SIAM J. SCI. COMPUT.
Vol. 43, No. 3, pp. B678–B711

CONSTRUCT DEEP NEURAL NETWORKS BASED ON DIRECT SAMPLING METHODS FOR SOLVING ELECTRICAL IMPEDANCE TOMOGRAPHY*

RUCHI GUO† AND JIAHUA JIANG‡

Abstract. This work investigates the electrical impedance tomography problem when only limited boundary measurements are available, which is known to be challenging due to the extreme ill-posedness. Based on the direct sampling method (DSM) introduced in [Y. T. Chow, K. Ito, and J. Zou, *Inverse Problems*, 30 (2016), 095003], we propose deep direct sampling methods (DDSMs) to locate inhomogeneous inclusions in which two types of deep neural networks (DNNs) are constructed to approximate the index function (functional): fully connected neural networks and convolutional neural networks. The proposed DDSMs are easy to be implemented, capable of incorporating multiple Cauchy data pairs to achieve high-quality reconstruction and highly robust with respect to large noise. Additionally, the implementation of DDSMs adopts offline-online decomposition, which helps to reduce a lot of computational costs and makes DDSMs as efficient as the conventional DSM proposed by Chow, Ito, and Zou. The numerical experiments are presented to demonstrate the efficacy and show the potential benefits of combining DNN with DSM.

Key words. deep learning, inverse problems, direct sampling methods, electrical impedance tomography, reconstruction algorithm, limited boundary data

AMS subject classifications. 35R30, 68U10

DOI. 10.1137/20M1367350

1. Introduction.

Electrical impedance tomography (EIT) is a very promising technique for a noninvasive, radiation-free type of medical imaging. In short, by alternating electrical current at a set of electrodes and measuring the corresponding voltages on the boundary of an unknown medium (e.g., on the skin), it is possible to reconstruct the internal electrical conductivity distribution image of the medium [30, 10]. EIT has wide applications, such as biomedicine [66, 30] including monitoring of ventilation distribution [61], geophysics [65], and industrial detection [39]. The particular application considered in this work is to detect the inclusions buried in a known homogeneous background, for example, measuring the resistivity of tissue associated with malignancy, ischemia, and lung water [7, 23].

In order to describe the isotropic mathematical model for EIT, we consider a bounded domain $\Omega \subseteq \mathbb{R}^n$, $n = 2, 3$, with C^1 boundary $\partial\Omega$ occupied by some conducting materials with the electrical conductivity described by a positive function $\sigma(x) \in L^\infty(\Omega)$. Let the background homogeneous material have the conductivity σ_0, and then the support of $\sigma - \sigma_0$ indicates the inhomogeneous inclusions denoted by D. Suppose that N different electrical currents are injected to the boundary $\partial\Omega$; then the resulting electrical potential should satisfy the following N governing equations with the same coefficient but different boundary conditions:

*Submitted to the journal's Computational Methods in Science and Engineering section September 17, 2020; accepted for publication (in revised form) February 19, 2021; published electronically June 1, 2021.

https://doi.org/10.1137/20M1367350

Funding: The first author was funded by NSF DMS-2012465. The second author was funded by NSF DMS-1654175.

†Department of Mathematics, University of California, Irvine, Irvine, CA 92697 USA (ruchig@uci.edu).

‡School of Information Science and Technology, ShanghaiTech University, Shanghai, China (jiangjh@shanghaitech.edu.cn).

$$\nabla \cdot (\sigma \nabla u_\omega) = 0 \quad \text{in} \quad \Omega, \tag{1.1a}$$

$$\sigma \frac{\partial u}{\partial n} = g_\omega \quad \text{on} \quad \partial\Omega, \quad \omega = 1, 2, \ldots, N, \tag{1.1b}$$

where $g_\omega(x) \in H^{-1/2}(\partial\Omega)$ are current density such that

$$\int_{\partial\Omega} g_\omega(s) ds = 0. \tag{1.2}$$

For the simplicity of notation, we write the surface potential over $\partial\Omega$ as $f_\omega := u_\omega|_{\partial\Omega}$. Mathematically, our inverse problem associated with EIT is to recover $\sigma(x)$ in Ω, specifically the shape and the position of the inclusions, based on the Neumann boundary value (current data) and the observed Dirichlet boundary value (voltage data). These data typically refer to a collection of the Cauchy data pairs $\{(g_\omega, f_\omega)\}_{\omega=1}^N$.

Let's temporarily write $u = u_\omega$ and $g = g_\omega$ for general discussion. Then the Neumann-to-Dirichlet (NtD) map associated to (1.1) is defined by

$$\Lambda_\sigma : H^{-1/2}(\partial\Omega) \to H^{1/2}(\partial\Omega), \quad \Lambda_\sigma g = u|_{\partial\Omega}. \tag{1.3}$$

Theoretically, the conductivity distribution σ as a positive L^∞ function can be recovered from a full knowledge of the NtD map Λ_σ [4]. However, in practice, the full knowledge of the NtD map needs a matrix approximation requiring a large number of Cauchy data pairs, for example, $N = 64$ in (1.1) [16], but may be unfeasible in real-world problems. Nevertheless the stability and accuracy of the approximation also need to be resolved. In addition, in many practical situations, the conductivity coefficient σ is not as rough as L^∞. In fact the piecewise constant conductivity widely appears, and some researches [3, 33, 2, 11] have shown how one or few boundary measurement(s), i.e., the Cauchy data pairs, can reconstruct the inclusions. So from both the practical and theoretical point of view, the full knowledge of the NtD map not be very necessary, which justifies the limited Cauchy data assumption aforementioned.

It is known that a high-quality reconstruction for the EIT problem is challenging due to its severe ill-posedness. Various numerical algorithms have been developed which can be categorized into two families. The first one is based on optimization algorithm which typically constructs a sequences of coefficient distributions converging to the true one. Methods in this family include the finite-element-based methods [48, 60, 38] and finite-difference-based methods [54], shape optimization methods [27, 12, 18, 36], and so on. For these methods, in general a good initial condition and a significant number of iterations are needed to ensure convergence which also rely on suitable regularization techniques such as Tikhonov regularization [60, 38], total variation regularization [18], and L^1 regularization [37]. Alternatively, a second family of methods, i.e., the direct methods, has been developed including the sampling and factorization methods [40, 16], the multiple signal classification algorithms [3, 2], and the topological sensitivity approaches [2, 6]. These methods are noniterative in nature and thus tend to be much more efficient than optimization-type methods. Besides the classical approaches, recently, deep neural networks (DNNs) also have shown great potential for solving the EIT problem [1, 22, 29], for example, combining fast direct reconstruction procedures with convolutional neural networks (CNNs) [43, 29, 59], the radial basis function neural network [51, 63], the multilayer neural network [41], and the Bayesian neural network [42] and improving the neural network training with particle swarm optimization [45, 46].

Recently, the authors in [16, 15, 17, 14, 34, 35] developed a so-called direct sampling method (DSM) to solve not only EIT but also a large group of inclusion iden-

tification problems which are shown to be easy to implement, computationally economical, highly robust, and very effective. The principal ingredient of DSMs is to construct an index function that attains extreme values for sampling points belonging to the inclusions and thus provides the estimate of the shape. In particular, the index function for EIT is derived exquisitely in [16] showing reasonably nice indication for the inclusions. However, we also note the DSM in [16] has the potential to be further improved in several aspects. First, it is mainly focused on the case of a *single* Cauchy data pair which may hinder its application to the more complicated case due to the limited accuracy. In engineering practice [30], the number of experimental measurement data is indeed limited but mostly more than one. Second, the explicit index function in [16] relies on the domain geometry and is obtained for some basic shapes such as circle, square, and open ball. Indeed, the derivation of suitable explicit index functions may become very complicated for multiple Cauchy data pairs or more complex domain shapes. Furthermore, even with a single Cauchy data pair, the optimality may not be guaranteed for the index functions derived in [16]. We believe it may not be easy to overcome these obstacles by conventional mathematical approaches.

Alternatively, to address these hurdles numerically, in this work we propose using DNNs to replace the classical intricate derivation to construct index functions, and the resulting methods, which we call *deep direct sampling methods* (DDSMs), have a few remarkable features. First, the DDSMs are capable of naturally both incorporating multiple Cauchy data pairs and fitting for any shaped Ω, and thus they can break the accuracy limits and result in high-quality reconstruction for both location and shape. Second, the DDSMs also inherit the robustness feature of the conventional DSM with respect to large noise. More specifically, these algorithms can smooth the noise appearing in the Cauchy data and handle noise as large as 20%. In addition, the proposed methods are able to handle only a few data points (electrodes) available on the boundary without much hindering the accuracy. Moreover, the application of DDSMs almost do not rely on the conductivity magnitude in the sense that they can successfully reconstruct inclusion shapes which have much different conductivity magnitude than those used in training. Furthermore, although the training process costs more computational budget than the conventional DSM since it requires solving an optimization problem, it only needs to be operated once in the offline phase. The online computation involved in reconstruction/prediction is only the fast evaluation of the DNN-based index function which is as efficient as the DSM. The offline-online decomposition structure makes the proposed DDSMs have more optimal index function without degrading the efficiency, where the optimality is benefited from the DNNs and the efficiency is kept from the DSM. Therefore, we think the DDSMs combine the advantages of both optimization methods and direct methods.

In particular, we develop two types of DDSMs: fully connected neural network based DDSM (FNN-DDSM) and CNN-based DDSM (CNN-DDSM). Suggested by the conventional DSM, both the networks take the mathematical format of input data which are the solutions of a back projection of boundary data, i.e., an elliptic equation with the known background coefficient and the boundary condition given by a certain difference of the Cauchy data pairs. It is noted that the choices of input and output for DNNs are in general crucial for their performance. For the FNN-DDSM, we consider the index function as a pointwise indicator that classifies the points in the domain into two categories, inside the inclusions and outside the inclusions, which can be naturally treated as a classification problem. A softmax layer is then used to normalize the output of the network to a probability distribution over predicted output classes. We highlight that the probabilistic meaning of the output of FNN-

DDSM computationally hinted the connection within the EIT problem, the DSM, and the probability problem. For CNN-DDSM, we recast the index function into a functional mapping data images to EIT images, so it can be viewed as a semantic image segmentation problem that aims to extract the contour (detect the edge of inclusions) from the image. It can be also considered as a further generalization of the structure of the DSM in [16] since it uses more neighbor information of each point to predict their location.

The remainder of this paper is organized as follows. The mathematical background of the DSM [16] is prepared in section 2. Section 3 is devoted to the development of our novel DDSMs consisting of FNN-DDSM and CNN-DDSM. Our numerical results are presented in section 5. Some concluding remarks and are given in section 6. The acknowledgments are provided at the end of this paper.

