



**Michigan
Technological
University**

Michigan Technological University
Digital Commons @ Michigan Tech

Michigan Tech Publications, Part 2

2-7-2024

Deep Neural Network-Oriented Indicator Method for Inverse Scattering Problems Using Partial Data

Yule Lin
Jinan University

Xiaoyi Yan
Jinan University

Jiguang Sun
Michigan Technological University, jiguangs@mtu.edu

Juan Liu
Jinan University

Follow this and additional works at: <https://digitalcommons.mtu.edu/michigantech-p2>



Part of the [Mathematics Commons](#)

Recommended Citation

Lin, Y., Yan, X., Sun, J., & Liu, J. (2024). Deep Neural Network-Oriented Indicator Method for Inverse Scattering Problems Using Partial Data. *Mathematics*, 12(4). <http://doi.org/10.3390/math12040522>
Retrieved from: <https://digitalcommons.mtu.edu/michigantech-p2/597>

Follow this and additional works at: <https://digitalcommons.mtu.edu/michigantech-p2>



Part of the [Mathematics Commons](#)

Article

Deep Neural Network-Oriented Indicator Method for Inverse Scattering Problems Using Partial Data

Yule Lin ^{1,†}, Xiaoyi Yan ^{1,†}, Jiguang Sun ^{2,†} and Juan Liu ^{1,*,†}¹ Department of Mathematics, Jinan University, Guangzhou 510632, China; linyule@stu2021.jnu.edu.cn (Y.L.); yanxiaoyi@stu2022.jnu.edu.cn (X.Y.)² Department of Mathematical Sciences, Michigan Technological University, Houghton, MI 49931, USA; jiguangs@mtu.edu

* Correspondence: liujuan@jnu.edu.cn

† These authors contributed equally to this work.

Abstract: We consider the inverse scattering problem to reconstruct an obstacle using partial far-field data due to one incident wave. A simple indicator function, which is negative inside the obstacle and positive outside of it, is constructed and then learned using a deep neural network (DNN). The method is easy to implement and effective as demonstrated by numerical examples. Rather than developing sophisticated network structures for the classical inverse operators, we reformulate the inverse problem as a suitable operator such that standard DNNs can learn it well. The idea of the DNN-oriented indicator method can be generalized to treat other partial data inverse problems.

Keywords: inverse obstacle scattering; deep neural network; partial data; indicator function

MSC: 65N21; 78A46



Citation: Lin, Y.; Yan, X.; Sun J.; Liu, J. Deep Neural Network-Oriented Indicator Method for Inverse Scattering Problems Using Partial Data. *Mathematics* **2024**, *12*, 522. <https://doi.org/10.3390/math12040522>

Academic Editor: Mohsen Tadi

Received: 19 December 2023

Revised: 25 January 2024

Accepted: 2 February 2024

Published: 7 February 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Inverse scattering problems arise in many areas such as medical imaging and seismic detection. The scattering from obstacles leads to exterior boundary value problems for the Helmholtz equation. The two-dimensional case is derived from the scattering from finitely long cylinders, and more importantly, it can serve as a model case for testing numerical approximation schemes in direct and inverse scattering [1]. In this paper, we consider the inverse problem to reconstruct an obstacle, given partial far-field data in \mathbb{R}^2 .

Let $D \subset \mathbb{R}^2$ be a bounded domain with a C^2 -boundary ∂D . Denoting by $u^i(x)$ the incident plane wave, the direct scattering problem for a sound-soft obstacle is to find the scattered field u^s such that

$$\begin{cases} \Delta u^s + k^2 u^s = 0, & \text{in } \mathbb{R}^2 \setminus \overline{D}, \\ u^s = -u^i, & \text{on } \partial D, \\ \lim_{r \rightarrow \infty} \sqrt{r} \left(\frac{\partial u^s}{\partial r} - iku^s \right) = 0, & r = |x|, \end{cases} \quad (1)$$

where $k > 0$ is the wavenumber. It is well known that u^s has an asymptotic expansion

