SIAM J. IMAGING SCIENCES
Vol. 16, No. 4, pp. 1929–1952

A Data-Assisted Two-Stage Method for the Inverse Random Source Problem*

Peijun Li†, Ying Liang‡, and Yuliang Wang§

Abstract. We propose a data-assisted two-stage method for solving an inverse random source problem of the Helmholtz equation. In the first stage, the regularized Kaczmarz method is employed to generate initial approximations of the mean and variance based on the mild solution of the stochastic Helmholtz equation. A dataset is then obtained by sampling the approximate and corresponding true profiles from a certain a priori criterion. The second stage is formulated as an image-to-image translation problem, and several data-assisted approaches are utilized to handle the dataset and obtain enhanced reconstructions. Numerical experiments demonstrate that the data-assisted two-stage method provides satisfactory reconstruction for both homogeneous and inhomogeneous media with fewer realizations.

Key words. inverse source problem, Helmholtz equation, neural network, deep learning

MSC codes. 35R30, 35R60, 62M45, 78A46

DOI. 10.1137/23M1562561

1. Introduction.

Inverse source problems are of great importance in many fields, such as antenna synthesis, medical imaging, and earthquake monitoring. These problems involve determining the unknown sources that generate prescribed radiated wave patterns. Over the past four decades, they have been extensively studied. However, due to the presence of nonradiating sources, inverse source problems generally do not have unique solutions when only boundary measurements at a fixed frequency are used [9, 14]. In order to tackle the difficulty of these issues, attempts have been made to find the least-energy solution [31]. The use of multifrequency data has been employed to guarantee uniqueness and improve the stability of the problem [7].

In many applications, it is desirable to take into account the uncertainties that may arise from the unpredictability of the surrounding environment, incomplete information of the system, or random noise of the measurements. These inverse problems involving such uncertainties are called stochastic inverse problems. A number of theoretical studies for stochastic inverse scattering problems can be found in [16, 22, 3]. The inverse random source problem seeks to identify the mean and variance of random sources from the data they produce. In

*Received by the editors March 29, 2023; accepted for publication (in revised form) June 5, 2023; published electronically October 12, 2023.
https://doi.org/10.1137/23M1562561
Funding: The work of the first author was partially supported by the National Science Foundation (grant DMS-2208256).
†Corresponding author. Department of Mathematics, Purdue University, West Lafayette, IN 47907 USA (lipeijun@math.purdue.edu).
‡Department of Mathematics, Purdue University, West Lafayette, IN 47907 USA (liang402@purdue.edu).
§Research Center for Mathematics, Beijing Normal University, Zhuhai 519087, People's Republic of China, and BNU-HKBU United International College, Zhuhai 519807, People's Republic of China (yuliangwang@uic.edu.cn).

[6], a computational framework was proposed for the one-dimensional problem by using the inverse sine or cosine transform to reconstruct the mean and variance of the random source. For higher-dimensional problems, the regularized block Kaczmarz and eigenfunction-based methods were developed in [4] and [27] for the inverse source problem in a homogeneous and inhomogeneous medium, respectively. In [30], a stability analysis was carried out for the inverse random source problem of the one-dimensional stochastic Helmholtz equation in a homogeneous medium driven by white noise. Uniqueness was examined in [29, 28] for the inverse source problem where the random source was modeled as a generalized microlocally isotropic Gaussian random field. The existing numerical methods for the inverse random source problem have several drawbacks: They require a large number of realizations of the stochastic direct problem, the mean and variance of the source must be continuous for satisfactory reconstruction, and the medium is assumed to be either homogeneous or inhomogeneous but known. In practice, the inhomogeneous medium may be unknown, which limits the feasibility of existing numerical methods. Moreover, the accuracy of the reconstruction is affected by the number of realizations of the stochastic direct problem and the continuity of the mean and variance of the source.

Recently, machine learning techniques have been successfully applied to a wide range of scientific areas, particularly for solving classification and segmentation problems. Motivated by these successes, machine learning algorithms are being applied to solve inverse problems modeled by parametric PDEs. We briefly introduce two common existing strategies based on neural networks. The first strategy utilizes neural networks as surrogates for differential equations, such as the physics-informed neural network [11], the Deep Ritz method [40], and the weak adversarial neural network [41]. Algorithms are developed to search for unknown parameters in the PDE model and to obtain numerical solutions over the approximation space consisting of neural networks. However, retraining the neural network for a different problem governed by the same PDE is quite expensive. The second main strategy utilizes the training dataset in the form of paired observations and parameters to create a model that can be used to make predictions. In such algorithms, the observations are treated as input and the corresponding exact parameters as the output of a given neural network architecture. The neural network is trained to learn the mapping from the observations to the parameters, as demonstrated in [1, 43]. This strategy has been used for data-driven model discovery [10, 36] to reconstruct governing equations from observed data. Besides these two strategies, there is an emerging body of work on operator learning. In [13], the authors propose the operator recurrent neural network to directly learn the inverse operator between infinite-dimensional spaces. There has also been interest in learning regularizers for inversion, and a comprehensive account of this topic can be found in [2].

A variety of research endeavors have focused on developing machine learning algorithms to solve inverse scattering problems. For instance, [26] presents a neural network model featuring a SwitchNet architecture, designed to learn the direct and inverse maps between the scatterers and the scattered wave field. The study [38] offers another approach, where the authors blend the physical model of wave propagation into a deep neural network architecture to address the inverse problem associated with nonlinear wave equations. A different strategy is evident in [18], where a fully connected neural network is trained to establish the mapping between the limited-aperture radar cross-section data and the Fourier coefficients of the unidentified scatterers.

In this work, we introduce a data-assisted two-stage method tailored to address the inverse random source problem. The objective is to enable satisfactory reconstructions for discontinuous mean and variance of the random sources with fewer realizations, even within a fixed but unknown inhomogeneous medium. During the first stage, the regularized block Kaczmarz method is deployed to obtain initial approximations, leveraging the mild solution of the stochastic Helmholtz equation within a homogeneous medium. As we transition to the second stage, a variety of data-driven techniques are harnessed to learn the mapping from the approximations to the exact parameters, thereby enhancing prediction accuracy. This stage is thus structured as an image-to-image translation problem, with the goal of transferring images between source and target domains while maintaining content representations. Our approach is characterized by two salient features. First, the incorporation of partial physical information into the approximation from the initial stage reduces the computational load on the neural network during the subsequent stage. Second, the approximation matches the discretized statistics of the source in dimension, facilitating the application of various advanced machine learning algorithms for image-to-image translation problems. Similar concepts have been implemented in [42, 39, 12, 33] for deterministic inverse scattering problems and [20] for diffusive optical tomography.

Data-driven techniques have found wide applications in solving image-to-image translation problems, including in computer vision, medical imaging, and photo enhancement. In this work, we carry out a comparative study on the performance of four different such techniques in the second stage of the experiment. The first method utilizes principal component analysis (PCA) to reduce the dimensionality of the training data, followed by regression to produce the reconstructed statistics of the random source. The second method approximates the image-to-image translation problem as a linear dynamical system using the dynamic mode decomposition (DMD) approach [35]. The other two methods are based on convolutional neural networks. The first approach utilizes the U-Net architecture [34], a supervised learning approach originally designed for image segmentation. The second algorithm, pix2pix [25], is an advanced method for image-to-image translation based on a conditional generative adversarial neural network (cGAN). Numerical experiments have been conducted extensively to verify that these algorithms can effectively improve the reconstruction of the random source.

The paper is outlined as follows. In section 2, the mathematical model for the inverse random source problem is introduced. In section 3, we present a reconstruction scheme for the inverse problem based on integral equations and the regularized block Kaczmarz method and provide an error analysis of the method. In section 4, we detail the two-stage approach, which involves obtaining an approximation of the statistics using the reconstruction scheme in the first stage and enhancing it with data-driven techniques in the second stage. Finally, we present numerical examples to demonstrate the performance of the proposed method in section 5.

2. Problem statement.

Consider the two-dimensional Helmholtz equation driven by a random source

$$\Delta u(x,\kappa) + \kappa^2(1+\eta(x))u(x,\kappa) = f(x), \quad x \in \mathbb{R}^2, \tag{2.1}$$

where the wavenumber κ is a positive constant, η is a deterministic function representing the relative permittivity of the medium, and the random source f is assumed to be given by

$$f(x) = g(x) + h(x)\dot{W}_x$$

with g and $h \geq 0$ being two real-valued functions and $\dot{W}_x$ being the spatial white noise. Additionally, we assume that g and h are compactly supported in a rectangular domain $D \subset \mathbb{R}^2$, where $\overline{D} \subset B_R = \{x \in \mathbb{R}^2 : |x| < R\}$ for some $R > 0$ and η is compactly supported in B_R. As usual, the scattered field satisfies the Sommerfeld radiation condition

$$\lim_{r=|x|\to\infty} \sqrt{r}(\partial_r u - \mathrm{i}\kappa u) = 0. \tag{2.2}$$

There are two problems associated with the model (2.1)–(2.2). The direct problem is to study the radiated random wave field u for a given random source f. This work focuses on the inverse random source problem, which aims at determining the mean g and the variance h^2 of the random source f from the radiated wave field u that is measured on the boundary $\Gamma_R = \{x \in \mathbb{R}^2 : |x| = R\}$ at a discrete set of wavenumbers κ_j, $j = 1, 2, \ldots, m$.

Previous studies have focused on solving the inverse random source problem with homogeneous background media. Based on Green's function, [4] showed that the direct source problem has a unique mild solution, and a regularized block Kaczmarz method was developed for the inverse problem. When the background medium is inhomogeneous with a fixed η, the explicit Green function is no longer available. In this case, [27] examined the Lippman–Schwinger integral equation for the direct problem and tackled the inverse problem by analyzing the Dirichlet eigenvalue problem of the corresponding Helmholtz equation. However, this strategy depends on the assumption of an explicit form of η for their operation.