2. Review of DSMs. This section summarizes some necessary theoretical background of the DSM [16] for solving the aforementioned EIT problem which also serves as the mathematical foundation of the proposed neural networks. The key idea is to derive a certain index function indicating the inclusion shape and locations, which should ideally satisfy

$$\mathcal{I}(x) = \begin{cases} 1, & x \in D, \\ 0, & x \in \Omega \backslash \overline{D}. \end{cases} \tag{2.1}$$

Since the work in [16] mainly focuses on the case of only a *single* Cauchy data pair available, we here just let (g, f) be the pair of Cauchy data measured over the surface $\partial\Omega$ with $f = \Lambda_\sigma g$ and the NtD map Λ_σ defined in (1.3). Without loss of generality, we assume the background medium has $\sigma_0 = 1$. A fundamental ingredient in the derivation of DSM is the duality product $\langle \cdot, \cdot \rangle_{\gamma,\partial\Omega}$ defined as

$$\langle \mathcal{X}, \phi \rangle_{\gamma,\partial\Omega} = \int_\Gamma (-\Delta_{\partial\Omega})^\gamma \mathcal{X} \phi ds = \langle (-\Delta_{\partial\Omega})^\gamma \mathcal{X}, \phi \rangle_{L^2(\partial\Omega)} \ \forall \mathcal{X} \in H^{2\gamma}(\partial\Omega), \forall \phi \in L^2(\partial\Omega),$$

where $\gamma \geq 0$ and $\Delta_{\partial\Omega}$ is the surface Laplacian operator. Let $|\cdot|_Y$ denote a certain seminorm in $H^{2\gamma}(\partial\Omega)$. An essential component to derive the index function is a family of probing functions $\{\eta_{x,d}\}_{x\in\Omega, d\in\mathbb{R}^n} \subset H^{2\gamma}(\partial\Omega)$ satisfying some conditions of which the critical one for us is represented below.

(C) The probing functions are almost orthogonal with each other. That is, for all $x, y \in \Omega, d_x, d_y \in \mathbb{R}^n$, the function

$$K_{d_x,d_y}(x, y) := \frac{\langle \eta_{x,d_x}, \eta_{y,d_y} \rangle_{\gamma,\partial\Omega}}{|\eta_{x,d_x}|_Y} \tag{2.2}$$

achieves maximum when $x = y$ and behaves like a sharply peaked Gaussian-like distribution, i.e., it is close to a kernel function $e^{\frac{-|x-y|^2}{a^2}}$ with small a.

We emphasize that the condition (C) is particularly helpful for us to choose suitable activation functions in the design of networks. The construction of suitable probing functions can be challenging in general. The probing functions introduced in [16] are based on the dipole potential [20, 25], and they satisfy

$$\eta_{x,d}(\xi) := w_{x,d}(\xi), \quad \xi \in \partial\Omega, \tag{2.3}$$

where $w_{x,d}$ is the solution of the following problem:

$$-\Delta w_{x,d} = -d \cdot \nabla \delta_x \quad \text{in} \quad \Omega, \quad \frac{\partial w_{x,d}}{\partial n} = 0 \quad \text{on} \quad \partial\Omega, \quad \int_{\partial\Omega} w_{x,d} ds = 0 \tag{2.4}$$

with given $x \in \Omega, d \in \mathbb{R}^n$ and $\delta_x(\xi)$ being the delta function for each x. With the probing functions, the index function can be then defined as

$$\mathcal{I}(x, d_x) := \frac{\langle \eta_{x,d_x}, f - \Lambda_{\sigma_0} g \rangle_{\gamma,\partial\Omega}}{||f - \Lambda_{\sigma_0} g||_{L^2(\partial\Omega)} |\eta_{x,d_x}|_Y}, \quad x \in \Omega, d_x \in \mathbb{R}^n. \tag{2.5}$$

With the definition of duality product $\langle \cdot, \cdot \rangle_{\gamma,\partial\Omega}$ in (2), the surface Laplacian operator only performs on the probing function $\eta_{x,d}$ that is itself infinitely smooth over the measurement surface. So the index function in (2.5) is well defined even for very irregular data $f - \Lambda_\sigma g$ containing very rough noise, for example, those data only in $L^2(\partial\Omega)$. As mentioned in [16], due to this feature the noise appearing in data can be directly smoothed by the duality product over the measurement surface, and it is the reason of the high robustness of the DSM against noise. We believe it is also one of the reasons that our DDSMs are highly stable with respect to noise as observed in section 5.

Note that (2.5) is not computable yet since the probing direction $d_x \in \mathbb{R}^n$ remains unknown. In order to find an appropriate direction d_x, an alternative characterization of the index function is considered in [16]. Define ϕ as the solution to the standard elliptic equation with the boundary condition $(-\Delta_{\partial\Omega})^\gamma (f - \Lambda_{\sigma_0} g)$:

$$-\Delta\phi = 0 \quad \text{in} \quad \Omega, \quad \frac{\partial\phi}{\partial n} = (-\Delta_{\partial\Omega})^\gamma (f - \Lambda_{\sigma_0} g) \quad \text{on} \quad \partial\Omega, \quad \int_{\partial\Omega} \phi ds = 0. \tag{2.6}$$

We highlight that the function ϕ in (2.6) is particularly important for designing our DNNs and processing given boundary data to the input (images) of the DNNs. Through (2.6), the index function in (2.5) can be equivalently rewritten as

$$\mathcal{I}(x, d_x) := \frac{d_x \cdot \nabla\phi(x)}{||f - \Lambda_{\sigma_0} g||_{L^2(\partial\Omega)} |\eta_{x,d_x}|_Y}, \quad x \in \Omega, d_x \in \mathbb{R}^n. \tag{2.7}$$

It is easy to see that the above index function reaches its maximum value at $d_x = \frac{\nabla\phi(x)}{|\nabla\phi(x)|}$ for each $x \in \Omega$ which is the appropriate choice of the probing direction d_x. As observed in the experiments in [16], such choice of d_x is crucial since, for example, the existing inclusions might be accidentally removed if d_x is wrongly chosen orthogonal to $\nabla\phi(x)$ at a point $x \in \Omega$. Thus $\mathcal{I}(x, d_x) = \mathcal{I}(x, \nabla\phi(x))$ which is the guideline for us to construct DNNs.

Notice that computing the index function (2.7) with $d_x = \nabla\phi(x)$ is only up to solving the probing functions from (2.3) and (2.5), and this is challenging since the probing functions vary with respect to $x \in \Omega$. Instead of directly solving (2.3) and (2.5) that may slow down the DSM, the authors in [16] derive the explicit forms of the probing functions with very delicate mathematical skills for some special domain:

- Ω is a circular domain:

$$\eta_{x,d}(\xi) = \frac{1}{\pi} \frac{(\xi - x) \cdot d}{|x - \xi|^2}, \quad \xi \in \partial\Omega. \tag{2.8}$$

- Ω is an open ball:

$$\eta_{x,d}(\xi) = \frac{d \cdot \xi - \frac{(x-\xi)\cdot d}{|x-\xi|}}{\sqrt{4\pi}(|x - \xi| - x \cdot \xi + 1)}, \quad \xi \in \partial\Omega. \tag{2.9}$$

The simple and explicit probing functions $\eta_{x,d}(\xi)$ in (2.8)–(2.9) are then put into (2.7), and thus the index function $\mathcal{I}(x, \nabla\phi(x))$ can be computed efficiently without solving any PDEs. So it is essentially different from optimization-based iterative methods which require repeatedly solving forward problems and adjoint problems for many times.

3. DDSMs. The whole design procedure of the index function of the DSM is delicate and intellectual. Despite these features and the successful application, the accuracy and further application of the DSM are limited mainly by the following two aspects. First, the derivation above is mainly focused on a *single* Cauchy data pair. When multiple pairs of Cauchy data are applied on Ω with any geometry, it is unclear so far how to systematically develop an index function incorporating all data by canonical mathematical derivation, though some basic operations can be applied such as average or maximum of each individual index function. Second, the design of $\mathcal{I}(x, d_x)$ is not necessarily optimal. For example, the tuning parameter γ and norm $|\cdot|_Y$ need to be chosen empirically, and the format itself may not be the best approximation to the true index function. So it motivates us to use DNN models to learn the index functions through a large number of data since DNNs are able to mimic the human learning process based on physical data.

Therefore, to make DSMs applicable to more general situations of EIT problems and ameliorate the quality of reconstruction, in this section we propose *DDSMs*. The major difficulty is on suitable design of DNN architectures. We build up the architectures based on two different perspectives to understand the mathematical form of the index function (2.7), which are the topics of the next two subsections. The first algorithm, FNN-DDSM, is from the perspective of approximating the index function $\mathcal{I}(x, d_x)$ in (2.7) by training it from coordinates x and values of $\nabla\phi^\omega (1 \le \omega \le N)$, which can be treated as a binary classification problem aiming to categorize the coordinates into two groups—inside of the inclusions and outside of the inclusions . The second one, CNN-DDSM, is from the angle of functional or operator approximation, where the goal is to train the mapping from Cauchy difference functions $\{\phi^\omega\}_{\omega=1}^N$ to the inclusion distribution. This can be viewed as an image segmentation problem if the input $(x, \phi^1, \ldots, \phi^N)$ is considered as a $(N+2)$-channel image. FNN-DDSM and CNN-DDSM are intrinsically different both in their network architectures and their mathematical foundations, yet our numerical experiments show that they both work very well, enhancing accuracy and stability without sacrificing efficiency.

For simplicity we mainly discuss our methods for the two-dimensional Ω, and they can be naturally extended to the three-dimensional case. To avoid repetition, we first prepare some notations which will be frequently used in the following discussion.

- $\mathcal{N}_h$: the discretization of domain Ω that consists of a group of discrete points, i.e., $\mathcal{N}_h = \{x^k = (x_1^k, x_2^k)\}_{k=1}^K$, where K is the total number of points.
- ϕ^ω with $1 \le \omega \le N$: the solutions of (2.6) with the boundary value from the ωth pair of Cauchy data, i.e., $f_\omega - \Lambda_{\sigma_0} g_\omega$. Since these functions are so critical in both the DSM and the proposed DDSMs, we shall call them *Cauchy difference functions*.

If the training data samples are involved in the context, we further define the following notations.

- S: the number of inclusion samples (coefficient distribution samples) in the training sets.
- $\mathcal{I}^s$ with $1 \le s \le S$: the true index function associated with the sth inclusion sample.

- $\phi^{(s,\omega)}$ with $1 \le s \le S$ and $1 \le \omega \le N$: the *Cauchy difference functions* associated with the ωth pair of Cauchy data generated by the sth inclusion sample.

3.1. FNN-DDSM. We first discuss the construction of FNN-DDSM. The derivation of $\mathcal{I}(x, d_x)$ in (2.7) suggests the existence of a nonlinear mapping from x, d_x to the location of x, i.e., whether it is inside or outside of the inclusions. Furthermore, inspired by the choice of d_x, we assume that the direction d_x is a function of $\nabla\phi^\omega(x)$ for all the Cauchy pairs. Therefore, we assume that the new index function takes the form

$$\mathcal{I}(x) = \mathcal{F}_{\text{FNN}}(x, \nabla\phi^1(x), \dots, \nabla\phi^N(x)), \quad x \in \Omega, \tag{3.1}$$

where $\mathcal{F}_{\text{FNN}}$ is trained by a fully connocted neural network (FNN) described in details below. Mathematically speaking, $\mathcal{F}_{\text{FNN}}$ is a nonlinear high-dimensional function mapping a data point in $\mathbb{R}^{2N+2}$ to the index of the associated point in Ω. Notice that the design of (3.1) is directly motivated by the form of (2.7) with multiple Cauchy data pairs. But at this moment we do not assume $\mathcal{F}_{\text{FNN}}$ takes any a priori format such as (2.7) and only make the assumptions on its input. So we believe the proposed new index function is more general than the index function (2.7) even with $N = 1$. Since the value of the index function (3.1) shows the possibility of the input coordinate x falling in the inclusions, where the larger value means that the point x is more possible to be in the inclusions, approximating this index function can be treated as training a binary classifier that aims to categorizes the input points into two groups—inside of the inclusions and outside of the inclusion. This point-categorization nature motivates training an FNN as a classifier and the employment of a cross entropy loss function discussed below.