$$u^s(x) = \frac{e^{ikr}}{\sqrt{r}} \left\{ u_\infty(\hat{x}) + \mathcal{O}\left(\frac{1}{r}\right) \right\} \quad \text{as } r := |x| \rightarrow \infty$$

uniformly in all directions $\hat{x} = x/|x|$. The function $u_\infty(\hat{x})$ defined on $\mathbb{S} := \{\hat{x} \in \mathbb{R}^2; |\hat{x}| = 1\}$ is the far-field pattern of u^s for D due to the incident field u^i . The inverse problem of interest is to reconstruct D from n far-field patterns $u_\infty(\hat{x}_1), u_\infty(\hat{x}_2), \dots, u_\infty(\hat{x}_n)$ for one incident wave. Note that the unique determination of D using the far-field pattern due to one incident wave is an open problem.

Classical inverse scattering methods to reconstruct an obstacle mainly fall into two groups. The first group is the optimization methods to minimize certain cost functions [2,3]. These methods usually solve a number of direct scattering problems and work well when proper initial guesses are available. The second group is the sampling methods, e.g., the linear sampling method, the extended sampling method, the orthogonality method, the reverse time migration, and the direct sampling method [4–8]. These methods often construct some indicator functions based on the analysis of the partial differential equations to determine if a sampling point is inside or outside the obstacle. In contrast to the optimization methods, the sampling methods do not need initial guesses and forward solvers but usually provide less information of the unknown obstacle.

Although successful for many cases, classical methods have difficulties for problems with partial data. For example, the linear sampling method cannot obtain a reasonable reconstruction if the data are collected on a small aperture. Recently, there has been an increasing interest in solving inverse problems using deep neural networks (DNNs). Various DNNs have been proposed to reconstruct the obstacle from the scattered field [9–12]. Although the deep learning methods can process partial data, the performance is highly problem-dependent and the network structures are often sophisticated. Combinations of deep learning and sampling methods for the inverse scattering problems have also been investigated recently [13–16]. These methods usually construct a DNN to learn the reconstruction of some sampling method.

In this paper, we propose a DNN-oriented indicator method. The idea is to design an indicator function which is simple enough for standard DNNs to learn it effectively. We borrow from the sampling methods the concept of the indicator. What sets our method apart from the sampling methods is that the indicator is defined as a signed distance function other than using the inner product related to Green's function or solving the linear ill-posed integral equations. The indicator is then learned by a DNN with partial scattering data as input. This DNN-oriented indicator method inherits the advantage of deep learning methods by compensating for partial information through empirical data. Meanwhile, it retains the simplicity and flexibility of sampling methods. Numerical experiments demonstrate its effectiveness for obstacle reconstructions with a few far-field data due to one incident wave. Note that the idea is to find a formulation for the inverse problem that is easy to learn instead of the design of sophisticated neural networks (see [17]).

The rest of this paper is organized as follows. In Section 2, the DNN-oriented indicator method is proposed, and the data structure is specified. In Section 3, numerical experiments are carried out to demonstrate the performance of the proposed method. Conclusions are given in Section 4.

2. Dnn-Oriented Indicator Method

We propose a DNN-oriented indicator method for the inverse obstacle scattering problem in \mathbb{R}^2 . The framework can be directly applied to other inverse problems and extended to \mathbb{R}^3 . For an obstacle D , we define a signed distance function

$$d_{\partial D}(z) = \begin{cases} -\inf_{x \in \partial D} |x - z|, & z \in D, \\ 0, & z \in \partial D, \\ \inf_{x \in \partial D} |x - z|, & z \in \mathbb{R}^2 \setminus \bar{D}. \end{cases} \quad (2)$$

It is clear that $d_{\partial D}$ is a continuous function with respect to z and can be used to reconstruct the obstacle since

$$\partial D = \{z \in \mathbb{R}^2; d_{\partial D}(z) = 0\}, \quad D = \{z \in \mathbb{R}^2; d_{\partial D}(z) < 0\}.$$