This study introduces an innovative data-driven approach suitable for solving the inverse random source problems mentioned previously for both homogeneous and inhomogeneous background media. It is worth mentioning that the solution procedure does not rely on the explicit form of η. The training process of the method consists of two stages, which we abstractly describe as follows:

Stage 1 The training dataset consists of M samples, denoted as $\{(p_i, Y_i)\}_{i=1}^M$, where each p_i represents multifrequency measurements of the wave field and each Y_i represents the corresponding statistics of the source for the ith sample. We apply a regularized block Kaczmarz method using measurement p_i, which will be further explained in section 3, leading to an initial approximation X_i.

Stage 2 Form a new dataset $\{(X_i, Y_i)\}_{i=1}^M$, and employ data-driven techniques to obtain a model χ that approximates the mapping from the initial approximation X_i in Stage 1 to the ground truth Y_i.

Having obtained the model χ in Stage 2, we can use it to reconstruct source statistics with any wave measurement p_{test} in two steps. First, by using the regularized Kaczmarz method as in the training process, we obtain an initial approximation X_{test} from the measurement data p_{test}. Second, we input X_{test} into the model χ to obtain a reconstruction $\chi(X_{\text{test}})$, which is expected to provide a better approximation of the exact statistics Y_{test} than the initial approximation X_{test}.

In this work, we generate the synthetic data by creating random samples of Y_i and solving the direct problem numerically for the measurement data p_i. The details for generating the training dataset can be found in section 5. Note that the function η is not a prerequisite in the Kaczmarz method in Stage 1 or the training process in Stage 2. Its necessity arises only during the generation of the training dataset, which is a result of acquiring synthetic data through numerical simulation. In a practical setting, it might be feasible to procure the dataset via physical measurements, where the η can be fixed, but its form is not required throughout the process. This is a notable aspect of our method, as existing methods typically rely on a known form of η.

3. Stage 1: The Kaczmarz method.

Stage 1 is to generate an initial image of the reconstruction. In this section, we briefly introduce the mild solution to the direct source problem of the stochastic Helmholtz equation in a homogeneous medium and the regularized block Kaczmarz method for solving the inverse random source problem. The details can be found in [4].

3.1. Integral equations.

Following [4, Hypothesis 2.4], we assume that $g \in L^2(D)$, $h \in C^{0,\alpha}(D)$, where $\alpha \in (0,1]$, and $h \in L^\beta(D)$, where $\beta \in (\beta_0, \infty]$ if $\frac{3}{2} \le \beta_0 \le 2$ or $\beta \in (\beta_0, \frac{3\beta_0}{3-2\beta_0})$ if $1 \le \beta_0 \le \frac{3}{2}$. It is shown in [4, Theorem 2.7] that the direct source problem (2.1)–(2.2) admits a unique mild solution

$$u(x,\kappa) = \int_D G(x,y,\kappa)g(y)dy + \int_D G(x,y,\kappa)h(y)dW_y \quad \text{a.s.}, \tag{3.1}$$

where $G(x,y,\kappa) = -\frac{\mathrm{i}}{4}H_0^{(1)}(\kappa|x-y|)$ is Green's function of the two-dimensional Helmholtz equation with $H_0^{(1)}$ being the Hankel function of the first kind with order zero.

Taking the expectation on both sides of (3.1), we get

$$\mathbb{E}(u(x,\kappa_j)) = \int_D G(x,y,\kappa_j)g(y)dy.$$

Splitting the above equation into the real and imaginary parts leads to

$$\mathbb{E}(\Re u(x,\kappa_j)) = \int_D \Re G(x,y,\kappa_j)g(y)dy, \tag{3.2}$$

$$\mathbb{E}(\Im u(x,\kappa_j)) = \int_D \Im G(x,y,\kappa_j)g(y)dy. \tag{3.3}$$

The mean g of the source f can be obtained by solving either (3.2) or (3.3). We use (3.2) to present the results in the numerical experiments.

Taking the variance on both sides of the real and imaginary parts of (3.1) yields

$$\mathbb{V}(\Re u(x,\kappa_j)) = \int_D |\Re G(x,y,\kappa_j)|^2 h^2(y)dy, \tag{3.4}$$

$$\mathbb{V}(\Im u(x,\kappa_j)) = \int_D |\Im G(x,y,\kappa_j)|^2 h^2(y)dy. \tag{3.5}$$

Subtracting (3.5) from (3.4), we obtain

$$\mathbb{V}(\Re u(x,\kappa_j)) - \mathbb{V}(\Im u(x,\kappa_j)) = \int_D \left(|\Re G(x,y,\kappa_j)|^2 - |\Im G(x,y,\kappa_j)|^2\right) h^2(y)dy. \tag{3.6}$$

In accordance with [4], (3.6) yields better reconstruction results than (3.4) or (3.5) since the singular values of the discretized integral kernel in (3.6) decay more slowly than those in (3.4) or (3.5). Therefore, (3.6) is adopted to reconstruct h^2 in our numerical experiments.

Neglecting the discretization error, the matrix equations corresponding to integral equations (3.2) and (3.6) can be formally expressed as

$$A_j q = p_j, \quad j = 1, \ldots, m, \tag{3.7}$$

where p_j denotes the data $\mathbb{E}(\Re u(x,\kappa_j))$ or $\mathbb{V}(\Re u(x,\kappa_j)) - \mathbb{V}(\Im u(x,\kappa_j))$ at measurement points on Γ_R, q stands for the unknown statistics g or h^2 taken at sampling points in D, and A_j represents the matrix arising from discretizing the integral kernels in (3.2) and (3.6). To solve the ill-posed equation (3.7), a regularized block Kaczmarz method is introduced in [4]: Let $q_\gamma^0 = 0$,

$$\begin{cases} q_0 = q_\gamma^k, \\ q_j = q_{j-1} + A_j^*(\gamma I + A_j A_j^*)^{-1}(p_j - A_j q_{j-1}), \quad j = 1, \ldots, m, \\ q_\gamma^{k+1} = q_m, \end{cases} \tag{3.8}$$

where $k = 0, 1, \ldots$ is the iteration index of the outer loop, A_j^* denotes the conjugate transpose of A_j, $\gamma > 0$ is the regularization parameter, and I is the identity matrix.

Combined with multifrequency data, the regularized Kaczmarz method effectively provides reconstructions for the statistics of the random source. However, the method has several limitations:

1. Accurately approximating the mean or variance using their corresponding averages in (3.2) or (3.6) typically requires a large number of trials. In practice, only a limited number of realizations are available, leading to significant errors in the approximation on the left-hand side of (3.2) or (3.6). Figure 1 shows the reconstructions of the mean of a random source using different numbers of realizations. The accuracy of the reconstruction improves as more realizations are used for the reconstruction.
2. The sequence generated by the Kaczmarz method typically converges to a minimum norm solution of the linear system as demonstrated by [32]. Consequently, it is difficult for the method to reconstruct a random source with discontinuous statistics, particularly for the regularized method. Figure 1 displays a random source with a piecewise constant mean. Although increasing the number of realizations improves the result, accurately capturing the edges of the true mean remains challenging.
3. The mild solution (3.1) and the Fredholm integral equations (3.2)–(3.6) are based on the assumption that the medium is homogeneous. Consequently, this method is inapplicable when the medium is inhomogeneous. In this case, [27] suggested an approach utilizing the corresponding Dirichlet eigenvalue problem of the Helmholtz equation. Nonetheless, this strategy falls short when dealing with complex media and

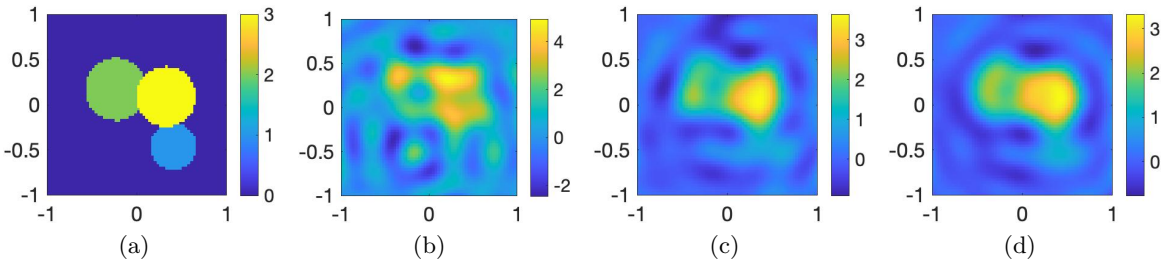


Figure 1. *Reconstructions with different numbers of trials:* (a) *the exact mean of a random source and* (b/c/d) *the reconstruction of mean with measurements of* 10/100/1000 *realizations.*

can become exceedingly time consuming when higher-order eigenfunctions are required to depict sources exhibiting higher-frequency modes. Crucially, the technique becomes inapplicable once more when the function η remains unidentified.

In this work, we adopt the regularized Kaczmarz method (3.8) as the first stage to generate an initial image of the reconstruction, assuming a homogeneous background medium for both homogeneous and inhomogeneous cases. Subsequently, the initial approximation is then passed to the second stage of the image-to-image translation problem, which can be effectively handled by data-driven techniques. When the medium is inhomogeneous, the information of the fixed but unknown function η is encoded in the mapping from the initial approximation to the ground truth and is thereby implicitly learned by the algorithms.

3.2. Error analysis. In this section, we present an error analysis for the regularized Kaczmarz method (3.8). The total error is composed of two parts: The expectation or variance in the expression of data p_j must be approximated by the sample mean, and the measurement of the wave field $u(x, \kappa_j)$ may be polluted by random noise. In [5], the convergence of the regularized Kaczmarz method (3.8) was shown for noise-free data. However, since data always contain noise in practice, an accumulated error may be generated during the iteration, leading to the semiconvergence phenomenon, which was examined in [15] for the Kaczmarz method. For a comprehensive account of the general Kaczmarz method for solving a linear system of equations, we refer the reader to [32].

