We report the optimal hyperparameters we found in Tables 3 to 5.

1d kernel size This is the number of frequency components in the 1D convolutional filters emulating F_k^*. We search over values $\{20, 40, 60\}$.

2d kernel size This is the size (in pixels) of the 2D convolutional kernel used in the layers emulating $(F_k^* F_k + \mu I)^{-1}$. We search over values $\{5, 7\}$.

Weight decay The weight decay parameter adds an ℓ_2 weight regularization term to the loss function. This hyperparameter determines the coefficient of this regularization term. We search over values $\{0.0, 1 \times 10^{-3}\}$.

Learning rate This is the step size for the Adam optimization algorithm. We search over values $\{1 \times 10^{-4}, 5 \times 10^{-4}, 1 \times 10^{-3}\}$.

LR decrease For MFISNet-Refinement models only, we decrease the learning rate each time we begin training a new network block. This parameter determines the multiplicative decrease that we apply to the learning rate. We search over values $\{1.0, 0.25\}$.

Hyperparameter	Data Setting (FYNet) $N_k = 1$
1d kernel size	60
2d kernel size	7
Weight decay	1×10^{-3}
Learning Rate	1×10^{-3}

Table 3: Optimal hyperparameters for FYNet.

Hyperparameter	Data Setting (MFISNet-Refinement)	
	$N_k = 2$	$N_k = 3$
1d kernel size	20	20
2d kernel size	5	5
Weight decay	1×10^{-3}	0.0
Learning rate	5×10^{-4}	1×10^{-3}
LR decrease	0.25	0.25

Table 4: Optimal hyperparameters for MFISNet-Refinement.

Hyperparameter	Training Method		
	No Iterative Refinement	No Homotopy through Frequency	Our Method
1d kernel size	20	40	20
2d kernel size	5	7	5
Weight decay	0.0	1×10^{-3}	0.0
Learning rate	1×10^{-3}	1×10^{-3}	1×10^{-3}
LR decrease	N/A	N/A	0.25
γ	N/A	1.1	N/A

Table 5: Optimal hyperparameters for the MFISNet-Refinement models in the training method experiment (Section 5.2). Here, γ is the factor which weights different loss terms in the "No Homotopy through Frequency" condition (cf. (14)). We searched over values $\gamma = \{0.9, 1.0, 1.1\}$. All of these models were trained with incident frequencies $[8\pi, 16\pi, 32\pi]$ and $n = 3333$ training samples.

C.2. Wide-Band Butterfly Network

To find the optimal Wide-Band Butterfly Network, we optimized over the following hyperparameters. We defined the grid of hyperparameters by taking the original hyperparameter from Li et al. (2022) and adding both higher and lower values, where possible. See Table 6 for the selected values.

Rank This parameter controls the rank of the compression of local patches in the butterfly factorization. Increasing this rank parameter increases the number of learnable parameters in the part of the network emulating F_k^*. We found that increasing the rank decreased the train and validation errors, and we increased the rank until we were unable to fit the model and data onto a single GPU. We searched over values $\{2, 3, 5, 10, 15, 20, 30, 50\}$.

Initial Learning Rate We decrease the learning rate by a multiplicative factor after $2,000$ minibatches, as suggested by Li et al. (2022). This parameter is the initial learning rate for the optimization algorithm. We searched over values $\{5 \times 10^{-4}, 1 \times 10^{-3}, 5 \times 10^{-3}\}$.

Learning Rate Decay This is the multiplicative decay parameter for the learning rate schedule. We searched over values $\{0.85, 0.95\}$.

Sigma Li et al. (2022) suggest training the network to match slightly-filtered versions of the ground-truth q. This is performed by applying a Gaussian filter to the targets $q^{(i)}$ before training. Sigma is the standard deviation of this

Gaussian filter. We do not blur the targets in the test set. We searched over values $\{0.75, 1.125, 1.5\}$.

Batch Size This is the number of samples per minibatch. We searched over values $\{16, 32\}$.

Hyperparameter	**Data Setting (Wide-Band Butterfly Network)** $N_k = 3$
Rank	50
Initial Learning Rate	5×10^{-4}
Learning Rate Decay	0.85
Sigma	1.5
Batch Size	16

Table 6: Optimal hyperparameters for Wide-Band Butterfly Networks.