Let $\mathbf{u}_\infty = [u_\infty(\hat{x}_1), u_\infty(\hat{x}_2), \dots, u_\infty(\hat{x}_n)]$ be the vector of far-field data due to one incident wave in n observation directions. Let Ω be a (large) domain, known as a prior, containing D . For a point $z \in \Omega$, we define the indicator function as

$$I(\mathbf{u}_\infty, \mathbf{z}) := d_{\partial D}(\mathbf{z}). \quad (3)$$

Our goal is to design a DNN to learn $I(\mathbf{u}_\infty, \mathbf{z})$ for input $(\mathbf{u}_\infty, \mathbf{z})$. More specifically, we expect to build a neural network $\tilde{I} : \mathbb{R}^{2n+2} \rightarrow \mathbb{R}$ to approximate the indicator function I . The input of \tilde{I} is a vector in \mathbb{R}^{2n+2} consisting of the real and imaginary parts of \mathbf{u}_∞ and the coordinates of a point $\mathbf{z} \in \Omega$. The output is the corresponding signed distance. Assume that there are L hidden layers for \tilde{I} . Each hidden layer is defined as

$$h_l(\mathbf{y}_{out}^l) = \sigma(\mathbf{b}_l + \mathbf{W}_l \mathbf{y}_{in}^l), \quad l = 1, \dots, L, \quad (4)$$

where \mathbf{y}_{in}^l and \mathbf{y}_{out}^l are the input and output vectors of the l th layer, respectively, \mathbf{W}_l is the output weight matrix, and \mathbf{b}_l is the bias vector. The non-linear function $\sigma(\cdot)$ is the rectified linear unit (ReLU) defined as $\sigma(y) = \max(0, y)$. The input layer with output weight matrix \mathbf{W}_0 and bias vector \mathbf{b}_0 is connected to the first hidden layer.

A set of N data is used to train the fully connected network \tilde{I} . The set of parameters is denoted by $\Theta = \{\mathbf{b}_0, \mathbf{b}_1, \dots, \mathbf{b}_L, \mathbf{M}_0, \mathbf{M}_1, \dots, \mathbf{M}_L\}$ and is updated by the backpropagation algorithm, minimizing the half quadratic error of the predicted responses between the set of training mapped measurements $\tilde{d}_j, j = 1, \dots, N$, and exact values $d_j, j = 1, \dots, N$. The loss function is the half mean-squared error given by

$$L(\Theta) = \frac{1}{2} (d_j - \tilde{d}_j)^2. \quad (5)$$

The algorithm for the DNN-oriented indicator method is illustrated in Algorithm 1. It contains an offline phase and an online phase. In the offline phase, the training data are used to learn the network parameters. Then, the network \tilde{I} is used in the online phase to output the indicator for each $\mathbf{z} \in \Omega$. Using the same notation \tilde{I} for the function produced by the trained network \tilde{I} , $\tilde{I}(\mathbf{u}_\infty, \mathbf{z}) < 0$ indicates that $\mathbf{z} \in D$, whereas $\tilde{I}(\mathbf{u}_\infty, \mathbf{z}) > 0$ indicates that $\mathbf{z} \in \mathbb{R}^2 \setminus \bar{D}$. The approximate support of D can be reconstructed accordingly.

Algorithm 1 DNN-oriented indicator method.

offline phase

- 1: Generate far-field vectors $\mathbf{u}_\infty^{(j_1)}, j_1 = 1, \dots, J_1$ for random objects $D^{(j_1)}$ in Ω
- 2: Generate points $\mathbf{z}^{(j_2)} := (z_1^{(j_2)}, z_2^{(j_2)})$, $j_2 = 1, \dots, J_2$ for Ω
- 3: Collect training data $\mathbf{y}_{in}^{(j_1, j_2)} := [\Re \mathbf{u}_\infty^{(j_1)}, \Im \mathbf{u}_\infty^{(j_1)}, z_1^{(j_2)}, z_2^{(j_2)}]$ and $\mathbf{y}_{out}^{(j_1, j_2)} := d_{\partial D^{(j_1)}}(\mathbf{z}^{(j_2)})$
- 4: Construct the DNN \tilde{I}
- 5: Train \tilde{I} using $\{\mathbf{y}_{in}^{(j_1, j_2)}, \mathbf{y}_{out}^{(j_1, j_2)}\}, j_1 = 1, \dots, J_1, j_2 = 1, \dots, J_2$