Now we proceed to describe the structure of the proposed FNN. Given an inclusion sample and the associated Cauchy difference functions ϕ^ω, we let the input of the FNN, denoted by $\boldsymbol{z}_{\text{in}} \in \mathbb{R}^{2(N+1)}$ (depicted in Figure 1(a)), contain the coordinates of x in the first two components and the value of $\nabla\phi^\omega(x)$ in the next $2N$ components, i.e.,

$$[\boldsymbol{z}_{\text{in}}]_1 = x_1, \tag{3.2a}$$

$$[\boldsymbol{z}_{\text{in}}]_2 = x_2, \tag{3.2b}$$

$$[\boldsymbol{z}_{\text{in}}]_{3,\dots,N+2} = \begin{bmatrix}\partial_x\phi^1(x_1,x_2) & \dots & \partial_x\phi^N(x_1,x_2)\end{bmatrix}^\top, \tag{3.2c}$$

$$[\boldsymbol{z}_{\text{in}}]_{N+3,\dots,2N+2} = \begin{bmatrix}\partial_y\phi^1(x_1,x_2) & \dots & \partial_y\phi^N(x_1,x_2)\end{bmatrix}^\top. \tag{3.2d}$$

If the length of the input $\boldsymbol{z}_{\text{in}}$ is smaller than the width of the input layer, to handle this discrepancy we can pad $\boldsymbol{z}_{\text{in}}$ by a zero vector. The corresponding input linear layer is defined as

$$\psi_{\text{in}}(\boldsymbol{z}_{\text{in}}) = W_{\text{in}}\boldsymbol{z}_{\text{in}} + \boldsymbol{b}_{\text{in}}, \tag{3.3}$$

where W_{in} is the weight matrix associated with the synapses connecting the input layer neurons to the neurons in the current layer and $\boldsymbol{b}_{\text{in}}$ is the bias. In the architecture that we use, except input layer and output layer, each layer is associated with a block, and each block is constructed by stacking several layers including two linear transformations, two activation functions, and a residual connection. In particular, suppose $\boldsymbol{z}_i$ is the input of the ith block then the ith block can be expressed as

$$\tau_i(\boldsymbol{z}_i) = \varphi(W_{i,2}\ \varphi(W_{i,1}\boldsymbol{z}_i + \boldsymbol{b}_{i,1}) + \boldsymbol{b}_{i,2}) + \boldsymbol{z}_i, \tag{3.4}$$

where $W_{i,1}, W_{i,2}, \boldsymbol{b}_{i,1}, \boldsymbol{b}_{i,2}$ are weight matrices and bias vectors. The output linear layer is

$$\psi_{\text{out}}(\boldsymbol{z}_{\text{out}}) = W_{\text{out}}\boldsymbol{z}_{\text{out}} + \boldsymbol{b}_{\text{out}}, \tag{3.5}$$

where $W_{\text{out}}, \boldsymbol{b}_{\text{out}}$ also denote weight matrices and bias vectors for this layer. Let Θ denote the collection of the unknown parameters in these matrices and vectors to be learned in training. The complete FNN with M blocks can now be represented as

$$\boldsymbol{y}_{\text{out}}(\boldsymbol{z}_{\text{in}}) = \kappa \circ \psi_{\text{out}} \circ \tau_M \circ \cdots \circ \tau_1 \circ \psi_{\text{in}}(\boldsymbol{z}_{\text{in}}), \tag{3.6}$$

where κ is the softmax layer. The entire structure including blocks and residual connections is shown in Figure 1.

In each block, we use the clipped rectified-linear (ReLu) activation function $\varphi(z) = \min\{\max\{0, z\}, 0.1\}$ to mimic the distribution of coefficients. Since the softmax layer normalizes the output of a network to a probability distribution over predicted output classes, the output $\boldsymbol{y}_{\text{out}}(\boldsymbol{z}_{\text{in}}) \in \mathbb{R}^{1\times 2}$ consists of two components within $[0, 1]$ with the unit sum, i.e.,

$$[\boldsymbol{y}_{\text{out}}(\boldsymbol{z}_{\text{in}})]_1,\ [\boldsymbol{y}_{\text{out}}(\boldsymbol{z}_{\text{in}})]_2 \in [0,1], \qquad [\boldsymbol{y}_{\text{out}}(\boldsymbol{z}_{\text{in}})]_1 + [\boldsymbol{y}_{\text{out}}(\boldsymbol{z}_{\text{in}})]_2 = 1, \tag{3.7}$$

where $[\boldsymbol{y}_{\text{out}}(\boldsymbol{z}_{\text{in}})]_1$ indicates the probability of the point $x = (x_1, x_2)$ locating inside the inclusion. Namely, the larger $[\boldsymbol{y}_{\text{out}}(\boldsymbol{z}_{\text{in}})]_1$ (closer to 1), the more possible that the point is inside the inclusion, and vice versa. Otherwise, it means that (x_1, x_2) is more likely to be outside of the inclusions.

It is remarked that the value of the index function (2.7) of the conventional DSM is not exactly 0 or 1 either. It naturally motivates some researchers to think that

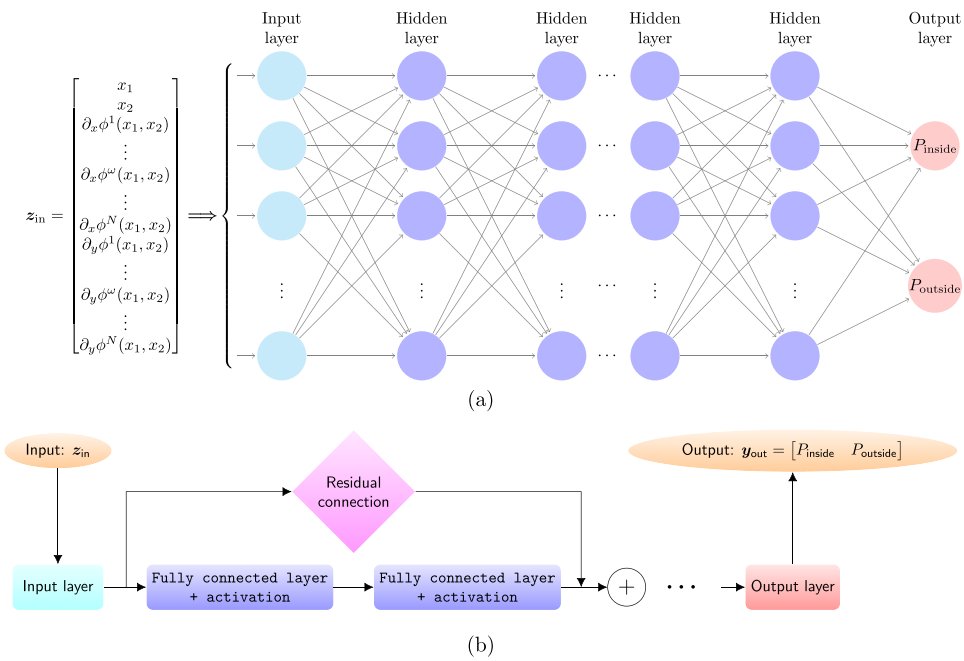

FIG. 1. *The structure of FNN. P_{inside} indicates the probability of a point $x = (x_1, x_2)$ being inside the inclusions. $P_{outside}$ indicates the probability of a point $x = (x_1, x_2)$ being outside the inclusions.*

the DSM is more like evaluating the chance of each mesh point situated within the inclusions. This feature is essentially different from some conventional optimization methods such as the shape optimization from [18, 12] that gives the sharp interface of inclusions. But notice that there is no rigorous theoretical analysis on the relationship between the DSM and probability. From this point of view, we think the proposed FNN in a certain sense hints at a connection between the DSM for EIT problems and probability due to the clear probabilistic meaning of its output $\boldsymbol{y}_{\text{out}}(\boldsymbol{z}_{\text{in}})$. In addition, due to the offline optimization stage of the DDSMs, the estimate of the probabilistic distribution is also much more accurate.

It is well known that the cross entropy is a suitable loss function for classification problems, since it minimizes the distance between two probability distributions—predicted and actual. In order to employ it to evaluate the proposed FNN, based on the notations introduced at the beginning of this section, we first consider the following loss function:

$$\mathcal{L}_{\text{loss}}(\Theta) = \frac{1}{S}\sum_{s=1}^{S}\left(\frac{1}{K}\sum_{k=1}^{K}\mathcal{I}^s(x^k)\log\left(\left[\boldsymbol{y}_{\text{out}}\left(\boldsymbol{z}_{\text{in}}^{(s,k)}\right)\right]_1\right)\right. \\ \left. + (1-\mathcal{I}^s(x^k))\log\left(\left[\boldsymbol{y}_{\text{out}}\left(\boldsymbol{z}_{\text{in}}^{(s,k)}\right)\right]_2\right)\right), \tag{3.8}$$

where $\mathcal{I}^s(x^k)$ denotes the sth true index function evaluated at the kth point x^k and $\boldsymbol{z}_{\text{in}}^{(s,k)}$ denotes the input formed by the kth point x^k and the sth Cauchy difference function, i.e., the ϕ^ω in (3.2) being $\phi^{(s,\omega)}$. However, we note that there are in general a large amount of inclusion samples and discrete points to capture the geometrical details of inclusions. To reduce the computational burden and achieve faster iterations, we only use a random subset of inclusion samples and discrete points at each iteration. We employ stochastic gradient descent (SGD) [19] to update the weight matrices and bias vectors according to the following formula:

$$\Theta^{(j+1)} = \Theta^{(j)} - \alpha\nabla_\Theta\mathcal{G}_{\text{loss}}(\Theta^{(j)}), \tag{3.9}$$

where α indicates the learning rate and j is the number of iteration. $\mathcal{G}_{\text{loss}}(\Theta)$ is defined as

$$\mathcal{G}_{\text{loss}}(\Theta) = \frac{1}{|S_r|}\sum_{s\in S_r}\left(\frac{1}{|K_r|}\sum_{k\in K_r}\mathcal{I}^s(x^k)\log\left(\left[\boldsymbol{y}_{\text{out}}\left(\boldsymbol{z}_{\text{in}}^{(s,k)}\right)\right]_1\right)\right. \\ \left. + (1-\mathcal{I}^s(x^k))\log\left(\left[\boldsymbol{y}_{\text{out}}\left(\boldsymbol{z}_{\text{in}}^{(s,k)}\right)\right]_2\right)\right), \tag{3.10}$$

where S_r, K_r are random subsets of the integer sets $\{1,2,\ldots,S\}$ and $\{1,2,\ldots,K\}$, respectively. More precisely, for each iteration, we randomly choose $|S_r|$ samples in all S inclusion samples and $|K_r|$ points from all K discrete points. The gradient of $\mathcal{G}_{\text{loss}}$ is computed by backpropagation [52].