We provide a theoretical analysis of the semiconvergence property of the regularized block Kaczmarz method. Recall that the integral equation (3.2) or (3.6) for the reconstruction of g or h^2 can be written as a matrix equation system (3.7):

$$A_j q = p_j, \quad j = 1, \ldots, m.$$

The system (3.7) is equivalent to

$$Aq = p,$$

where

$$A = \begin{bmatrix} A_1 \\ \vdots \\ A_m \end{bmatrix}, \quad p = \begin{bmatrix} p_1 \\ \vdots \\ p_m \end{bmatrix}.$$

In the inverse source problem, there are usually fewer measurement points compared to the number of sampling points of the solution. As a result, it is reasonable to assume that the matrix A has full row rank. Let $A^\dagger$ denote the pseudoinverse of A and $q^\dagger = A^\dagger p$ denote the minimal norm solution of the inverse source problem with the noise-free data p. Let $\tilde{p} = [\tilde{p}_1^*, \tilde{p}_2^*, \ldots, \tilde{p}_m^*]^*$ denote the perturbed data and q_γ^k, $\tilde{q}_\gamma^k$ denote the outputs of the scheme (3.8) with p and $\tilde{p}$ as inputs, respectively, after k iterations.

The error between the minimal norm solution $q^\dagger$ and the kth iteration $\tilde{q}_\gamma^k$ can be decomposed into two terms:

$$\tilde{q}_\gamma^k - q^\dagger = (\tilde{q}_\gamma^k - q_\gamma^k) + (q_\gamma^k - q^\dagger).$$

According to [5, Theorem 3.7], we have $q_\gamma^k - q^\dagger \to 0$ as $k \to \infty$ and $\gamma \to 0$. In the following, we analyze $\tilde{q}_\gamma^k - q_\gamma^k$, which represents the data error accumulated to the kth iteration. To this end, we decompose

$$\gamma I + AA^* = \begin{bmatrix} \gamma I + A_1 A_1^* & \cdots & A_1 A_m^* \\ \vdots & \ddots & \vdots \\ A_m A_1^* & \cdots & \gamma I + A_m A_m^* \end{bmatrix} = D_\gamma + L + L^*,$$

where the matrices D_γ and L are given by

$$D_\gamma = \begin{bmatrix} \gamma I + A_1 A_1^* & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \gamma I + A_m A_m^* \end{bmatrix}, \quad L = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ A_2 A_1^* & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ A_m A_1^* & \cdots & A_m A_{m-1}^* & 0 \end{bmatrix}.$$

Let

$$M_\gamma = (D_\gamma + L)^{-1}, \quad Q_\gamma = A^* M_\gamma A, \quad B_\gamma = I - Q_\gamma. \tag{3.9}$$

The regularized block Kaczmarz method (3.8) can be equivalently formulated as

$$q_\gamma^{k+1} = B_\gamma q_\gamma^k + A^* M_\gamma p. \tag{3.10}$$

To estimate the error term $\tilde{q}_\gamma^k - q_\gamma^k$, we define

$$\tilde{B}_\gamma = B_\gamma P, \quad R_\gamma = (D_{2\gamma})^{1/2} M_\gamma A, \tag{3.11}$$

where P is the orthogonal projection onto the range space of A^*, i.e., $\mathcal{R}(A^*)$, and $D_{2\gamma}$denotes the matrix $D_\gamma$with γ being replaced by 2γ.

Lemma 3.1. *Let σ_γ be the smallest nonzero singular value of R_γ. Then it holds that $\sigma_\gamma \leq 1$and*

$$\|\tilde{B}_\gamma\| = \sqrt{1 - \sigma_\gamma^2}.$$

Proof. From (3.9), we can see that

$$B_\gamma^* B_\gamma = (I - Q_\gamma)^*(I - Q_\gamma) = I - A^*(M_\gamma + M_\gamma^*)A + Q_\gamma^* Q_\gamma.$$

By substituting the expression of Q_γ and performing a straightforward calculation, we obtain

$$\begin{aligned} Q_\gamma^* Q_\gamma &= A^* M_\gamma^* A A^* M_\gamma A = A^* M_\gamma^* (D_\gamma + L + L^* - \gamma I) M_\gamma A \\ &= A^* M_\gamma^* [(D_\gamma + L) + (D_\gamma + L^*) - D_{2\gamma}] M_\gamma A \\ &= A^* M_\gamma^* \left[M_\gamma^{-1} + (M_\gamma^*)^{-1} - D_{2\gamma} \right] M_\gamma A \\ &= A^*(M_\gamma + M_\gamma^*)A - A^* M_\gamma^* D_{2\gamma} M_\gamma A. \end{aligned}$$

Combining the above equations with the expression in (3.11) of R_γ, we obtain

$$B_\gamma^* B_\gamma = I - A^* M_\gamma^* D_{2\gamma} M_\gamma A = I - R_\gamma^* R_\gamma,$$

which then implies that $\sigma_\gamma \le 1$.

A simple calculation shows that

$$\tilde{B}_\gamma^* \tilde{B}_\gamma = P(I - R_\gamma^* R_\gamma)P = P - P R_\gamma^* R_\gamma P = P - R_\gamma^* R_\gamma P = (I - R_\gamma^* R_\gamma)P.$$

Let $U\Sigma V^*$ be the compact singular value decomposition (SVD) of R_γ. Then we have $P = VV^*$ and

$$\tilde{B}_\gamma^* \tilde{B}_\gamma = (I - V\Sigma^2 V^*)VV^* = VV^* - V\Sigma^2 V^* = V(I - \Sigma^2)V^*.$$

Letting $\lambda_{\max}$ denote the largest singular value of a square matrix, we get

$$\|\tilde{B}_\gamma\| = \sqrt{\lambda_{\max}(\tilde{B}_\gamma^* \tilde{B}_\gamma)} = \sqrt{1 - \sigma_\gamma^2},$$

which completes the proof. ∎

The following result is concerned with the accumulated data error $\tilde{q}_\gamma^k - q_\gamma^k$ of the regularized Kaczmarz method.

Theorem 3.2. *Let $e^k = \tilde{q}_\gamma^k - q_\gamma^k$. Then*

$$\|e^k\| \le \left[\frac{1 - (1 - \sigma_\gamma^2)^{\frac{k}{2}}}{1 - (1 - \sigma_\gamma^2)^{\frac{1}{2}}} \right] \|A^* M_\gamma (\tilde{p} - p)\|. \tag{3.12}$$

Proof. Using (3.10), we can write

$$e^{k+1} = B_\gamma e^k + A^* M_\gamma (\tilde{p} - p),$$

which leads to

$$e^k = \left[\sum_{i=0}^{k-1} (B_\gamma)^i \right] A^* M_\gamma (\tilde{p} - p). \tag{3.13}$$

Since $\mathcal{R}(A^*)$ is an invariant subspace of B_γ, we can further deduce from (3.13) that

$$e^k = \left[\sum_{i=0}^{k-1}(\tilde{B}_\gamma)^i\right] A^* M_\gamma(\tilde{p}-p). \tag{3.14}$$

Let $S_\gamma^k = \sum_{i=0}^{k-1}(\tilde{B}_\gamma)^i$. Utilizing Lemma 3.1, we have

$$\|S_\gamma^k\| \leqslant \sum_{i=0}^{k-1}\|\tilde{B}_\gamma\|^i = \frac{1-\|\tilde{B}_\gamma\|^k}{1-\|\tilde{B}_\gamma\|} = \frac{1-(1-\sigma_\gamma^2)^{\frac{k}{2}}}{1-(1-\sigma_\gamma^2)^{\frac{1}{2}}}.$$

Combining the above inequality with (3.14), we obtain the estimate given in (3.12). ∎

The following corollary further provides an upper bound of the accumulated data error.

Corollary 3.3. *The error $e^k = \tilde{q}_\gamma^k - q_\gamma^k$ in the regularized block Kaczmarz algorithm satisfies the following estimates:*

(i) *if $2 \leq k \leq 2\sigma_\gamma^{-2}$, then*

$$\|e^k\| \leq \frac{\delta_\gamma \sigma_\gamma^2}{2(1-(1-\sigma_\gamma^2)^{\frac{1}{2}})}k; \tag{3.15}$$

(ii) *if $k > 2\sigma_\gamma^{-2}$, then*

$$\|e^k\| \leq \frac{\delta_\gamma}{1-(1-\sigma_\gamma^2)^{\frac{1}{2}}}, \tag{3.16}$$

where $\delta_\gamma = \|A^* M_\gamma(\tilde{p}-p)\|$.

Proof. We start by introducing the function

$$\Psi(\sigma,k) = \frac{1-(1-\sigma^2)^{\frac{k}{2}}}{1-(1-\sigma^2)^{\frac{1}{2}}}.$$

It follows from Lemma 3.1 that $0 < \sigma_\gamma \leq 1$. It is easy to verify that the function $\Psi(\sigma_\gamma, k)$ admits an upper bound for any k:

$$\Psi(\sigma_\gamma,k) \leq \frac{1}{1-(1-\sigma_\gamma^2)^{\frac{1}{2}}}. \tag{3.17}$$

Substituting (3.17) into (3.12) leads to the upper bound (3.16) for any k.

For $k \geq 2$, we observe that the function $1-(1-\sigma_\gamma^2)^{\frac{k}{2}}$ satisfies the estimate

$$1-(1-\sigma_\gamma^2)^{\frac{k}{2}} \leq \frac{k}{2}\sigma_\gamma^2.$$

This implies that

$$\Psi(\sigma_\gamma,k) \leq \frac{k\sigma_\gamma^2}{2(1-(1-\sigma_\gamma^2)^{\frac{1}{2}})}. \tag{3.18}$$

When $k \le 2\sigma_\gamma^{-2}$, the estimate in (3.18) can be further reduced to

$$\frac{k\sigma_\gamma^2}{2(1-(1-\sigma_\gamma^2)^{\frac{1}{2}})} \le \frac{1}{1-(1-\sigma_\gamma^2)^{\frac{1}{2}}}.$$

For $2 \le k \le 2\sigma_\gamma^{-2}$, substituting (3.18) into (3.12) yields a sharper upper bound (3.15). ∎

Remark 3.4. The above corollary indicates that the data error is bounded above by an increasing function of k when k is smaller than $2\sigma_\gamma^{-2}$, which is consistent with the observed accumulation of data errors during iteration in numerical experiments. In practice, the value of σ_γ^{-2} might be very large. Although it is unclear how to achieve a sharper estimate in Corollary 3.3, the result tracks the behavior of the data error.