online phase

- 6: Measure far-field data \mathbf{u}_∞ for unknown object D
 - 7: Generate a set T of uniformly distributed points for any subset of Ω which contains D
 - 8: For each $\mathbf{z} = (z_1, z_2) \in T$, use $[\Re \mathbf{u}_\infty, \Im \mathbf{u}_\infty, z_1, z_2]$ as the input for \tilde{I} to predict the indicator $\tilde{I}(\mathbf{u}_\infty, \mathbf{z})$
 - 9: Approximate D using $\{\mathbf{z} | \tilde{I}(\mathbf{u}_\infty, \mathbf{z}) \leq 0\}$
-

Remark 1. Other indicator functions, e.g., the characteristic function for D , can be used and have similar performance. The advantage of the indicator function (3) is that it provides information on the distance of a point to the boundary of the obstacle.

Remark 2. The set Ω in the offline phase and online phase can be different, which allows additional flexibility.

3. Numerical Experiments

We present some numerical examples to demonstrate the proposed DNN-oriented indicator method. All the experiments are conducted on a laptop with an Intel(R) Core(TM) i7-10510U CPU @ 1.80 GHz (up to 2.30 GHz) and 16 GB of RAM.

The wave number is $k = 1$, and the incident plane wave is $u^i(x) = e^{ikx \cdot d}$ with $d = (1, 0)$. The training data are generated using a boundary integral method for (1) [1]. The observation data $\mathbf{u}_\infty = [u_\infty(\hat{x}_1), u_\infty(\hat{x}_2), \dots, u_\infty(\hat{x}_6)]$ contain the far-field patterns of 6 observation directions

$$\hat{x}_j = (\cos \theta_j, \sin \theta_j), \quad \theta_j \in [0, \pi/2], \quad j = 1, \dots, 6$$

uniformly distributed on one-quarter of the unit circle.

For the offline phase, we use $J_1 = 1600$ random obstacles $D^{(j_1)}, j_1 = 1, \dots, J_1$, which are star-shaped domains [18]:

$$\rho(\theta)(\cos \theta, \sin \theta) + (c_1, c_2), \quad \theta \in [0, 2\pi), \quad (6)$$

where

$$\rho(\theta) = a_0 \left\{ 1 + \frac{1}{2M} \sum_{m=1}^M (a_m \cos(m\theta) + b_m \sin(m\theta)) \right\}. \quad (7)$$

In the experiments, $M = 5$ and the coefficients $a_0 \in [0.5, 1.5]$, $a_m, b_m \in [-1, 1]$, $c_1, c_2 \in [-2, 2]$ are uniformly distributed random numbers. We use $J_2 = 51 \times 51$ points uniformly distributed in the region $\Omega = [-5, 5] \times [-5, 5]$. The training data consist of $\mathbf{u}_\infty^{(j_1)}$ for each obstacle $D^{(j_1)}$ and each sampling point $\mathbf{z}^{(j_2)}$ as input and the signed distance function at $\mathbf{z}^{(j_2)}$ for $D^{(j_1)}$ as output. The size of the training set is $N = J_1 \times J_2$.

The neural network is fully connected feed-forward with $L + 2$ layers and the numbers of nodes in each layer are $14, N_h, \dots, N_h, 1$, respectively, where the number of hidden layers L and the number of neurons in each hidden layer N_h are chosen using the rule of thumb [19] and a trial-and-error method. The network is trained for up to 5.5×10^4 iterations using a mini-batch size of 300 and 4 epochs. We normalize the input per feature to zero mean and standard deviation one, and use the adaptive moment estimation (Adam) with an initial learning rate of 0.01, gradually reducing the learning rate by a factor of 0.1 every 3 epochs. When $L = 2$ and $N_h = 20$, the training duration is approximately 13 min and 57 s, while the testing duration is less than 1 s.