In addtion, we note that the index function (2.6) in the original DSM can be viewed as a fractional function of $\mathbf{z} = [z_1, z_2] \in \mathbb{R}^2$ with $z_1 = \partial_{x_1}\phi(x)$ and $z_2 = \partial_{x_2}\phi(x)$. So the index function in the FNN-DDSM might not be able to exactly recover the fractional function due to the composite linear transformation and the activation function. However, by Theorem 2 in [58], there indeed exists an FNN with some proper parameters which is arbitrarily close to the original index function (2.6). We believe it may be one of the reasons that the reconstruction results of the

FNN-DDSM with a single measurement have similar behavior to the index function in the conventional DSM. More discussions and results about this are provided in section 5.

Implementation issues. We emphasize the proposed DDSMs are very easy to implement based on the current sophisticated DNN packages. One only needs to prepare the data including the true index functions and Cauchy difference functions for the inclusion samples, and these data are only needed at some discrete points $\mathcal{N}_h$ of Ω. In principle these points do not need to be determined or fixed a priori, and instead they can be completely randomly chosen during each iteration. But this approach gives an extra computation burden to evaluate $\nabla\phi^\omega(x)$ since the functions ϕ^ω are in general numerically solved without an analytical formula. Therefore, for simplicity's sake, we propose directly using the discretization points involved in the computation of ϕ as $\mathcal{N}_h$ for training since the values of $\nabla\phi^\omega(x)$ at these points can be predetermined and thus do not need be computed during training. For example, if the popular finite element method is used to compute ϕ^ω, then the underlying mesh points can be used as the discretization points for training. Furthermore, we mention that the number of these points is related to the resolution of the inclusion geometry. In general, the more complicated geometry requires more points to resolve, but it also makes the training more difficult. We notice that FNN-DDSM requires the computation of each discretization point which can be accelerated by parallel techniques.

3.2. CNN-DDSM. In this subsection, we present the CNN-DDSM which is essentially different from the FNN-DDSM in both the computation and mathematical foundation. Our motivation consists of multiple levels. First, while being structure agnostic makes FNN broadly applicable, such networks do tend to require a larger number of parameters exacerbated by the fully connected structure. Second and more importantly, the gradient operator used in (3.1) can be considered as a special convolutional operator utilizing the neighboring pixels for the operation in the context of image process. More specifically, if we define $\boldsymbol{A}$ as the source image, the finite difference approximation for horizontal derivative and the vertical derivative are given by

$$\boldsymbol{G}_{x_1} = \frac{1}{h}\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} * \boldsymbol{A} \qquad \text{and} \qquad \boldsymbol{G}_{x_2} = \frac{1}{h}\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} * \boldsymbol{A}, \tag{3.11}$$

where $*$ denotes the convolution operation. If we consider $(\phi^1(x), \ldots, \phi^N(x))$ as an image with N features (channels) and treat x as pixels, then (3.1) can be understood as the convolution in (3.11) operating on these images. Moreover, from the discretization perspective, gradient or (3.11) only involves the information of the direct neighbor points of a point x to predict the index value at this point. We note that the true situation may be much more complicated than this, i.e., it may involve more neighbor terms and the operation may be more complicated than gradients (weights may not be those in (3.11)). Thus, it is reasonable to increase the matrix size in (3.11) and treat their special weights as unknown parameters to be learned. All these considerations motivate us to hybridize CNN and DSM, and our CNN-DDSM naturally arises.

Mathematically speaking, different from FNN-DDSM, in CNN-DDSM the key is to understand the conventional DSM (2.7) from a novel perspective that the desired index functions can be viewed as a functional or operator from Cauchy difference functions $\{\phi^\omega\}_{\omega=1}^N$ to the inclusion distribution instead of a function from spatial variables to the indexes 0 or 1, i.e.,

$$\mathcal{I} = \mathcal{F}_{\text{CNN}}(x, \phi^1, \ldots, \phi^N) \ : \ \left[H^1(\Omega)\right]^{2N+2} \to L^2(\Omega). \tag{3.12}$$

By (3.12), we only assume the index functionals rely on the entire set of Cauchy difference functions rather than their values at a specific point, which is certainly more relaxed than the assumption of FNN-DDSM. This structure actually agrees with the theory discussed in next section. Based on the previous discussion, we can use CNN to approximate (3.12). Additionally, from image processing perspectives, if we consider $\mathcal{I}$ as the dense prediction of a $(N+2)$-channel image $(x, \phi^1, \ldots, \phi^N)$, then the nonlinear functional $\mathcal{F}_{\text{CNN}}$ can be also treated as semantic image segmentation process [13, 31] that is partitioning a digital image into multiple segments (set of pixels) based on two characteristics: inside or outside the inclusions. It suggests a relationship between DSM for EIT problems and semantic image segmentation problems which is illustrated in Table 1 for readers from different background.

In order to describe the structure of the CNN-DDSM, for simplicity we first assume Ω has a rectangular shape and leave the general situation to the later discussion about the implementation details. Then we suppose Ω is discretized by an $n_1 \times n_2$ Cartesian grid, where n_1 and n_2 are for the x_1 and x_2 direction, respectively. Based on the previous explanation, the input to the CNN is not a vector but a 3D matrix. In particular, let's focus on an inclusion sample with N Cauchy difference functions $\{\phi^\omega\}_{\omega=1}^N$ solved with $\{(g_\omega, f_\omega)\}_{\omega=1}^N$. Then the input denoted by $\boldsymbol{z}_{\text{in}} \in \mathbb{R}^{n_x \times n_y \times (N+2)}$ is a stack of $N+2$ matrices in $\mathbb{R}^{n_1 \times n_2}$, where the first two slices are formed by spatial coordinates x_1 and x_2, respectively, and the remaining N slices correspond to the numerical solutions $\phi^1(x)$, $\phi^2(x)$, $\ldots$, $\phi^N(x)$ evaluated at the Cartesian grid points; its pictorial elucidation is provided in Figure 2.

The proposed CNN architecture is composed of convolution networks and transposed convolution networks, where the detailed configuration is illustrated in Figure 2. The convolution part consists of several blocks, and each block includes convolution layers, activation layers, and max-pooling layers. In particular, the max-pooling layers mainly help in extracting the sharpest features of the input image and reducing the input image size by computing the maximum over each nonoverlapping rectangular region. The ith block in the convolution part can be expressed as

TABLE 1
Relationship between DSM for EIT problems and image segmentation problems.

	DSM for EIT	Image segmentation
x	Mesh point in Ω	Pixel
$\phi^\omega(x)$	Cauchy difference functions	Image features (channels)
$\mathcal{I}$	Index of inclusion distribution	Dense prediction

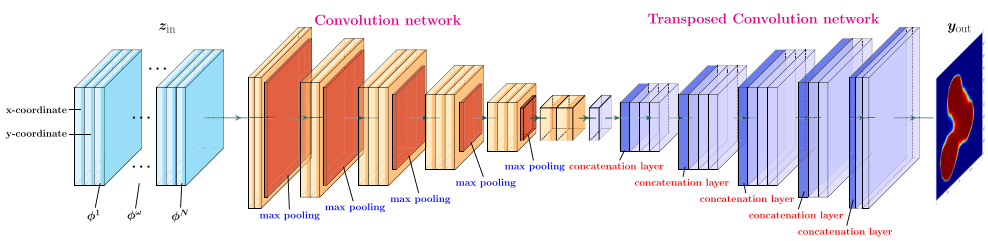

FIG. 2. *The structure of CNN. The input is a 3D matrix in* $\mathbb{R}^{n_1 \times n_2 \times (N+2)}$*, and the output is a matrix in* $\mathbb{R}^{n_1 \times n_2}$.

$$\tau_i^c(\boldsymbol{z}_{\text{conv}}) = \mathcal{M}(\zeta(\varrho(W_{\text{conv}} * \boldsymbol{z}_i + \boldsymbol{b}_{\text{conv}}))), \tag{3.13}$$

where $*$ denotes the convolution operation, W_{conv} refers to the convolution filter for the 2D convolutional layer, $\boldsymbol{b}_{\text{conv}}$ denotes the bias, $\boldsymbol{z}_i$ is the input image, $\mathcal{M}$ is the max-pooling layer, ζ denotes the activation, and ϱ is batch normalization [32] layer that aims to accelerate the training and reduce the sensitivity of the network initialization. The transposed convolution part also contains several blocks each including a transposed convolution to extrapolate the output of the convolution part to an image with large size (higher resolution). A typical example of the ith block is

$$\tau_i^t(\boldsymbol{z}_{\text{trans}}) = \mathcal{C}(\zeta(\mathcal{T}(\boldsymbol{z}_i, W_{\text{trans}}, \boldsymbol{b}_{\text{trans}}))), \tag{3.14}$$

where $\mathcal{T}$ refers to the transposed convolution operator, W_{trans} and $\boldsymbol{b}_{\text{trans}}$ are the corresponding transposed convolutional filter and the bias, $\mathcal{C}$ is a concatenation layer, and other notations are as the same as (3.13). Again Θ denotes the set of all the unknown parameters including the convolutional and transposed convolutional filters and the biases to be learned in training. Notice that convolution and batch normalization layers are also inserted into (3.14) at some blocks. Our experience and condition (C) have suggested choosing sigmoid function as the activation function,

$$\zeta(z) = \frac{1}{1+e^{-z}}. \tag{3.15}$$

We think one explanation is that the function (2.2) behaves like a sharply peaked behavior $e^{-\frac{|x-y|^2}{a^2}}$ with small a [16], namely, it decays rapidly when x moves away from y, that is also the behavior of derivative of the chosen sigmoid function. In addition, the shape of sigmoid function is similar to the Heaviside function that mimics the discontinuity of the inclusion distribution.

Then, the full CNN model can be represented as

$$\boldsymbol{y}_{\text{out}}(\boldsymbol{z}_{\text{in}}) = \tau_{M_t}^t \circ \cdots \circ \tau_1^t \circ \tau_{M_c}^c \circ \cdots \circ \tau_1^c(\boldsymbol{z}_{\text{in}}), \tag{3.16}$$

where the output $\boldsymbol{y}_{\text{out}}(\boldsymbol{z}_{\text{in}})$ is a $n_1 \times n_2$ matrix which is supposed to approximate an inclusion distribution.