In the following analysis, we examine the impact of data perturbations on the reconstruction process, taking the example of reconstructing the mean g. For simplicity, we use the same notation to represent the function and its discrete values obtained from measurements or sampling. Recall that the data p_j correspond to measurements of $\mathbb{E}(\Re u(x,\kappa_j))$. Due to data noise, the perturbed data $\tilde{p}_j$ are modeled by $\mathbb{E}_N(\Re\tilde{u}(x,\kappa_j))$, where N denotes the number of realizations, $\mathbb{E}_N$ stands for the sample mean, and $\tilde{u}(x,\kappa_j)$ represents the measurement of $u(x,\kappa_j)$ with additive noise. We use $\tilde{u}_j$ and u_j to denote $\tilde{u}(x,\kappa_j)$ and $\mathbb{E}(u(x,\kappa_j))$, respectively. It should be noted that the analysis for reconstructing h^2 follows a similar process.

By (3.1), the wave field u_j satisfies

$$u(x,\kappa_j) = \int_D G(x,y,\kappa_j)g(y)dy + \int_D G(x,y,\kappa_j)h(y)dW_y \quad \text{a.s.}$$

Neglecting the discretization error, we can express u_j and $\tilde{u}_j$ as

$$\Re u_j = A_j g, \quad \Re\tilde{u}_j = A_j(g + h\dot{W}_x) + r_j,$$

where $h \sim \mathcal{N}(0,H)$, $r_j \sim \mathcal{N}(0,\delta^2 I)$ with $H = \mathrm{diag}[h(x_i)^2]$ and δ indicating the noise level. We define

$$u_R = \begin{bmatrix} \Re u_1 \\ \vdots \\ \Re u_m \end{bmatrix}, \quad \tilde{u}_R = \begin{bmatrix} \Re\tilde{u}_1 \\ \vdots \\ \Re\tilde{u}_m \end{bmatrix}, \quad \text{and} \quad r = \begin{bmatrix} r_1 \\ \vdots \\ r_m \end{bmatrix}.$$

We further assume that the components of h, $r_1, r_2, \ldots, r_m$ are independent, leading to

$$\tilde{u}_R \sim \mathcal{N}(Ag, AHA^* + \delta^2 I).$$

Thus, we obtain

$$\tilde{u}_R - u_R \sim \mathcal{N}(0,\Theta),$$

where $\Theta = AHA^* + \delta^2 I$. Moreover, from the definition of $\tilde{p}$ and the central limit theorem, we deduce that

$$\sqrt{N}(\tilde{p} - p) \sim \mathcal{N}(0,\Theta).$$

The following corollary provides an estimate for the accumulated data error in the distribution sense and is a direct consequence of Theorem 3.2.

Corollary 3.5. *Let* $e^k = \tilde{q}_\gamma^k - q_\gamma^k$. *Then*

$$\sqrt{N} e^k \xrightarrow{d} \mathcal{N}(0, \Lambda_k),$$

where

$$\|\Lambda_k\| \leq \left[\frac{1-(1-\sigma_\gamma^2)^{\frac{k}{2}}}{1-(1-\sigma_\gamma^2)^{\frac{1}{2}}}\right]^2 \|A^* M_\gamma\|^2 \|\Theta\|.$$

These results demonstrate the dependence of the perturbation in the data and the variance of the accumulated data error on the number of realizations N and the noise level δ.

4. Stage 2: Data-driven techniques. In this section, we present four data-driven techniques as the second stage of our proposed method. Let $\{(X_i, Y_i)\}_{i=1}^M$ be the training dataset, where $X_i \in \mathbb{R}^{n\times n}$ represents a matrix computed using the measurement data corresponding to the ith sample from the first stage, $Y_i \in \mathbb{R}^{n\times n}$ denotes the exact mean or variance of the ith sample after discretization, and M is the total number of samples. The goal of the second stage is to learn the mapping from the approximations obtained in the first stage to the exact statistical properties by utilizing the training dataset. We present two methods in data analysis and two methods based on neural networks.

4.1. PCA. PCA is a well-established approach for dimensionality reduction in statistical analysis [23]. Recently, Bhattacharya et al. [8] combined PCA with neural network techniques to solve inverse problems of parametric PDEs. In our work, we build on the idea of PCA to reduce the dimension of input and output spaces and then apply linear regression as the second stage of our algorithm.

To prepare for training, we flatten each X_i into a vector $\mathbf{x}_i \in \mathbb{R}^{n^2}$ and construct the input data matrix $\mathbf{X} \in \mathbb{R}^{n^2\times M}$ by concatenating $\mathbf{x}_i$ as columns. Similarly, we construct the output data matrix $\mathbf{Y} \in \mathbb{R}^{n^2\times M}$ with the flattened Y_i, denoted by $\mathbf{y}_i \in \mathbb{R}^{n^2}$. After concatenating the input and output data matrices, we perform mean subtraction and use $\mathbf{X}$ and $\mathbf{Y}$ to denote the data matrices after the mean centering step for convenience. Then we utilize SVD to derive the low-dimensional representation of the data. Specifically, we first implement SVD

$$\mathbf{E} := \begin{bmatrix}\mathbf{X}\\ \mathbf{Y}\end{bmatrix} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*,$$

where $\mathbf{U} \in \mathbb{R}^{2n^2\times 2n^2}$ and $\mathbf{V} \in \mathbb{R}^{M\times M}$ are orthonormal matrices and $\mathbf{\Sigma} \in \mathbb{R}^{2n^2\times M}$ is the diagonal singular value matrix. Then we keep the first s_{PCA} principal components to form a low-rank approximation of the data matrix $\mathbf{E}$. Let $\mathbf{\Psi}_{s_{\mathrm{PCA}}} = [\mathbf{u}_1\ \mathbf{u}_2\ \ldots\ \mathbf{u}_{s_{\mathrm{PCA}}}]$, where $\mathbf{u}_i$ denotes the ith column of $\mathbf{U}$. From the Eckart–Young theorem, we conclude that $\mathbf{\Psi}_{s_{\mathrm{PCA}}}$ minimizes

$$\|\mathbf{E} - \mathbf{\Psi}\mathbf{\Psi}^*\mathbf{E}\|_F$$

over all orthonormal $\mathbf{\Psi} \in \mathbb{R}^{2n^2\times s_{\mathrm{PCA}}}$, where $\|\cdot\|_F$ is the Frobenius norm.

To predict reconstructions in the testing process, we use linear regression with the selected principal components. Denote

$$\mathbf{\Psi}_{s_{\mathrm{PCA}}} = \begin{bmatrix} \mathbf{\Psi}_X \\ \mathbf{\Psi}_Y \end{bmatrix},$$

where $\mathbf{\Psi}_X, \mathbf{\Psi}_Y \in \mathbb{R}^{n^2 \times s_{\mathrm{PCA}}}$. Given a matrix X_{test} obtained from the first stage, we flatten it into a vector $\mathbf{x}_{\mathrm{test}} \in \mathbb{R}^{n^2}$ and compute the corresponding output $\mathbf{y}_{\mathrm{test}}$ as

$$\mathbf{y}_{\mathrm{test}} = \mathbf{\Psi}_Y \mathbf{\Psi}_X^{\dagger} \mathbf{x}_{\mathrm{test}},$$

where $\mathbf{\Psi}_X^{\dagger}$ denotes the pseudoinverse of $\mathbf{\Psi}_X$. We finally obtain the prediction Y_{test} by reshaping $\mathbf{y}_{\mathrm{test}}$ into an $n \times n$ matrix.

4.2. DMD. DMD is a powerful technique for dimensionality reduction of dynamic systems [35]. By casting the mapping between the approximation and the exact statistical properties as a dynamical system, DMD can be leveraged to linearly approximate the inverse mapping for our problem.

We assume that the columns of data matrices $\mathbf{X}$ and $\mathbf{Y}$ obey a dynamical system of the form $\mathbf{y}_i = F(\mathbf{x}_i)$. The objective of DMD is to approximate this dynamic system using a linear operator $\mathbf{Y} = \mathbf{A}\mathbf{X}$. To this end, we employ the truncated SVD for matrix $\mathbf{X}$ to obtain a rank-s_{DMD} low-dimensional approximation of the form $\mathbf{X}_{s_{\mathrm{DMD}}} = \mathbf{U}_{s_{\mathrm{DMD}}} \mathbf{\Sigma}_{s_{\mathrm{DMD}}} \mathbf{V}^*_{s_{\mathrm{DMD}}}$, where $\mathbf{U}_{s_{\mathrm{DMD}}} \in \mathbb{R}^{n^2 \times s_{\mathrm{DMD}}}$, $\mathbf{V}_{s_{\mathrm{DMD}}} \in \mathbb{R}^{M \times s_{\mathrm{DMD}}}$, and $\mathbf{\Sigma}_{s_{\mathrm{DMD}}} \in \mathbb{R}^{s_{\mathrm{DMD}} \times s_{\mathrm{DMD}}}$. The optimal linear transformation $\mathbf{A}_{s_{\mathrm{DMD}}}$ is then computed from the reduced dimension-s_{DMD} coefficient space as

$$\mathbf{A}_{s_{\mathrm{DMD}}} = \mathbf{U}^*_{s_{\mathrm{DMD}}} \mathbf{Y} \mathbf{V}_{s_{\mathrm{DMD}}} \mathbf{\Sigma}^{-1}_{s_{\mathrm{DMD}}}.$$

Computing the eigendecomposition of matrix $\mathbf{A}_{s_{\mathrm{DMD}}}$ leads to an approximation of the spectrum of the underlying Koopman operator of the system.