D. Additional empirical results

In this section, we illustrate the predictions generated by the different models used in our experiments on four randomly-selected test samples from our dataset. For each prediction, we include the associated error plot. The visualizations are provided in Figure 6.

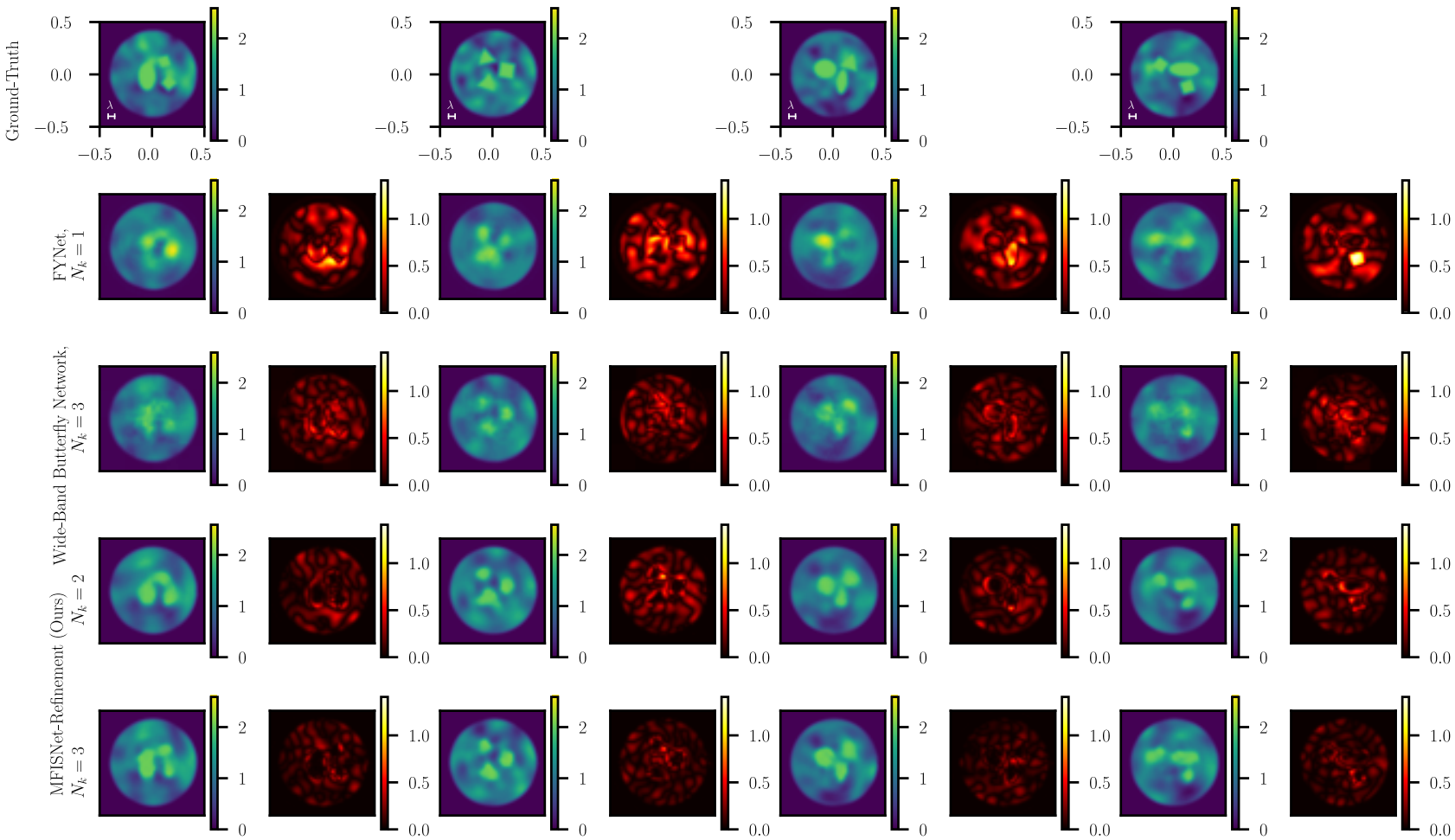


Figure 6: Sample predictions from four randomly-selected test samples.