To measure the accuracy of the CNN model (3.16), we employ the mean squared error (MSE) as the loss function

$$\mathcal{L}_{\text{loss}}(\Theta) = \frac{1}{S}\sum_{s=1}^{S}\big((\boldsymbol{y}_{\text{out}}(\boldsymbol{z}_{\text{in}}^s) - \mathcal{I}^s)^\top(\boldsymbol{y}_{\text{out}}(\boldsymbol{z}_{\text{in}}^s) - \mathcal{I}^s)\big), \tag{3.17}$$

where $\mathcal{I}^s$ is the true distribution (index function) corresponding to the sth inclusion sample and $\boldsymbol{z}_{\text{in}}^s$ is the input image (3D matrix) also corresponding to the sth inclusion sample, namely, the Cauchy difference functions $\phi^\omega = \phi^{(s,\omega)}$. Similarly, to overcome the infeasibility of gradient descent algorithm when training data size is huge, we apply SGD to find the minimization of the loss function (3.17):

$$\Theta^{(j+1)} = \Theta^{(j)} - \alpha\nabla_\Theta\mathcal{G}_{\text{loss}}(\Theta^{(j)}) \tag{3.18}$$

where α indicates the learning rate and j is the number of iteration. $\mathcal{G}_{\text{loss}}(\Theta)$ is defined as

$$\mathcal{G}_{\text{loss}}(\Theta) = \frac{1}{|S_r|}\sum_{s\in S_r}\big((\boldsymbol{y}_{\text{out}}(\boldsymbol{z}_{\text{in}}^s) - \mathcal{I}^s)^\top(\boldsymbol{y}_{\text{out}}(\boldsymbol{z}_{\text{in}}^s) - \mathcal{I}^s)\big), \tag{3.19}$$

where S_r is a subset of $\{1, 2, \ldots, S\}$ randomly chosen for each iteration.

Implementation issues. Different from FNN-DDSM, the discretization points of Ω have to be chosen and fixed a priori. In order to perform convolution, they have to be Cartesian grid points which are natural for rectangular domain. For a domain with general shape, we only need to immerse it into a rectangle such that the Cartesian grid can be generated on the whole rectangle. If Cauchy difference functions ϕ^ω are computed by finite element methods on a general triangulation of Ω, then the values ϕ^ω at the mesh points cannot be directly used for CNN computation, though they are already available. Instead we need to recompute their values at the newly generated Cartesian grid points. Since CNN-DDSM is approximating the index functional (3.12), the computation may not be pointwise parallelized. But many other parallelization techniques for CNN [62, 44, 64] can be used to resolve this problem if large number discretization points are needed.

4. Existence of the index function. In this section, we provide some theoretical justification on the existence of a functional $\mathcal{I}$ from the Cauchy difference functions ϕ^ω, $\omega = 1, 2, \ldots$, to the inclusion distribution given that all the Cauchy pairs $\{f_\omega, g_\omega\}$ are known, i.e., the case of full data. Then this theoretical result can serve as the mathematical foundation of the proposed neural networks. From the theory, we can also observe some other nice properties of the index functions which will be validated in the numerical experiments in the next section.

The main technique we shall use is the characterization approach developed in [8, 9]. Let's first introduce some notations and fundamental results. Define the spaces

$$H_0^{\pm 1/2}(\partial\Omega) = \left\{ v \in H^{\pm 1/2}(\partial\Omega) \ : \ \int_{\partial\Omega} uds = 0 \right\}. \tag{4.1}$$

Using (2.3) and (2.4) and imposing the normalization condition $\int_{\partial\Omega} uds = 0$, we immediately have that Λ_σ is defined from $H_0^{-1/2}(\partial\Omega)$ to $H_0^{1/2}(\partial\Omega)$. From the well known results in [8, 24], we know $\Lambda_\sigma : H_0^{-1/2}(\partial\Omega) \to H_0^{1/2}(\partial\Omega)$ is compact, self-adjoint, and positive, and so is $\Lambda_{\sigma-\sigma_0}$ if $\sigma > \sigma_0$. Therefore, by the spectral theorem, there exists a countably infinite orthogonal basis $\{v_\omega\}_{\omega=1}^\infty$ of $H_0^{-1/2}(\partial\Omega)$ that form the eigenvectors of $\Lambda_{\sigma-\sigma_0}$, namely,

$$(\Lambda_\sigma - \Lambda_{\sigma_0}) v_\omega = \lambda_\omega v_\omega, \tag{4.2}$$

where $\lambda_\omega \geq 0$ are the corresponding eigenvalues with $\lambda_\omega \to 0$ as $\omega \to \infty$. Now we can show the following theorem.

THEOREM 4.1. *Let $\{g_\omega\}_{\omega=1}^\infty$ be a fixed orthonormal basis of $H^{-1/2}(\partial\Omega)$. Given an arbitrary σ such that $\sigma > \sigma_0$ or $\sigma < \sigma_0$, let $\{g_\omega, \Lambda_\sigma g_\omega\}_{\omega=1}^\infty$ be the Cauchy data pairs and let $\{\phi^\omega\}_{\omega=1}^\infty$ be the corresponding Cauchy difference functions with $\gamma = 0$. Then the inclusion distribution σ can be purely determined from $\{\phi^\omega\}_{\omega=1}^\infty$.*

Proof. Let's first assume the applied data g_ω are just the eigenfunctions v_ω, and denote ϕ_v^ω as the corresponding Cauchy functions. Then, since $\sigma > \sigma_0$, using Theorem 3.1 in [8] or Theorem A of [9], we have $x \in D$ if and only if the probing function $\eta_{x,d} \in \mathcal{R}((\Lambda_\sigma - \Lambda_{\sigma_0})^{1/2})$, where $\mathcal{R}(\cdot)$ denotes the range of an operator, that is, by the Picard criterion [21], further equivalent to the convergence of following series:

$$\mathcal{S}(x; \{v_\omega\}_{\omega=1}^\infty) = \sum_{\omega=1}^\infty \frac{(\eta_{x,d}, v_\omega)_{\partial\Omega}^2}{\lambda_\omega} < \infty, \tag{4.3}$$

where $(\cdot,\cdot)_{\partial\Omega}$ denotes the standard L^2 inner product on $\partial\Omega$. Since $\lambda_\omega > 0$, using (4.2) we have

$$(\eta_{x,d}, v_\omega)_{\partial\Omega} = \frac{1}{\lambda_\omega}\int_{\partial\Omega} \eta_{x,d}(\Lambda_\sigma - \Lambda_{\sigma_0})v_\omega \, ds. \tag{4.4}$$

By the definition of the Cauchy difference functions in (2.6) with $\gamma = 0$, the definition of probing functions in (2.3) and (2.4), and using Green's identity, we obtain from (4.4) that

$$\begin{aligned}(\eta_{x,d}, v_\omega)_{\partial\Omega} &= \frac{1}{\lambda_\omega}\int_{\partial\Omega} \eta_{x,d}\frac{\partial \phi_v^\omega}{\partial \mathbf{n}} ds = \frac{1}{\lambda_\omega}\int_{\partial\Omega} w_{x,d}\frac{\partial \phi_v^\omega}{\partial \mathbf{n}} ds \\ &= \frac{1}{\lambda_\omega}\left(\int_{\partial\Omega} \phi_v^\omega \frac{\partial w_{x,d}}{\partial \mathbf{n}} ds + \int_\Omega w_{x,d}\triangle\phi_v^\omega dx - \int_\Omega \phi_v^\omega \triangle w_{x,d} dx\right) \\ &= -\frac{1}{\lambda_\omega}\int_\Omega d\cdot\nabla\delta_x \phi_v^\omega dx = -\frac{1}{\lambda_\omega} d\cdot\nabla\phi_v^\omega(x).\end{aligned} \tag{4.5}$$

Putting (4.5) into the series (4.3), we arrive at

$$\mathcal{S}(x;\{\phi_v^\omega\}_{\omega=1}^\infty) = \sum_{\omega=1}^\infty \frac{(d\cdot\nabla\phi_v^\omega(x))^2}{\lambda_\omega^3} = \sum_{\omega=1}^\infty c_\omega(\phi_v^\omega)(d\cdot\nabla\phi_v^\omega(x))^2, \tag{4.6}$$

where the coefficients can be computed, for example, as $c_\omega(\phi_\omega) = \frac{1}{\lambda_\omega^3} = \frac{\|v_\omega\|^3_{L^2(\partial\Omega)}}{\|\phi_v^\omega\|^3_{L^2(\partial\Omega)}}$, and the direction d is an arbitrary fixed unit vector. Since $\{g_\omega\}_{\omega=1}^\infty$ also forms an orthonormal basis of $H^{-1/2}(\partial\Omega)$ and $\Lambda_{\sigma-\sigma_0}$ is a compact self-adjoint operator, the eigenpairs of $\Lambda_{\sigma-\sigma_0}$ can be completely determined from the full data set $\{g_\omega, \Lambda_\sigma g_\omega\}_{\omega=1}^\infty$, namely, there exist coefficients r_k^ω depending on $\{g_\omega, \Lambda_\sigma g_\omega\}_{\omega=1}^\infty$ such that

$$v_\omega = \sum_{k=1}^\infty r_k^\omega g_k \quad \text{and} \quad \phi_v^\omega = \sum_{k=1}^\infty r_k^\omega \phi_k \tag{4.7}$$

which are then put into (4.6) to obtain the expression of $\mathcal{S}(x;\{\phi^\omega\}_{\omega=1}^\infty)$ in terms of $\{\phi^\omega\}_{\omega=1}^\infty$. We emphasize that since $\{g_\omega\}_{\omega=1}^\infty$ are fixed applied data and independent of σ, then $\mathcal{S}(x,\{\phi_\omega\}_{\omega=1}^\infty)$ can be understood as purely determined by the value of $\{\phi^\omega\}_{\omega=1}^\infty$ and x for various σ. ∎

In the derivation above, one of the keys is Green's identity in (4.5) extending the boundary data from boundary to domain interior that relies on the Cauchy difference function (2.6) introduced by the original DSM [16]. Note that we need to let $\gamma = 0$ in order to apply Green's identity which is another reason we choose different γ from [16]. Theorem 4.1 and its proof immediately yield an index function $\mathcal{I}$ defined as

$$\mathcal{I}(x) = 1 \quad \text{if } \mathcal{S}(x;\{\phi^\omega\}_{\omega=1}^\infty) < \infty; \quad \text{otherwise } \mathcal{I}(x) = 0. \tag{4.8}$$

Since $\{g_\omega\}_{\omega=1}^\infty$ are fixed, for neural networks, one only needs to input ϕ^ω, while g_ω becomes some intrinsic structure of $\mathcal{I}$ not needing to be explicitly specified in training just as some other intrinsic properties like the domain geometry. Numerically and practically, one may choose the Fourier modes defined on $\partial\Omega$ as the fixed data functions g_ω [8, 16] which are indeed the eigenfunctions of the radially symmetric σ [53]. This is also our choice for numerical experiments in the next section.

By carefully inspecting the series in (4.6), it is not guaranteed that $\mathcal{S}(x)$ purely depends on $\nabla\phi^\omega(x)$ and x, since the coefficients c_ω also rely on ϕ^ω according to (4.7). This may explain the reason why CNN-DDSM provides a better reconstruction with only one pair of Cauchy data compared with the results of the FNN-DDSM and the DSM in [16]. To our best knowledge, theoretically it still remains unknown whether $\{\nabla\phi^\omega(x)\}_{\omega=1}^\infty$ is sufficient to determine the location of the point x, which is certainly an important and interesting mathematical question. But we think the formula in (4.6) still justifies the expectation that the information of $\{\nabla\phi^\omega(x)\}_{\omega=1}^\infty$ may indicate the location of x to a certain extent.