To predict outputs using the linearized system, we take a matrix X_{test} generated in the first stage and flatten it to obtain $\mathbf{x}_{\mathrm{test}}$. Subsequently, the flattened prediction $\mathbf{y}_{\mathrm{test}}$ is computed via

$$\mathbf{y}_{\mathrm{test}} = \mathbf{U}_{s_{\mathrm{DMD}}} \mathbf{A}_{s_{\mathrm{DMD}}} \mathbf{U}^*_{s_{\mathrm{DMD}}} \mathbf{x}_{\mathrm{test}} = \mathbf{Y} \mathbf{V}_{s_{\mathrm{DMD}}} \mathbf{\Sigma}^{-1}_{s_{\mathrm{DMD}}} \mathbf{U}^*_{s_{\mathrm{DMD}}} \mathbf{x}_{\mathrm{test}}.$$

Finally, we obtain the prediction Y_{test} by reshaping $\mathbf{y}_{\mathrm{test}}$ into an $n \times n$ matrix.

4.3. U-Net–based network. The disparity between the reconstructed profile from Stage 1, exhibiting a smooth image, and the true sharp profile of the source (as shown in Figure 1) presents an opportunity to leverage image segmentation tools for learning the mapping between them. In this regard, we test the applicability of the U-Net convolutional neural network [34], which offers a unique encoder-decoder architecture that delivers highly accurate image segmentation. As shown in Figure 2, the U-Net approach preserves features of the input images with skip connections. The architecture of the neural network consists of a contracting path and an expansive path. The contracting path involves sequentially applying a convolution, followed by a batch normalization layer and a ReLU activation function, to

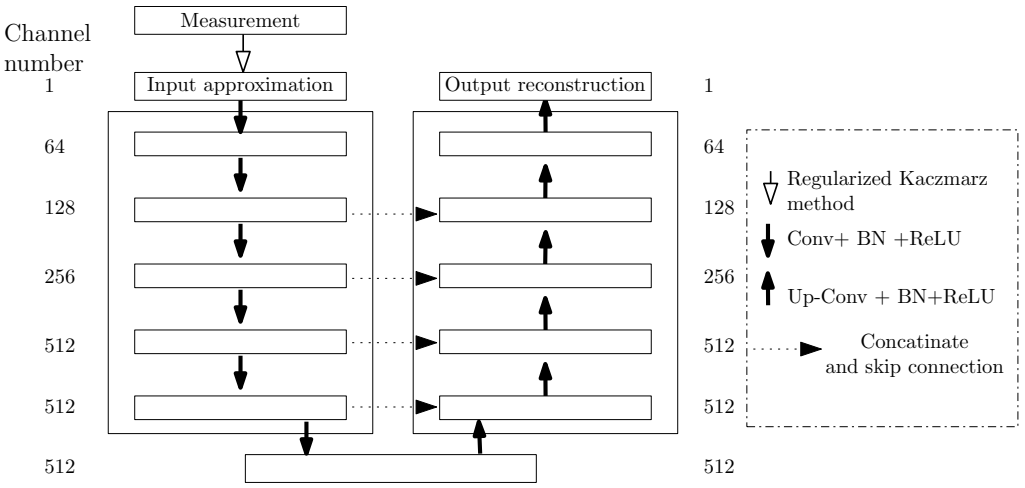


Figure 2. *The U-Net architecture.*

perform downsampling. The expansive path is comprised of three components: an upsampling of the feature map, a batch normalization layer, and a ReLU activation function. The first component is followed by an up-convolution that halves the number of feature channels, and the third component is used to restore the image to the size of the input image. For each convolutional layer in the contracting path, the feature maps are passed to the corresponding layers in the expansive path via the skip connections.

In the training process of the U-Net model χ, we adopt the L^1 error as the loss function; i.e., we minimize the following functional over the parameter θ of the neural network model χ:

$$\mathcal{L}_{L^1}(\chi) = \mathbb{E}(\|Y - \chi(X;\theta)\|_{L^1}),$$

where $\mathbb{E}$ denotes the expectation of the underlying distribution from which the training dataset is sampled. In practice, the expectation is approximated by taking the average value of the loss function over a batch of training examples. Empirical results suggest that this choice of L^1 error yields more favorable performance as compared to the mean squared error.

Remark 4.1. If we consider the first stage of the overall method as a layer of the neural network architecture, it may be regarded as a deconvolution layer that incorporates a priori understanding of the physical characteristics.

4.4. Pix2pix neural network. As an alternative to the U-Net model, we also evaluated the pix2pix neural network architecture. The pix2pix structure belongs to the family of cGANs, comprised of a discriminator network $\mathcal{D}$ and a generator network $\mathcal{G}$, which cooperate in order to generate realistic reconstructions [25].

In our experiments, the generator network maps an input image, $X_i \in \mathbb{R}^{n\times n}$, derived from the first stage, to an associated reconstruction, $\mathcal{G}(X_i) \in \mathbb{R}^{n\times n}$. On the other hand, the discriminator network takes in a pair of images, $(X, Y) \in \mathbb{R}^{n\times n} \times \mathbb{R}^{n\times n}$, as input and produces a scalar-valued output ranging between 0 and 1. The two networks are jointly trained in

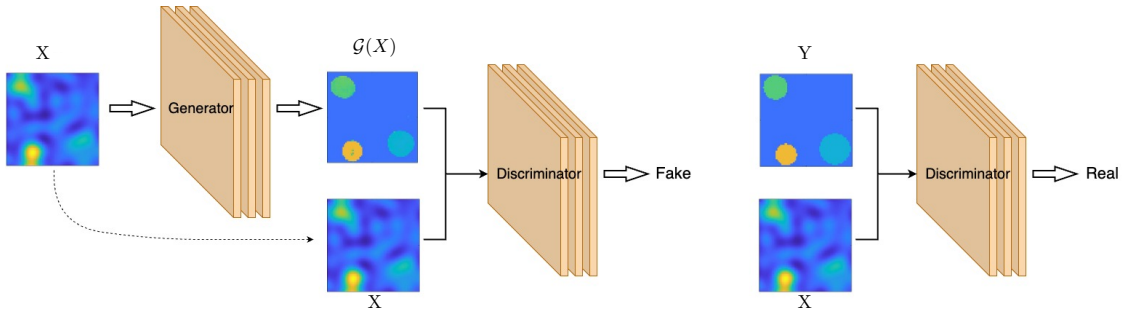


Figure 3. *The pix2pix algorithm. The generator $\mathcal{G}$ learns to fool the discriminator $\mathcal{D}$, while the discriminator $\mathcal{D}$ learns to distinguish the fake $(X, \mathcal{G}(X))$ and real (X, Y) tuples in the training set.*

a competitive manner wherein the generator network endeavors to generate images $\mathcal{G}(X_i)$ that resemble the target images Y_i, while the discriminator network attempts to differentiate between real image pairs (X_i, Y_i) and generated image pairs $(X_i, \mathcal{G}(X_i))$, with the output indicating the probability that an input image pair is real (i.e., it originates from the training dataset) or generated (i.e., it was produced by the generator network). The training process is illustrated in Figure 3.

The pix2pix algorithm utilizes a loss function consisting of two distinct components. The first component, denoted as the adversarial loss, is expressed in terms of a binary cross-entropy loss and is represented as follows:

$$\mathcal{L}_{\mathrm{cGAN}}(\mathcal{G}, \mathcal{D}) = \mathbb{E}(\log \mathcal{D}(X, Y)) + \mathbb{E}(\log(1 - \mathcal{D}(X, \mathcal{G}(X)))).$$

The second component of the loss function is referred to as the reconstruction loss and is defined as follows:

$$\mathcal{L}_{L^1}(\mathcal{G}) = \mathbb{E}(\|Y - \mathcal{G}(X)\|_{L^1}).$$

The total loss is then given by

$$\mathcal{L}(\mathcal{G}, \mathcal{D}) = \mathcal{L}_{\mathrm{cGAN}}(\mathcal{G}, \mathcal{D}) + \lambda \mathcal{L}_{L^1}(\mathcal{G}),$$

where λ represents a weight term that determines the relative importance of each component in the overall loss function. In practice, the expectations involved in the loss functions are approximated by sample means over batches of the training data. In the training process, we solve the optimization problem

$$\arg\min_{\theta_{\mathcal{G}}} \max_{\theta_{\mathcal{D}}} \mathcal{L}(\mathcal{G}, \mathcal{D}),$$

where $\theta_{\mathcal{G}}$ and $\theta_{\mathcal{D}}$ denote the parameters of the neural network models $\mathcal{G}$ and $\mathcal{D}$, respectively.

Following the approach proposed in [25], we employ the U-Net and patch-GAN architectures for the generator and discriminator networks, respectively. This choice enables improved performance and efficiency in a variety of image-to-image translation tasks.

5. Numerical experiments.

In this section, we present a comparative study of the data-assisted methods for the inverse random source problem.

5.1. Preparation for training.

For the reconstruction of the mean function g, the training dataset is generated as follows. The support of g is the union of three disks D_l with radius r_l and center (a_l, b_l), $l = 1, 2, 3$. For any fixed l, the radius r_l is generated from the uniform distribution on $[0.2, 0.4]$. The coordinates a_l and b_l are generated from the uniform distribution in $[-1 + r_l, 1 - r_l]$ so that the disk is randomly located but always contained in the rectangle $D = [-1, 1] \times [-1, 1]$. Note that the disks are allowed to overlap. All the random numbers are generated independently from each other, with different but fixed seeds so that the results can be reproduced. The source term g is set to 0 in D and l in D_l for $l = 1, 2, 3$ in a successive manner. The source term h is fixed as

$$h(x) = 0.6e^{-8[(x_1^2+x_2^2)^{1.5}-0.75(x_1^2+x_2^2)]}. \tag{5.1}$$

The pseudocolor plot of the function $h(x_1, x_2)$ is shown in Figure 4(a).