To evaluate the effectiveness of the DNN \tilde{I} , we generate a test set of size $N^* = \frac{1}{4}J_1 \times J_2$. The following relative error for the test set is used:

$$\epsilon = \frac{1}{N^*} \sum_{j_1=1}^{\frac{1}{4}J_1} \sum_{j_2=1}^{J_2} \frac{|I(\mathbf{u}_\infty^{(j_1)}, \mathbf{z}^{(j_2)}) - \tilde{I}(\mathbf{u}_\infty^{(j_1)}, \mathbf{z}^{(j_2)})|}{|I(\mathbf{u}_\infty^{(j_1)}, \mathbf{z}^{(j_2)})|}.$$

We test the network with different L and N_h and show the relative errors utilizing noiseless far-field data and far-field data with 5% random noises in Table 1. For each L , the upper row is for the noiseless data and the lower row is for noisy data. Since the errors are smaller for $L = 2$ and $N_h = 20$, we use them for \tilde{I} in the online phase.

Remark 3. The results indicate that the inverse operator only needs a simple network with reasonably small L and N_h . More sophisticated networks negatively affect the performance.

Remark 4. The complexity of the neural network depends on various elements, including the number and locations of far-field data, the wavenumber k , etc. One would expect different N_h and L for different settings.

Table 1. Relative errors for DNNs with different L and N_h .

	$N_h = 10$	$N_h = 15$	$N_h = 20$	$N_h = 25$	$N_h = 30$
$L = 2$	19.008%	16.066%	12.940%	13.791%	16.091%
	21.669%	18.409%	15.531%	16.580%	18.883%
$L = 3$	15.048%	13.823%	14.245%	13.808%	13.390%
	16.959%	16.047%	16.157%	15.932%	16.101%
$L = 4$	14.937%	14.791%	15.854%	15.464%	14.348%
	17.522%	16.680%	18.546%	18.369%	17.822%

To visualize the reconstructions, we use three obstacles: a triangle with boundary given by

$$(1 + 0.15 \cos 3t)(\cos \theta, \sin \theta) + (-2, 1), \quad \theta \in [0, 2\pi),$$

a peanut with boundary given by

$$1.5\sqrt{\cos^2 \theta + 0.25 \sin^2 \theta}(\cos \theta, \sin \theta) + (1, 2), \quad \theta \in [0, 2\pi),$$

a kite with boundary given by

$$(0.75 \sin t, 0.5 \cos t + 0.325 \cos 2t - 0.325) + (-2, -1), \quad t \in [0, 2\pi),$$

and a square whose vertices are

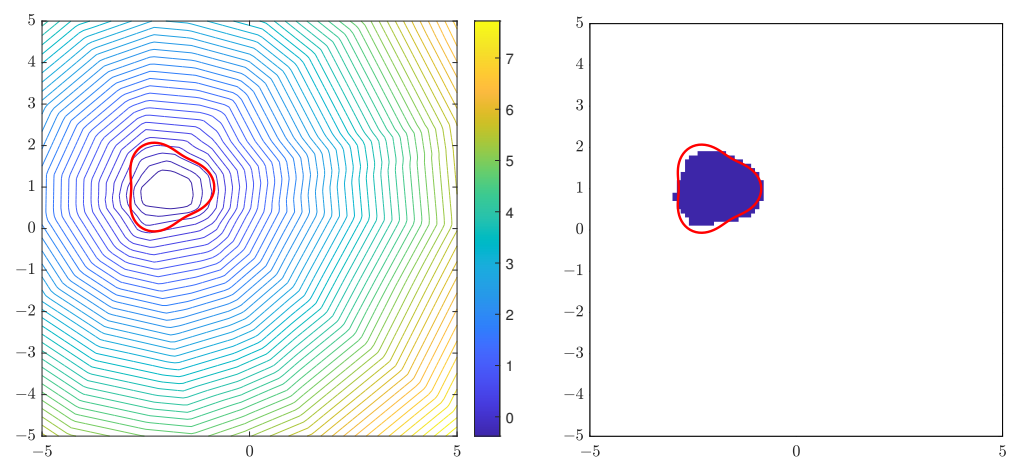
$$(1, 0.5), (2.5, -1), (1, -2.5), (-0.5, -1).$$

Let the uniformly distributed point set T for Ω be

$$T := \{(-5 + 0.1m, -5 + 0.1n), \quad m, n = 0, 1, \dots, 100\}. \quad (8)$$

For each $z \in T$, the trained DNN is used to predict $\tilde{I}(\mathbf{u}_\infty, z)$. We plot the contours for $\tilde{I}(\mathbf{u}_\infty, z)$ with respect to $z \in T$, and obtain the reconstructions by finding all $z \in T$ such that $\tilde{I}(\mathbf{u}_\infty, z) \leq 0$.