We believe Theorem 4.1 together with its proof provide some mathematical support to use the data functions from (2.6) as the input of the neural networks. It is rather different from the deep learning approaches in the literature [1, 43, 59] that generally put the data collected at the boundary into the neural network directly. As another perspective to understand this procedure, the index function to be approximated is defined on specially constructed higher-dimensional manifolds instead of just the original boundary data. In many applications, the structure of the data manifolds are unknown. But in the proposed algorithm, the data manifolds can be, in a certain sense, described by the elliptic equations corresponding to the background conductivity. We think the delicate choice of input data manifold really helps the neural networks better approximate the nonlinear structure of the index functions. However, we also note that the approximation theory of neural networks for functions defined on manifolds is so far less known compared with those on Euclidian spaces [5, 50]. Recently, there are some results established in [49, 55, 56, 57] generally based on some specially constructed networks with ReLU activations.

Furthermore, we notice that the index function relies on the series in (4.6) which does not need any knowledge of conductivity values. So it is interesting to question whether a suitable approximation of this index function may still have the same nice feature. Our numerical experiments in the section 5.4 suggest that the index function learned from the boundary data generated by some fixed conductivity value is competent in reconstructing the inclusions with different conductivity values. It is particularly useful in practical situations since the users don't need to have a prior knowledge of the media conductivity values.

5. Numerical expeiments.

In this section, we present numerical experiments to demonstrate that our newly proposed DDSMs are effective and robust for the reconstruction of inhomogeneous inclusions in the EIT problem.

5.1. Problem setting and data generation.

Let the modeling domain be $\Omega = (-1, 1) \times (-1, 1)$ which contains two media with the different conductivity: 10 (inclusion) and 1 (background). The magnitude of conductivity may not be necessarily known (precisely) in our algorithm; see the discussion of the results in Figure 18. Let $\mathcal{U}(a, b)$ be uniform distribution in $[a, b]$. To verify the efficacy of DDSMs for general inclusion distribution, we explore the following three typical scenarios:

- Scenario 1: the inclusions are generated by three random circles with the radius sampled from $\mathcal{U}(0.2, 0.4)$.
- Scenario 2: the inclusions are generated by five random circles with the radius sampled from $\mathcal{U}(0.2, 0.3)$.
- Scenario 3: the inclusions are generated by four random ellipses with the length of the semiminor axis and semimajor axis sampled from $\mathcal{U}(0.1, 0.2)$ and $\mathcal{U}(0.2, 0.4)$, respectively, and the rotation angle sampled from $\mathcal{U}(0, 2\pi)$.

It is reasonable to require that the inclusions do not touch the boundary [22], so we assume the circles/ellipses have at least distance 0.1 to the boundary. More precisely, they are uniformly sampled in the square $(-0.9, 0.9) \times (-0.9, 0.9)$. This sampling strategy has been widely used in solving EIT with deep learning [1, 22, 43, 59]. However, as the major difference from this literature, to make the shape of the inclusions more general and various, we do not require that these circles and ellipses are disjoint from each other, that is, they are free to touch each other. This mechanism will generate much more complicated shapes than basic geometric components, which makes the reconstruction more arduous. Let the circles or ellipses be represented by $c_i(x_1, x_2) = 0$, $i = 1, 2, \ldots, N_c$, where $c_i(x_1, x_2)$ are the related level-set functions and N_c denotes the number of circles/ellipses in each configuration. Then the level-set function of their union can be defined as

$$c(x_1, x_2) = \min_{i=1,..,N_c} \{c_i(x_1, x_2)\}. \tag{5.1}$$

The homogenous background with the conductivity 1 fills the subdomain $\Omega \cap \{(x_1, x_2) : c(x_1, x_2) > 0\}$, and the inhomogeneous inclusions with the conductivity 10 fills the subdomain $\Omega \cap \{(x_1, x_2) : c(x_1, x_2) < 0\}$. For training set, Scenarios 1 and 2 both have 11200 samples, and Scenario 3 has 14400 samples. For testing set, we use 2000 samples for all the three scenarios. For learning rate α, we set $\alpha = 0.0001$ for FNN-DDSM and $\alpha = 0.00025$ for CNN-DDSM for all the cases. For the inserted current data g_ω on the boundary, we follow the idea in [16], but here we use multiple terms,

$$g_\omega(x) = \cos(\omega\theta(x)), \quad x \in \partial\Omega, \quad \omega = 1, 2, \ldots, N, \tag{5.2}$$

where $\theta(x)$ is the polar angle of x, and N is mainly chosen to be $1, 10$, or 20 to get three different training sets for each scenario. Some other choices of small N will also be discussed in the last paragraph in subsection 5.2. We have also tried larger N, for example, $N = 40$, but no significant improvement is observed compared with $N = 20$, and no overfitting phenomena happen either, so we herein focus on relatively smaller N. In order to generate the synthetic data, i.e., $u_\omega|_{\partial\Omega}$, we need to repeatedly solve the modeling equations (1.1)–(1.2) with different distributions of discontinuous conductivity coefficients. Here we employ the immersed finite element method [26, 28], which does not require the mesh to resolve the conductivity discontinuity, to efficiently solve all the equations on the same 200×200 Cartesian mesh. Standard finite element methods can also be used on the mesh generated by (5.1). Then we collect all the values of u_ω at the mesh points on boundary and use them to generate ϕ^ω, $\omega = 1, 2, \ldots, N$, on mesh points by solving (2.6) for each inclusion sample, where for most results (subsections 5.5–5.4) we choose $\gamma = 0$; the nonzero γ will be explored in subsection 5.5. The algorithm can be also applied to the case of only very limited boundary data points available; see the discussion of results in Figure 21. For the prediction stage, since the number of discretization points is not large ($\mathcal{N}_h = 200^2$), no parallelization strategies are applied in the DDSMs in this work.

According to the practical experience [43], only very limited number of real data samples can be obtained in the experimental environments. [43] suggests training the DNNs with simulation data rather than experimental data since the simulation model is not subject to the objective factors. However, as far as we know there are no general rules or empirical suggestions about whether or what types of noise should be added into the training data. On one hand, as mentioned in [47], the noise used in training must be similar to the physiological noise presented in human data varying

across patients and hardware systems and usually unknown in practice. On the other hand, the conventional DSM [16] is developed in such a manner that the noise on the boundary is smoothed out in the duality product (2.2), and thus it can handle relatively large noise in reconstruction. Therefore, due to the insufficient knowledge about the noise and inspired by the robustness feature of the DSM, we here do not include noise in the training set and instead add very large noise in the test set which is similar to the strategy in [43]. This is intentionally to test the robustness of the proposed DNNs with respect to noise if it is not presented in the training set. Then we shall see below that our DNNs can actually handle very large noise in the test set even without any denoising procedure. The noising Cauchy data in the test set are generated by

$$u_\omega^\delta = (1 + \delta G_\omega) u_\omega, \quad \omega = 1, 2, \ldots, N, \tag{5.3}$$

for each inclusion sample, where G_ω are Gaussian random variables with zero mean and unity variation, and $\delta = 0, 10\%, 20\%$ controls the signal-to-noise ratio. These noising data will be used to compute the noising image features ϕ_ω^δ used for both FNN-DDSM and CNN-DDSM.

5.2. Basic numerical results. In this subsection, we present and discuss the training and test results of both the FNN-DDSM and CNN-DDSM for all three scenario mentioned above. We first show the evolution of loss function values versus training iterations in Figures 3 and 4 in which the x-axis is in log-scale to capture the very quick convergence at the beginning. For both DNNs, the training errors of the single Cauchy data pair are larger than those of 10 and 20 pairs, while the errors of 20 pairs are almost comparable with those of 10 pairs. For FNN-DDSM, the error of the single pair stagnates after the first thousand of iterations. The gap of CNN-DDSM between the errors of the single pair and the errors of multiple pairs is clearly much smaller than the one of FNN-DDSM. This also reflects that with a single pair of

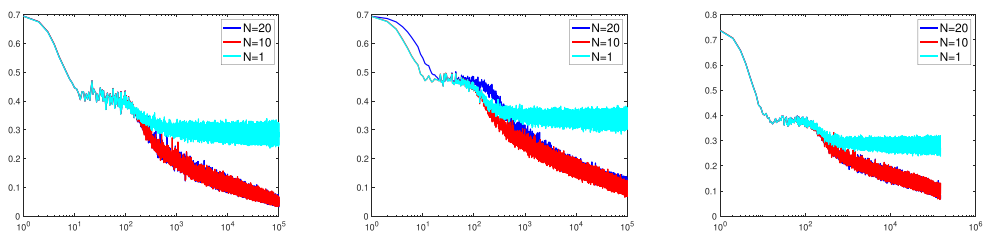

FIG. 3. *Evolution of cross entropy loss values versus training iterations of the FNN-DDSM (from left to right) for Scenario* 1 *(*3 *circles), Scenario* 2 *(*5 *circles) and Scenario* 3 *(*4 *ellipses).*

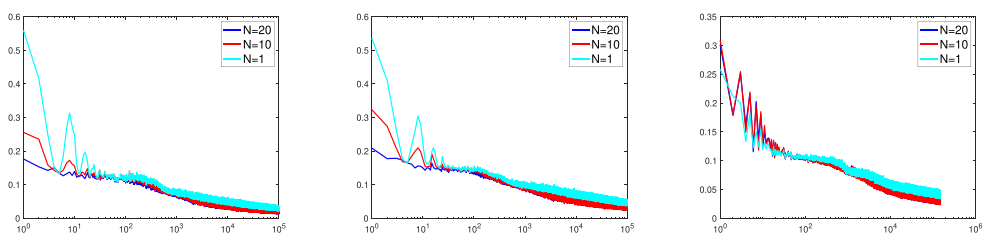

FIG. 4. *Evolution of MSE loss values versus training iterations of the CNN-DDSM (from left to right) for Scenario* 1 *(*3 *circles), Scenario* 2 *(*5 *circles) and Scenario* 3 *(*4 *ellipses).*

Cauchy data, the CNN-DDSM performs better than the FNN-DDSM, which can be noticed in the second column of Figures 7–12. Although a single pair of Cauchy data is sufficient for CNN-DDSM to generate a satisfying reconstruction for some simple shaped inclusions, multiple measurements are needed for more complicated cases and more accurate reconstruction.