To generate the training dataset for the reconstruction of the variance function h, we switch the roles of g and h. More specifically, we consider the mean function g to be fixed and given by (5.1). In this setting, we set h to be equal to 0 within the domain D and assign the values of $1 + \frac{l}{2}$ inside each of the three respective disks, D_l, in a stepwise manner for $l = 1, 2, 3$. The disks D_l are generated in accordance with the same method as that used for constructing the mean function g.

For experiments with an inhomogeneous medium, the function $\eta(x)$ is fixed as

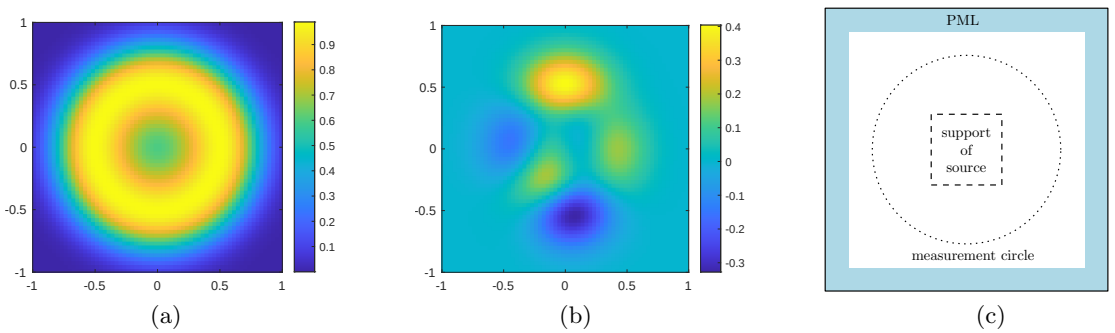
$$\eta(x) = q(3x_1, 3x_2), \tag{5.2}$$

where

$$q(x) = 0.5\left[0.3(1 - x_1)^2 e^{-x_1^2-(x_2+1)^2} - (0.2x_1 - x_1^3 - x_2^5)e^{-x_1^2-x_2^2} - 0.03e^{-(x_1+1)^2-x_2^2}\right].$$

The pseudocolor plot of $\eta(x)$ is shown in Figure 4(b).

The infinite domain is truncated to a finite one using the technique of perfectly matched layer (PML). The computational domain is $[-3, 3] \times [-3, 3]$ with a rectangular PML layer



(a) (b) (c)

Figure 4. (a) *Pseudocolor plot of the function h,* (b) *pseudocolor plot of the function η, and* (c) *computational domain for the direct problems.*

of thickness 0.5. The direct problem is solved by the finite element method (FEM), and the synthetic data are measured at 32 uniformly distributed points on the circle centered at the origin with radius 2. For any fixed source terms, the direct problem is solved for 5 wavenumbers $\kappa_j = (0.5 + j)\pi, j = 0, 1, 2, 3, 4$. A schematic of the computational domain for the direct problems is shown in Figure 4(c).

To generate random numbers, the GNU Scientific Library [17] is used. Mesh generation is carried out using Gmsh [19], and the FEM is implemented by FreeFEM [21]. To accelerate the computation of numerous direct problems, PETSc [37] is employed to solve linear systems with multiple right-hand sides for different realizations. Moreover, Open MPI is utilized for parallel computing with varying source terms and wavenumbers.

5.2. Stage 1 configuration. The direct problem is solved for fixed source terms g and h, with 5 distinct wavenumbers $\kappa_j = (0.5+j)\pi, j = 0, 1, 2, 3, 4$. The program is run for each fixed κ_j using either 100 realizations for the reconstruction of the mean or 1000 realizations for the reconstruction of the variance. The resulting data on the measurement circle are processed using the Kaczmarz algorithm for a homogeneous medium to produce initial reconstructed images [4]. The reconstructed images are sampled at a uniform grid size of 64×64 within the range of $[-1, 1] \times [-1, 1]$. For regularization, the parameter γ is set to 10^{-8}. The number of outer loops is taken to be 1.

5.3. Stage 2 configuration. For the PCA-based regression approach, the reduced dimension is set to $s_{\text{PCA}} = 1000$, while for the DMD approach, the reduced dimension is set to $s_{\text{DMD}} = 100$. We implement the U-Net architecture comprised of convolutional layers of kernel size 4×4 and batch normalization layers, following [24]. The weight λ in the pix2pix algorithm is set to 10. All learning-based approaches are trained and tested using the PyTorch framework. The training process is accelerated using an Nvidia Tesla P100 GPU.

5.4. Numerical experiments. We present numerical experiments in this section to compare the performance of different data-assisted methods. Unless otherwise specified, the testing dataset consists of 200 randomly generated samples.

Example 5.1. In the first example, we assess the performance of the U-Net method in Stage 2 with training datasets of varying sizes.

The reconstruction quality of the samples in the testing set is measured by the averaged L^1 relative errors of all samples. The accuracy and training time of the U-Net method are reported in Table 1. It is observed that as the size of the training dataset increases, the training time increases and the relative error decreases. However, the accuracy does not improve significantly when the number of examples exceeds 1600. Therefore, we present results with a training dataset consisting of $M = 1600$ samples in the subsequent examples to compare the performance of different approaches.

Figure 5 displays the loss versus epochs for the U-Net–based method, indicating that the loss decays slowly after 100 epochs. In the following examples, we present the neural networks trained with 100 epochs.

Example 5.2. In this example, we reconstruct the mean function g and variance function h^2 of the random source when the background medium is homogeneous.

Table 1

The training time and L^1 relative error for the U-Net–based method with different numbers of training samples.

Size of training set	400	800	1600	3200
Time	34	51	92	130
Relative error	0.24	0.21	0.20	0.19

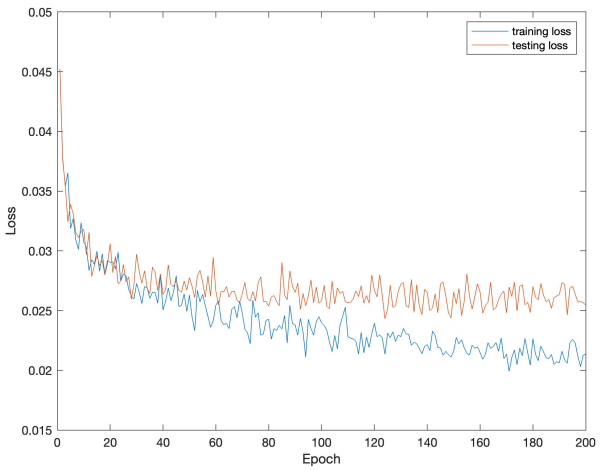


Figure 5. *Loss versus epoches for Example 5.1.*

We show representative pseudocolor plots of reconstructed images in Figures 6 and 7. Columns 1–2 display the ground-truth images and reconstructions from Stage 1 (the Kaczmarz method), while columns 3–6 show reconstructions after Stage 2 using PCA, DMD, U-Net, and pix2pix approaches.

The L^1 relative error in the testing set is presented in Tables 2 and 3. We observe that the Kaczmarz reconstructions capture the rough locations of the source but not their boundary and amplitude. In Stage 2, the PCA and the DMD methods provide better reconstructions, but the background is still not clear, and the boundary is blurry. With the two neural network algorithms U-Net and pix2pix, the reconstructions are noticeably enhanced, and the homogeneous background is clearly separated from the inclusions. This improvement is also verified by the L^1 relative errors in Tables 2 and 3.

Example 5.3. In this example, we reconstruct the mean function g and variance function h^2 of the random source when the medium is inhomogeneous, and the function η is given by (5.2).

Figures 8 and 9 present pseudocolor plots of some representative reconstructions. The reconstructions from the first stage are worse than the initial approximations in Example 5.2 since the Kaczmarz method is derived utilizing the integral equations corresponding to the direct problem with a homogeneous medium. In the subsequent stage, the PCA and DMD approaches provide enhanced reconstructions; however, the background is still unclear. The U-Net and pix2pix algorithms lead to much improved reconstructions with a clean back-

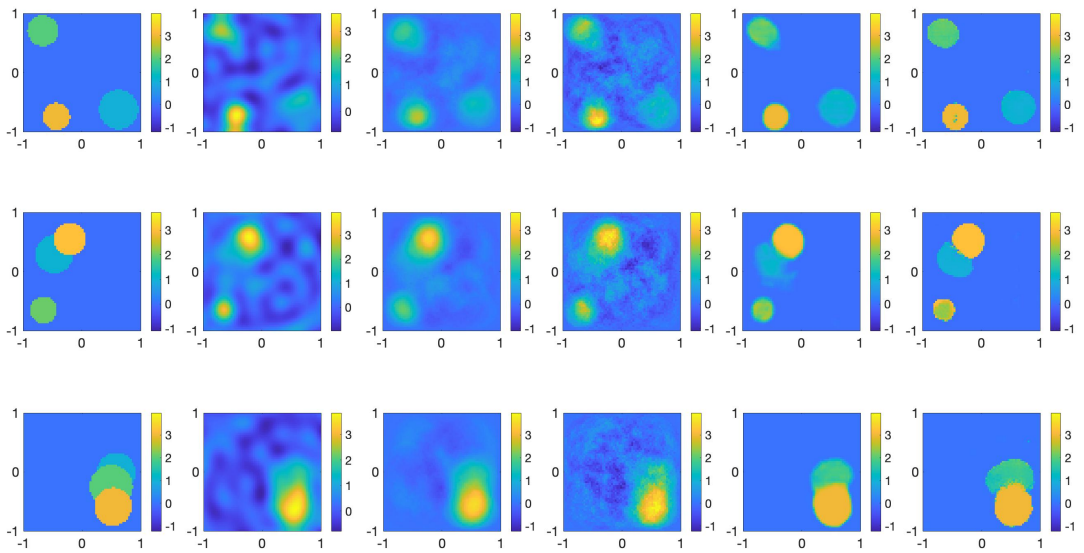


Figure 6. *Pseudocolor plots of the mean functions g and their reconstructions in a homogeneous medium. The columns from the left are the ground-truth images followed by the reconstructed images by the Kaczmarz method, the PCA-based method, the DMD-based method, the U-Net–based method, and the pix2pix method.*