For the triangle obstacle, Figure 1 shows the contour plot of the indicator function $\tilde{I}(\mathbf{u}_\infty, \cdot)$ using 6 far-field data with 5% random noise, and the reconstruction of D formed by points $z \in T$ satisfying $\tilde{I}(\mathbf{u}_\infty, z) \leq 0$. Similar results for the peanut, kite and square are shown in Figures 2–4, respectively. The location and size of the obstacle can be reconstructed well, taking the amount of data used into account.

**Figure 1.** The true triangle is the red line. (Left): contour plot of \tilde{I} . (Right): reconstruction D .

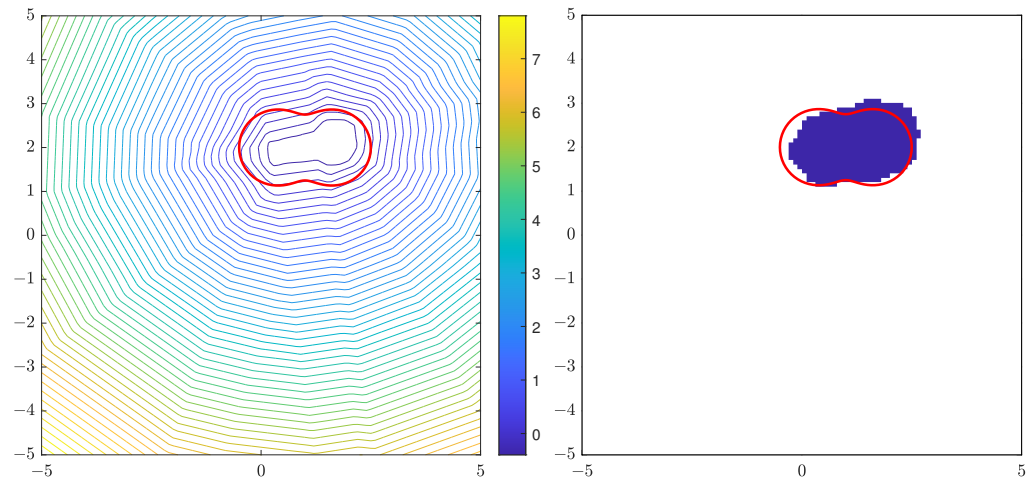


Figure 2. The true peanut is the red line. (Left): contour plot of \tilde{I} . (Right): reconstruction D .