Figures 5 and 6 show the reconstruction at certain iterations during the training progress in Scenario 1. It is very clear that both FNN-DDSM and CNN-DDSM will reconstruct the inclusions starting from the boundary to the center, which matches the nature of the EIT problem in that only boundary data are available, and in general the closer to the boundary the easier the reconstruction. From the perspective of learning manner, the inclusions near the boundary can dominate the behavior of the boundary data which are relatively easier to be recognized and learned by DNNs during training, while the center inclusions have very minor effect on the boundary data which are more difficult to be recognized. Similar behavior can be also observed in prediction performance which we will discuss later. Many conventional approaches for EIT such as [38] have shown that inclusions near the center and far away from the boundary

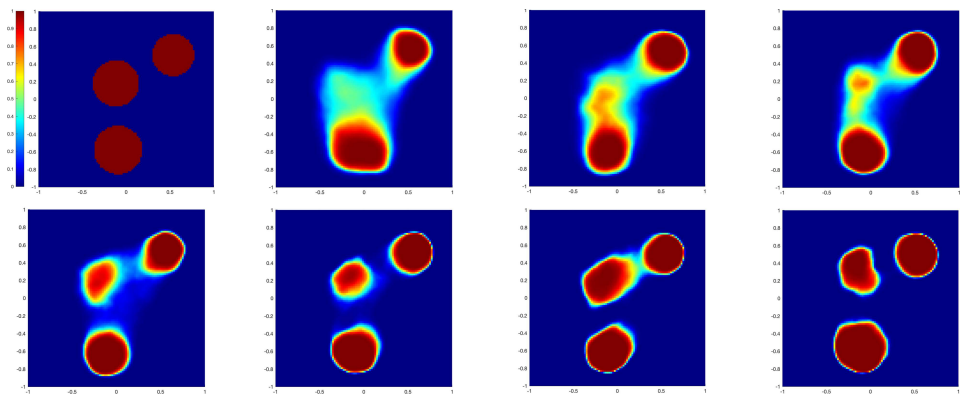

FIG. 5. *Some intermediate results in the training progress of the FNN-DDSM for Scenario* 1 *at iterations:* 2000, 5000, 10000, 25000, 50000, 75000, *and* 100000 *(ordered from left to right and from top to bottom).*

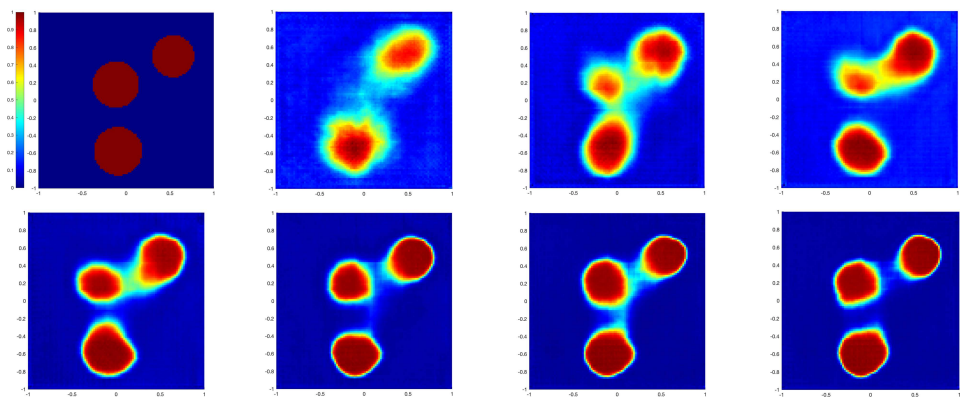

FIG. 6. *Some intermediate results during the training progress of CNN-DDSM for Scenario* 1 *at iterations:* 2000, 5000, 10000, 25000, 50000, 75000, *and* 100000 *(ordered from left to right and from top to bottom).*

are indeed very difficult to detect since the boundary data are very unsensitive to their shape and location. But our DDSMs have some promising results for addressing this issue, which will be discussed in section 5.3. For the time cost of the prediction stage without applying any parallelization techniques, CNN-DDSM takes around 0.2 seconds for one prediction regardless of the number of Cauchy data pairs, while FNN-DDSM takes $0.5 \sim 1$ seconds per prediction depending on the number of Cauchy data pairs. This may be due to the facts that CNN is more sparsely connected and less computation is needed compared with FNN.

Now we present and discuss the reconstruction obtained by DDSMs for all three scenarios. For each one, we choose three inclusion samples in the test set and show the reconstruction in Figures 7–12. We plot the reconstruction generated by $N = 1, 10, 20$ pairs of Cauchy data, and for $N = 20$ we include the noise $\delta = 10\%$ and 20% to test the numerical stability. Based on these figures, we can clearly see that both FNN-DDSM and CNN-DDSM with multiple but limited boundary Cauchy data can yield quite accurate reconstruction for all these inclusions having different geometry and topology. In particular, for some complicated geometry with concavity changes such as Case 3 of Scenario 1(Figures 7–8), Cases 2 and 3 of Scenario 2 (Figures 9–10) and Cases 2 and 3 of Scenario 3 (Figures 11–12), both CNN-DDSM and FNN-DDSM can accurately capture the shape and the position of the inclusions. We highlight that our DDSMs show great potential to handle the case that the inclusions are not star-shaped such as Case 3 of Scenario 1 (Figures 7–8) and Cases 2 and 3 of Scenario 2 (Figures 9–10), which is very hard to achieve in general. Furthermore, we observe that the DDSMs are highly reliable with respect to the noise (up to 20%) with 20 pairs. In some cases, even the 20% noise has no obvious effect on the reconstruction. For other more challenging cases such as Case 2 of Scenario 2 (Figures 9–10), the reconstruction with the 20% noise can be still used as reasonable predictions of the true inclusions. We emphasize that such a large noise can totally destroy the reconstruction for many conventional approaches. So we believe the proposed DDSMs inherit and enhance the robustness feature of the DSM [16], which is a considerable merit for solving the EIT problem that is extremely ill-posed and sensitive to noise.

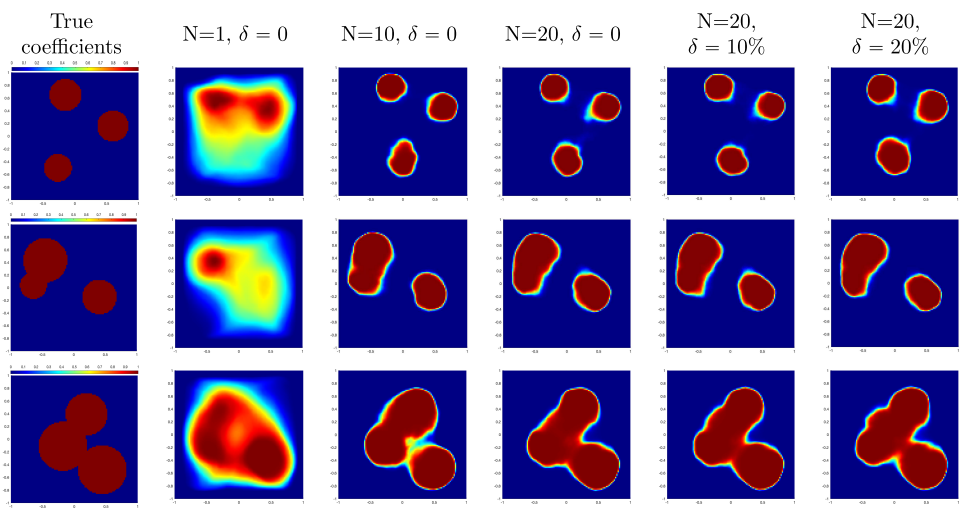

FIG. 7. *FNN-DDSM reconstruction for* 3 *cases in Scenario* 1 *(*3 *circles) with different Cauchy data number and noise level: Case* 1 *(top), Case* 2 *(middle), and Case* 3 *(bottom).*

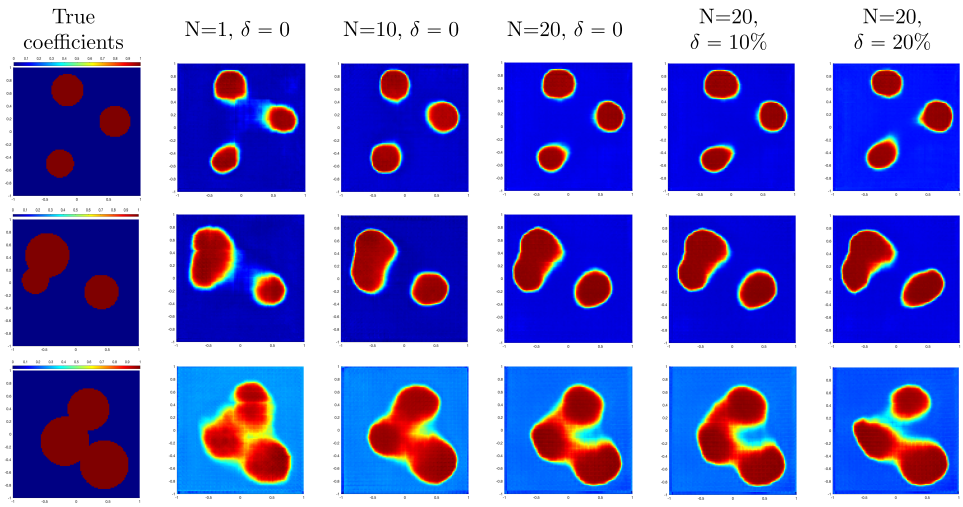

FIG. 8. *CNN-DDSM reconstruction for* 3 *cases in Scenario* 1 *(*3 *circles) with different Cauchy data number and noise level: Case* 1 *(top), Case* 2 *(middle), and Case* 3 *(bottom).*

Moreover, comparing the reconstruction from the FNN-DDSM and CNN-DDSM, we notice that the CNN-DDSM is a bit better than the FNN-DDSM which can be seen from the following two points. First, with the single Cauchy data pair, the reconstructions of the FNN-DDSM are too rough and barely convey any information, while the reconstructions of the CNN-DDSM are able to contain the main features of the true coefficients. In particular, for the relatively simple case that the basic circular and elliptic components are disjoint, the CNN-DDSM can yield much better reconstruction such as Case 1 in each scenario. Second, the comparison between Cases 2 and 3 of Scenario 2 (Figures 9–10) and Case 3 of Scenario 3 (Figures 11–12) shows that the CNN-DDSM can yield slightly better reconstruction at those portions near the domain center away from the boundary. All these gains of the CNN-DDSM are within our expectation since, as mentioned before, to predict the location of a point, the CNN architecture incorporates the information of more neighbor points near this point which may better reflect or approximate the format of the true high-dimensional index function. However, we feel that the FNN-DDSM seems more stable with respect to noise which can be seen from Case 3 of each scenario, since, we think, it only uses the information at the single point to predict its location and thus involves relatively less noise. Another attractive feature of the FNN-DDSM is that its output value has clear probabilistic interpretation. So we can directly read from the plots to conclude which portion is almost certainly inside (red) or outside (blue) the inclusion and which portion can be hardly determined due to lack of information.

To end this subsection, we note that the FNN-DDSM does not perform very well with just $N = 1$. So it would be interesting to study the minimum number N of the Cauchy data pairs that can give reasonable reconstruction. For this purpose, we further train the networks with $N = 2, 3, 4$ and plot the evolution loss values versus training iterations on the left of Figure 13. Indeed the loss values are decreased a lot by increasing N. Moreover, as we can observe from the three reconstruction images on the right of Figure 13, the results certainly become more accurate as N increases, and $N = 4$ almost yields the accurate reconstruction.