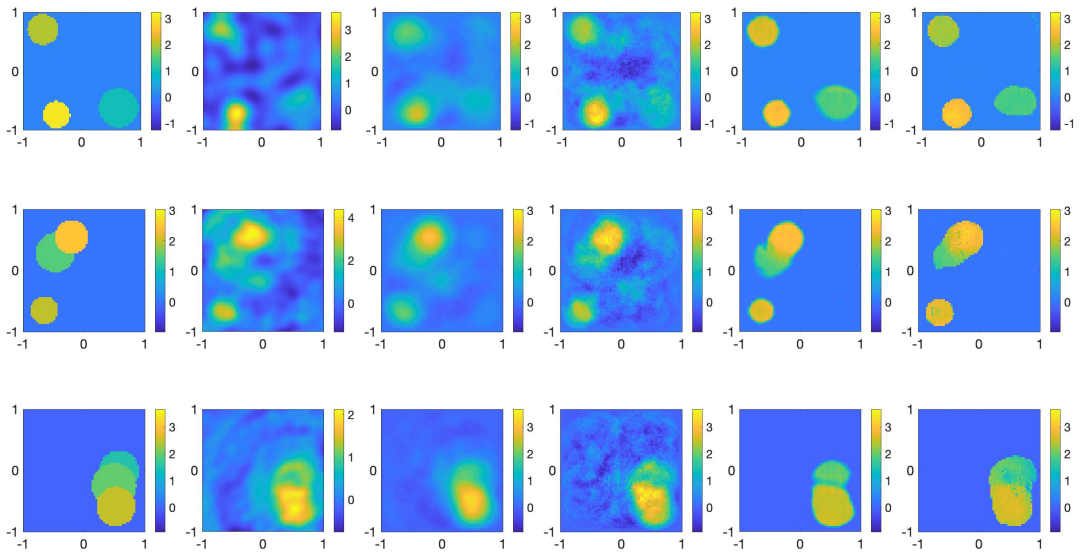


Figure 7. *Pseudocolor plots of the variance functions h and their reconstructions in a homogeneous medium. The columns from the left are the ground-truth images followed by the reconstructed images by the Kaczmarz method, the PCA-based method, the DMD-based method, the U-Net–based method, and the pix2pix method.*

ground, sharp boundary, and correct amplitude. The improvements are also confirmed by the L^1 relative errors shown in Tables 4 and 5, indicating that the hidden information of the inhomogeneity has been successfully captured by the learning methods.

Table 2

Training time and L^1 relative error for the reconstruction of the mean function g in a homogeneous medium.

Algorithm	PCA	DMD	U-Net	pix2pix
Time	4	6	74	693
Relative error	0.62	0.63	0.20	0.22

Table 3

Training time and L^1 relative error for the reconstruction of the variance function h in a homogeneous medium.

Algorithm	PCA	DMD	U-Net	pix2pix
Time	4	6	74	729
Relative error	0.90	0.77	0.28	0.30

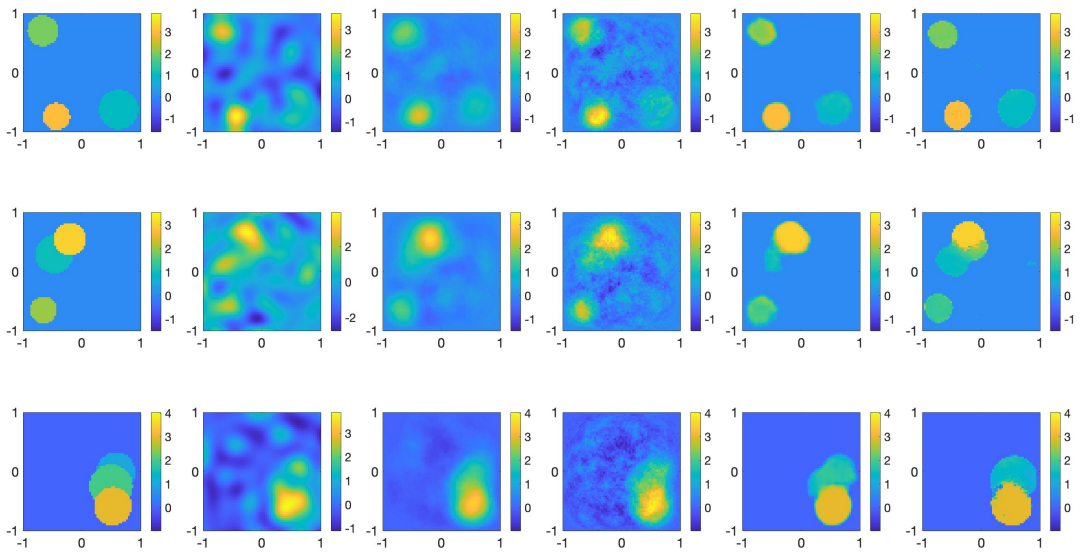


Figure 8. *Pseudocolor plots of the mean functions g and their reconstructions in an inhomogeneous medium. The columns from the left are the ground-truth images followed by the reconstructed images by the Kaczmarz method, the PCA-based method, the DMD-based method, the U-Net–based method, and the pix2pix method.*

Example 5.4. In this example, the methods are trained with noisy measurements to reconstruct the mean of the random source when medium is inhomogeneous.

It can be observed from Table 6 that the reconstructions obtained through all the employed techniques exhibit considerable stability with respect to measurement noise. The next example is to address the generalization ability of our methods.

Example 5.5. In this example, we test the performance of the methods trained with the Cylinder dataset on several samples that are distinct from the training set. The medium is homogeneous, and the mean reconstruction is presented. The first sample includes five inclusionsm and the second sample is comprised of one disk and two ellipses.

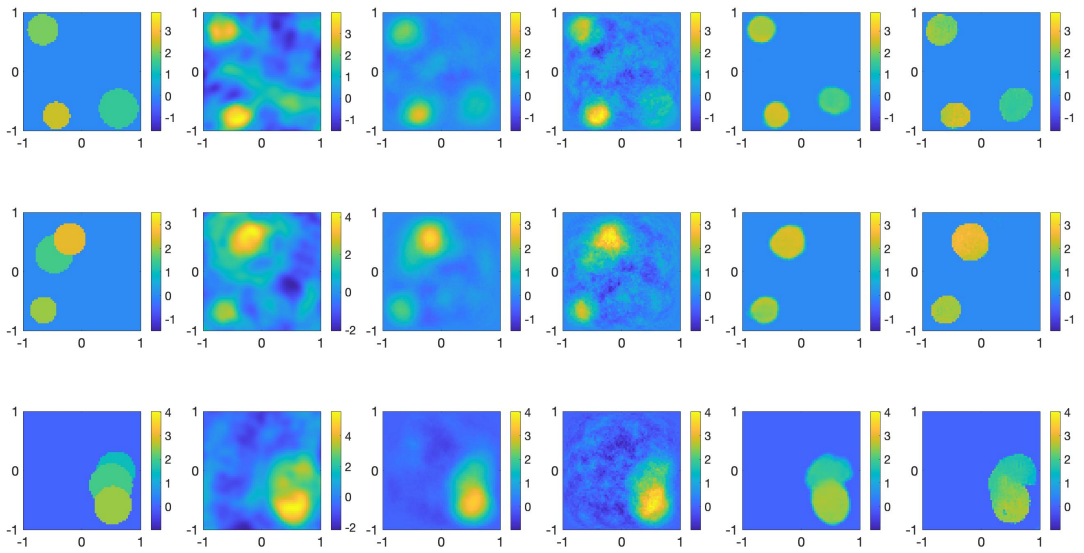


Figure 9. *Pseudocolor plots of the variance functions h and their reconstructions in an inhomogeneous medium. The columns from the left are the ground-truth images followed by the reconstructed images by the Kaczmarz method, the PCA-based method, the DMD-based method, the U-Net–based method, and the pix2pix method.*

Table 4

Training time and L^1 relative error for the reconstruction of the mean function g in an inhomogeneous medium.

Algorithm	PCA	DMD	U-Net	pix2pix
Time	4	6	79	697
Relative error	0.74	0.64	0.27	0.28

Table 5

Training time and L^1 relative error for the reconstruction of the variance function h in an inhomogeneous medium.

Algorithm	PCA	DMD	U-Net	pix2pix
Time	4	6	104	694
Relative error	1.09	0.75	0.30	0.36

Table 6

L^1 relative error of the mean reconstruction with different methods trained with measurement data of different noise levels in an unknown inhomogeneous medium.