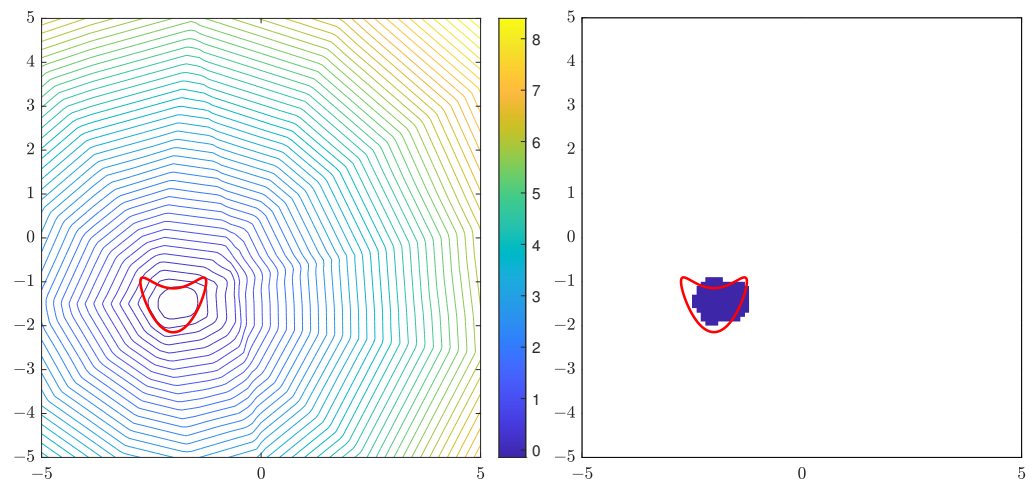


Figure 3. The true kite is the red line. (Left): contour plot of \tilde{I} . (Right): reconstruction D .

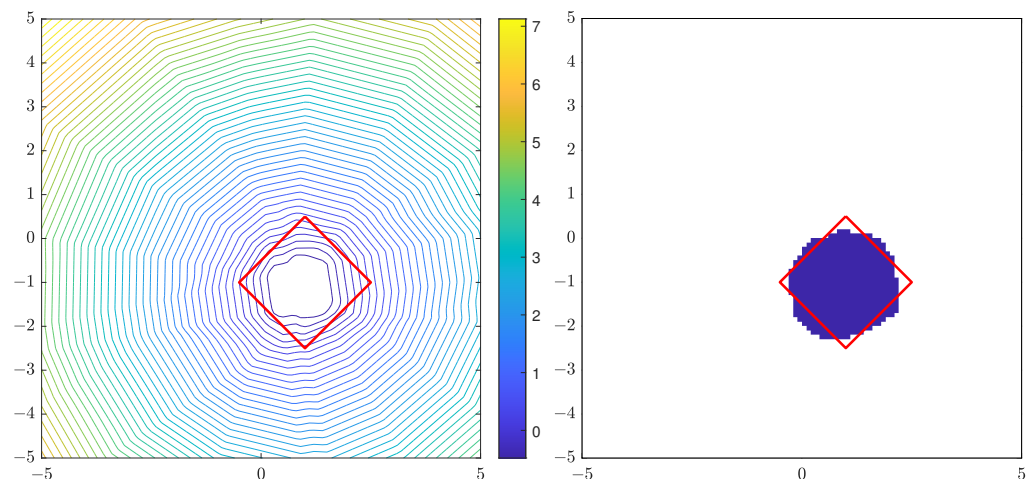


Figure 4. The true square is the red line. (Left): contour plot of \tilde{I} . (Right): reconstruction D .

4. Conclusions

We consider the reconstruction of a sound-soft obstacle using only a few far-field data due to a single incident wave. In addition to the inherent nonlinearity and ill-posedness of inverse problems, the presence of partial data introduces additional challenges for classical methods. In this paper, we propose a simple and novel DNN-oriented indicator method

for this inverse problem, utilizing an indicator function to determine whether any chosen sample point is inside the scatterer, thus identifying the scatterer. What distinguishes this method from existing sampling methods is that our indicator function no longer relies on solving integrals or integral equations. Instead, it is defined as a signed distance function approximated using standard deep neural networks.

Rather than developing sophisticated networks, we focus on the formulation of suitable inverse operators and data structures such that standard DNNs can be employed. This method maintains the simplicity and flexibility of using indicators in reconstructing scatterers. In comparison to the existing sampling methods, it leverages the advantages of deep learning methods by compensating for insufficient known information through empirical data, allowing it to work effectively even with limited far-field data. Numerical experiments demonstrate that the location and size of the obstacle are reconstructed effectively. Such a reconstruction can serve as a starting point for other methods.

Since the performance of a DNN is highly dependent on the training data, data tailored for single scatterers in the numerical implementation can only train a deep neural network suitable for the reconstruction of single scatterers. In the future, we plan to introduce training data for multiple scatterers and extend the idea of the DNN-oriented indicator method to other inverse problems involving partial data.

Author Contributions: Methodology, J.L. and J.S.; software, Y.L. and X.Y.; data curation, Y.L. and X.Y.; writing—original draft preparation, Y.L. and X.Y.; writing—review and editing, J.L. and J.S.; project administration, J.S.; funding acquisition, J.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by NSFC grant number 11801218.