Noise level	PCA	DMD	U-Net	pix2pix
0.05	0.64	0.77	0.26	0.27
0.1	0.64	0.73	0.27	0.27
0.2	0.65	0.73	0.27	0.27

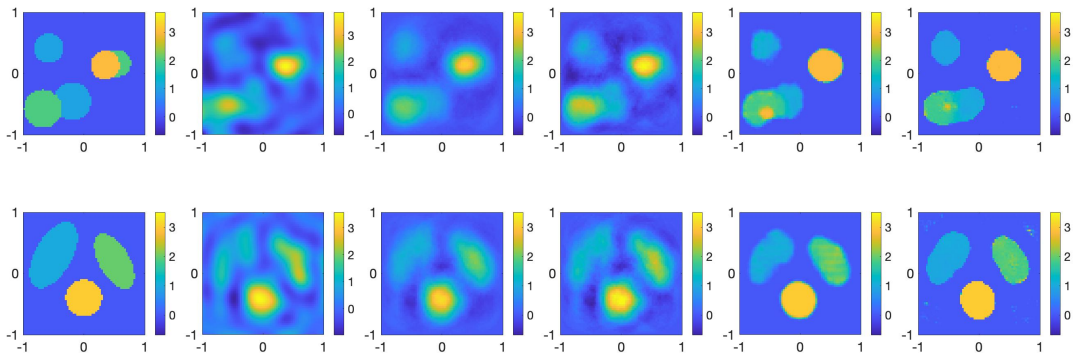


Figure 10. *Reconstructed mean of the random source of different shapes with the neural network trained with the Cylinder dataset. The columns from the left are the ground-truth images followed by reconstructed images by the Kaczmarz method, the PCA-based method, the DMD-based method, the U-Net–based method, and the pix2pix method.*

Although the test samples differ from the samples in the training set, with more inclusions or inclusions of different shapes, the U-Net and pix2pix methods can still provide satisfactory reconstructions of well-separated inclusions with accurate positions, as shown in Figure 10. All data-based methods improve the reconstruction to varying extents. Compared to the other methods, the reconstructions of the pix2pix algorithm are less blurry, with the edges clearly reconstructed.

6. Conclusion.

In this work, we have presented a novel approach to solving the inverse random source problem, a fundamental challenge in many scientific and engineering applications. Our two-stage method leverages data-assisted techniques to provide accurate and efficient solutions. In the first stage, we utilize the regularized Kaczmarz algorithm to obtain an approximation of the statistical properties of the random source based on integral equations. This approximation serves as the input to the second stage, where data-assisted methods are used to learn the mapping from the approximations to the exact mean and variance of the source.

To evaluate the effectiveness of our approach, we conduct a comparative study of different data-assisted approaches. The results demonstrate that neural network–assisted techniques outperform other methods, offering stable and accurate reconstructions with a relatively small number of realizations for both homogeneous and inhomogeneous media.

REFERENCES

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

[2] S. Arridge, P. Maass, O. Öktem, and C. Schönlieb, *Solving inverse problems using data-driven models*, Acta Numer., 28 (2019), pp. 1–174.

[3] M. Badieirostami, A. Adibi, H. Zhou, and S.-N. Chow, *Wiener chaos expansion and simulation of electromagnetic wave propagation excited by a spatially incoherent source*, Multiscale Model. Simul., 8 (2010), pp. 59–604.

[4] G. Bao, C. Chen, and P. Li, *Inverse random source scattering problems in several dimensions*, SIAM-ASA J. Uncertain. Quantif., 4 (2016), pp. 1263–1287.
[5] G. Bao, C. Chen, and P. Li, *Inverse random source scattering for elastic waves*, SIAM J. Numer. Anal., 55 (2017), pp. 2616–2643.
[6] G. Bao, S.-N. Chow, P. Li, and H. Zhou, *An inverse random source problem for the Helmholtz equation*, Math. Comp., 83 (2014), pp. 215–233.
[7] G. Bao, P. Li, J. Lin, and F. Triki, *Inverse scattering problems with multi-frequencies*, Inverse Problems, 31 (2015), 093001.
[8] K. Bhattacharya, B. Hosseini, N. B. Kovachki, and A. M. Stuart, *Model Reduction and Neural Networks for Parametric PDEs*, preprint, arXiv:2005.03180, 2020.
[9] N. Bleistein and J. K. Cohen, *Nonuniqueness in the inverse source problem in acoustics and electromagnetics*, J. Math. Phys., 18 (1977), pp. 194–201.
[10] J. Bongard and H. Lipson, *Automated reverse engineering of nonlinear dynamical systems*, Proc. Natl. Acad. Sci. USA, 104 (2007), pp. 9943–9948.
[11] Y. Chen, L. Lu, G. E. Karniadakis, and L. D. Negro, *Physics-informed neural networks for inverse problems in nano-optics and metamaterials*, Opt. Express, 28 (2020), pp. 1161–11633.
[12] X. Chen, Z. Wei, M. Li, and P. Rocca, *A review of deep learning approaches for inverse scattering problems (invited review)*, Prog. Electromagn. Res., 167 (2020), pp. 67–81.
[13] M. V. de Hoop, M. Lassas, and C. A. Christopher, *Deep learning architectures for nonlinear operator functions and nonlinear inverse problems*, Math. Stat. Learn., 4 (2022), pp. 1–86.
[14] A. Devaney and G. Sherman, *Nonuniqueness in inverse source and scattering problems*, IEEE Trans. Antennas and Propagation, 30 (1982), pp. 1034–1037.
[15] T. Elfving, P. C. Hansen, and T. Nikazad, *Semi-convergence properties of Kaczmarz's method*, Inverse Problems, 30 (2014), 055007.
[16] J. P. Fouque, J. Garnier, G. Papanicolaou, and K. Solna, *Wave Propagation and Time Reversal in Randomly Layered Media*, Springer, New York, 2007.
[17] M. Galassi, J. Davies, J. Theiler, B. Gough, G. Jungman, P. Alken, M. Booth, F. Rossi, and R. Ulerich, *GNU Scientific Library*, Network Theory Ltd, Godalming, UK, 2002.
[18] Y. Gao, H. Liu, X. Wang, and K. Zhang, *On an artificial neural network for inverse scattering problems*, J. Comput. Phys., 448 (2022), 110771.
[19] C. Geuzaine and J. F. Remacle, *Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities*, Internat. J. Numer. Methods Engrg., 79 (2009), pp. 1309–1331.
[20] R. Guo and J. Jiang, *Construct deep neural networks based on direct sampling methods for solving electrical impedance tomography*, SIAM J. Sci. Comput., 43 (2021), pp. B678–B711.
[21] F. Hecht, *New development in FreeFem++*, J. Numer. Math., 20 (2012), pp. 251–266.
[22] T. Helin, M. Lassas, and L. Päivärinta, *Inverse acoustic scattering problem in half-space with anisotropic random impedance*, J. Differential Equations, 262 (2017), pp. 3139–3168.
[23] H. Hotelling, *Analysis of a complex of statistical variables into principal components*, J. Educ. Psychol., 24 (1933), pp. 417–441.
[24] S. Ioffe and C. Szegedy, *Batch normalization: Accelerating deep network training by reducing internal covariate shift*, in Proceedings of the International Conference on Machine Learning, 2015, pp. 448–456.
[25] P. Isola, J. Zhu, T. Zhou, and A. A. Efros, *Image-to-Image translation with conditional adversarial networks*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1125–1134.
[26] Y. Khoo and L. Ying, *SwitchNet: A neural network model for forward and inverse scattering problems*, SIAM J. Sci. Comput., 41 (2019), pp. A3182–A3201.
[27] M. Li, C. Chen, and P. Li, *Inverse random source scattering for the Helmholtz equation in inhomogeneous media*, Inverse Problems, 34 (2017), 015003.
[28] J. Li, P. Li, and X. Wang, *Inverse source problems for the stochastic wave equations: Far-field patterns*, SIAM J. Appl. Math., 82 (2022), pp. 1113–1134.
[29] P. Li and X. Wang, *An inverse random source problem for the one-dimensional Helmholtz equation with attenuation*, Inverse Problems, 37 (2020), 015009.

[30] P. Li and G. Yuan, *Stability on the inverse random source scattering problem for the one-dimensional Helmholtz equation*, J. Math. Anal. Appl., 450 (2017), pp. 872–887.
[31] E. A. Marengo and A. J. Devaney, *The inverse source problem of electromagnetics: Linear inversion formulation and minimum energy solution*, IEEE Trans. Antennas and Propagation, 47 (1999), pp. 410–412.
[32] F. Natterer, *The Mathematics of Computerized Tomography*, Teubner, Stuttgart, 1986.
[33] J. Ning, F. Han, and J. Zou, *A Direct Sampling-Based Deep Learning Approach for Inverse Medium Scattering Problems*, preprint, arXiv:2305.00250, 2023.
[34] O. Ronneberger, P. Fischer, and T. Brox, *U-net: Convolutional networks for biomedical image segmentation*, in Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, 2015, pp. 234–241.
[35] P. J. Schmid, *Dynamic mode decomposition of numerical and experimental data*, J. Fluid Mech., 656 (2010), pp. 5–28.
[36] M. Schmidt and H. Lipson, *Distilling free-form natural laws from experimental data*, Science, 324 (2009), pp. 81–85.
[37] B. F. Smith, *PETSc (Portable, Extensible Toolkit for Scientific Computation)*, in Encyclopedia of Parallel Computing, D. Padua, ed., Springer, New York, 2011, pp. 1530–1539.
[38] G. Uhlmann and Y. Wang, *Convolutional neural networks in phase space and inverse problems*, SIAM J. Appl. Math., 80 (2020), pp. 2560–2585.
[39] K. Xu, L. Wu, X. Ye, and X. Chen, *Deep learning-based inversion methods for solving inverse scattering problems with phaseless data*, IEEE Trans. Antennas and Propagation, 68 (2020), pp. 7457–7570.
[40] B. Yu and W. E, *The deep Ritz method: A deep learning-based numerical algorithm for solving variational problems*, Commun. Math. Stat., 6 (2018), pp. 1–12.
[41] Y. Zang, G. Bao, X. Ye, and H. Zhou, *Weak adversarial networks for high-dimensional partial differential equations*, J. Comput. Phys., 411 (2020), 109409.
[42] Y. Zhou, Y. Zhong, Z. Wei, T. Yin, and X. Chen, *An improved deep learning scheme for solving 2-D and 3-D inverse scattering problems*, IEEE Trans. Antennas and Propagation, 69 (2020), pp. 2853–2863.
[43] Y. Zhu and N. Zabaras, *Bayesian deep convolutional encoder-decoder networks for surrogate modeling and uncertainty quantification*, J. Comput. Phys., 366 (2018), pp. 415–447.