Data Availability Statement: Data sharing is not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

- Colton, D.L.; Kress, R.; Kress, R. *Inverse Acoustic and Electromagnetic Scattering Theory*; Springer: London, UK, 2019; Volume 93.
- Hettlich, F. Fréchet derivatives in inverse obstacle scattering. *Inverse Probl.* **1995**, *11*, 371. [\[CrossRef\]](#)
- Johansson, T.; Sleeman, B.D. Reconstruction of an acoustically sound-soft obstacle from one incident field and the far-field pattern. *IMA J. Appl. Math.* **2007**, *72*, 96–112. [\[CrossRef\]](#)
- Colton, D.; Haddar, H. An application of the reciprocity gap functional to inverse scattering theory. *Inverse Probl.* **2005**, *21*, 383. [\[CrossRef\]](#)
- Potthast, R. A study on orthogonality sampling. *Inverse Probl.* **2010**, *26*, 074015. [\[CrossRef\]](#)
- Ito, K.; Jin, B.; Zou, J. A direct sampling method to an inverse medium scattering problem. *Inverse Probl.* **2012**, *28*, 025003. [\[CrossRef\]](#)
- Chen, J.; Chen, Z.; Huang, G. Reverse time migration for extended obstacles: Acoustic waves. *Inverse Probl.* **2013**, *29*, 085005. [\[CrossRef\]](#)
- Liu, J.; Sun, J. Extended sampling method in inverse scattering. *Inverse Probl.* **2018**, *34*, 085007. [\[CrossRef\]](#)
- Chen, X.; Wei, Z.; Maokun, L.; Rocca, P. A review of deep learning approaches for inverse scattering problems (invited review). *Prog. Electromagn. Res.* **2020**, *167*, 67–81. [\[CrossRef\]](#)
- Gao, Y.; Zhang, K. Machine learning based data retrieval for inverse scattering problems with incomplete data. *J. Inverse -Ill-Posed Probl.* **2021**, *29*, 249–266. [\[CrossRef\]](#)
- Sun, Y.; He, L.; Chen, B. Application of neural networks to inverse elastic scattering problems with near-field measurements. *Electron. Res. Arch.* **2023**, *31*, 7000–7020. [\[CrossRef\]](#)
- Yang, H.; Liu, J. A qualitative deep learning method for inverse scattering problems. *Appl. Comput. Electromagn. Soc. J. (ACES)* **2020**, *35*, 153–160.
- Guo, R.; Jiang, J. Construct deep neural networks based on direct sampling methods for solving electrical impedance tomography. *SIAM J. Sci. Comput.* **2021**, *43*, B678–B711. [\[CrossRef\]](#)
- Le, T.; Nguyen, D.L.; Nguyen, V.; Truong, T. Sampling type method combined with deep learning for inverse scattering with one incident wave. *arXiv* **2022**, arXiv:2207.10011.
- Ning, J.; Han, F.; Zou, J. A direct sampling-based deep learning approach for inverse medium scattering problems. *arXiv* **2023**, arXiv:2305.00250.
- Ruiz, Á.Y.; Cavagnaro, M.; Crocco, L. A physics-assisted deep learning microwave imaging framework for real-time shape reconstruction of unknown targets. *IEEE Trans. Antennas Propag.* **2022**, *70*, 6184–6194. [\[CrossRef\]](#)

17. Du, H.; Li, Z.; Liu, J.; Liu, Y.; Sun, J. Divide-and-conquer DNN approach for the inverse point source problem using a few single frequency measurements. *Inverse Probl.* **2023**, *39*, 115006. [[CrossRef](#)]
18. Gao, Y.; Liu, H.; Wang, X.; Zhang, K. On an artificial neural network for inverse scattering problems. *J. Comput. Phys.* **2022**, *448*, 110771. [[CrossRef](#)]
19. Heaton, J. *Introduction to Neural Networks with Java*; Heaton Research, Inc.: Chesterfield, MO, USA, 2008.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.