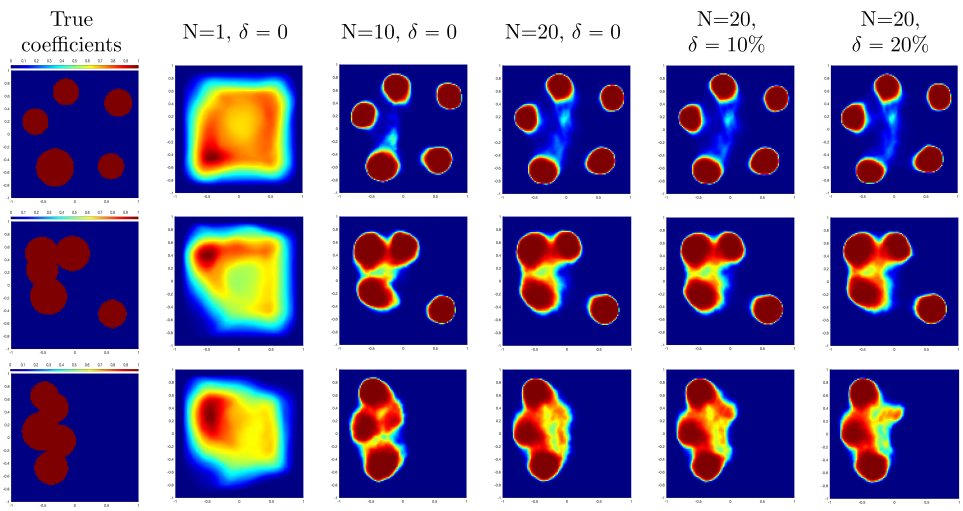

FIG. 9. *FNN-DDSM reconstruction for* 3 *cases in Scenario* 2 *(*5 *circles) with different Cauchy data number and noise level: Case* 1 *(top), Case* 2 *(middle), and Case* 3 *(bottom).*

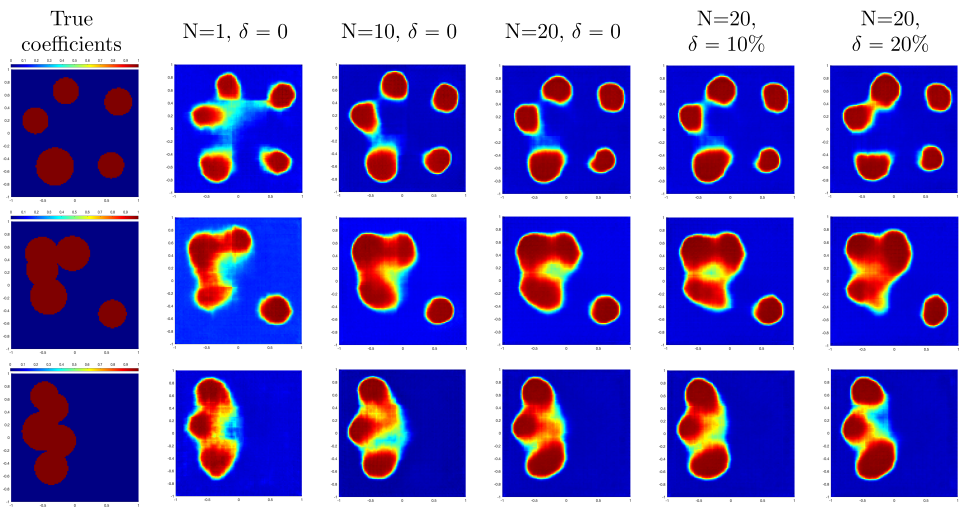

FIG. 10. *CNN-DDSM reconstruction for* 3 *cases in Scenario* 2 *(*5 *circles) with different Cauchy data number and noise level: Case* 1 *(top), Case* 2 *(middle), and Case* 3 *(bottom).*

5.3. Sensitivity to data. The results in the previous subsection have shown the effectiveness and robustness of the proposed DDSMs. In this subsection, we investigate the sensitivity of the DDSMs with respect to the Cauchy data on the boundary. We modify the true coefficients in Case 1 of Scenario 2 and Scenario 3 (first subplot in the first row in Figures 9–12) by artificially moving one of the circles and ellipses to somewhere near the domain center away from the boundary and blocked by the surrounding circles or ellipses to get the subplots in the first three columns of Figures 16 and 17. For the purpose of comparison, we also consider another situation in which this center circle/ellipse is completely removed as shown in the other columns of Figures 16 and 17. We denote these two different coefficient distribution by σ_1^s and σ_2^s and compute the corresponding Dirichlet data $u_{1,\omega}^s|_{\partial\Omega}$ and $u_{2,\omega}^s|_{\partial\Omega}$ according to

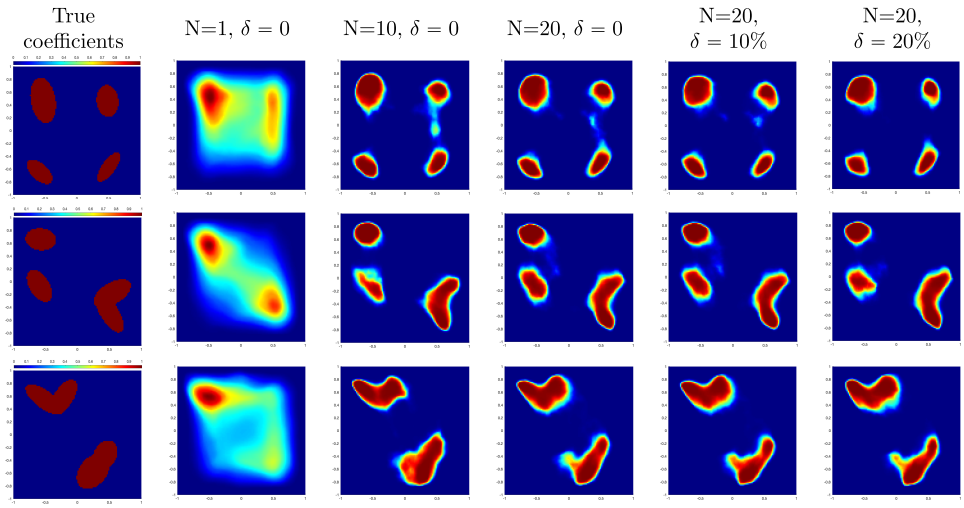

FIG. 11. *FNN-DDSM reconstruction for* 3 *cases in Scenario* 3 *(4 ellipses) with different Cauchy data number and noise level: Case* 1 *(top), Case* 2 *(middle), and Case* 3 *(bottom).*

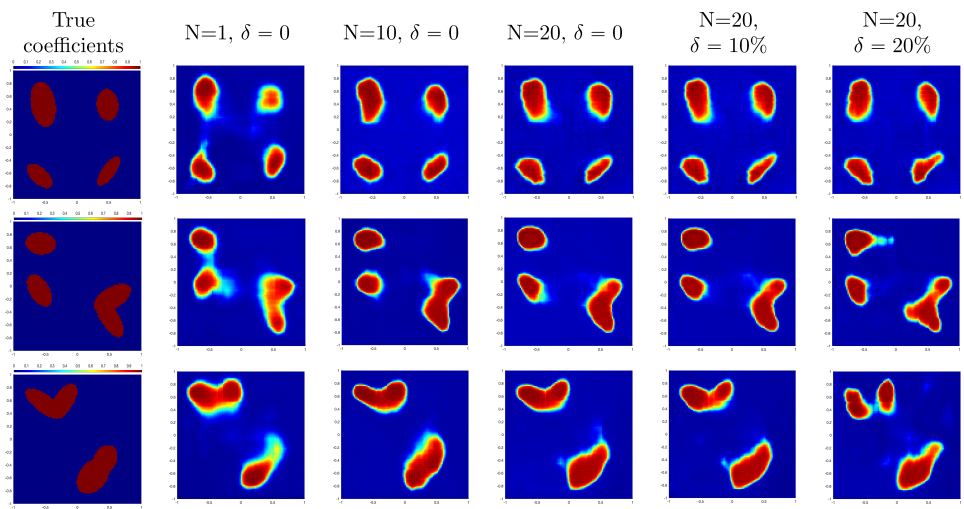

FIG. 12. *CNN-DDSM reconstruction for* 3 *cases in Scenario* 3 *(4 ellipses) with different Cauchy data number and noise level: Case* 1 *(top), Case* 2 *(middle), and Case* 3 *(bottom).*

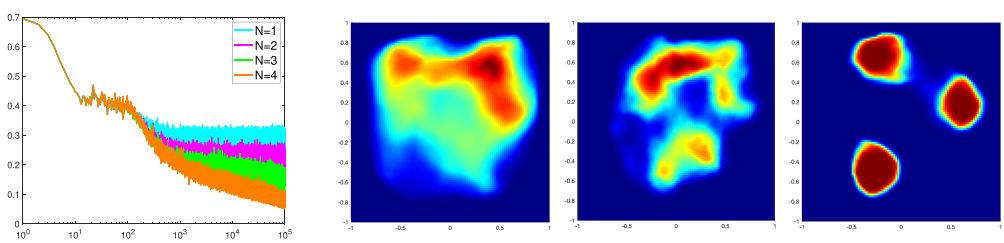

FIG. 13. *Evolution of cross entropy loss values versus training iterations of the FNN-DDSM* $N = 1, 2, 3, 4$ *(left) and the reconstruction results for* $N = 2, 3, 4$ *for Scenario* 1.

the Neumann data in (5.2). Let $s = c$ indicate the case of circles and $s = e$ indicate the case of ellipses. We plot $u^s_{1,\omega}|_{\partial\Omega}$, $u^s_{2,\omega}|_{\partial\Omega}$ and $u^s_{1,\omega}|_{\partial\Omega} - u^s_{2,\omega}|_{\partial\Omega}$ versus the polar angle $\theta \in [0.2\pi]$ in Figures 14 and 15. We observe that with the same Neumann data g_ω, the two Dirichlet data $u^s_{1,\omega}|_{\partial\Omega}$, $u^s_{2,\omega}|_{\partial\Omega}$ are very close with each other, although their coefficient distributions σ^s_1, σ^s_2 are very different. We can see that the relative difference between $u^s_{1,\omega}|_{\partial\Omega}$ and $u^s_{2,\omega}|_{\partial\Omega}$,

(5.4)
$$\frac{\|u^s_{1,\omega} - u^s_{2,\omega}\|_{L^2(\partial\Omega)}}{\|u^s_{1,\omega}\|_{L^2(\partial\Omega)}} \leq 6\% \quad \text{for } \omega = 1, 2, \ldots, 5, \quad \text{and} \quad \frac{\|u^s_{1,\omega} - u^s_{2,\omega}\|_{L^2(\partial\Omega)}}{\|u^s_{1,\omega}\|_{L^2(\partial\Omega)}} \leq 2\% \quad \text{for } \omega \geq 5,$$

is in the same magnitude of regular noise level. It means that the small inclusions near the domain center have very subtle effect on the boundary Cauchy data, which makes them easily hidden in the domain and very difficult to detect. In fact, such differences are small enough to be considered as just noise in the data for many conventional approaches. Fortunately, Figure 16 shows that both the FNN-DDSM and CNN-DDSM are able to sense such small changes in the data and reflect it in a correct manner in the prediction. In Figure 17, the CNN-DDSM still captures the center inclusion quite well, but FNN-DDSM barely gives this information. These phenomena agree with our observation from the previous results that the CNN-DDSM is more sensitive to the center inclusions. Although the reconstruction of the center inclusions is not as accurate as the surrounding inclusions near the boundary, we think it is still satisfactory to a certain extent given that the difference between the Cauchy data of the two different coefficient distributions is very small. Furthermore, we note that this kind of sensitivity to data is also stable with respect to the noise, namely, we can observe the reconstructed center inclusions even with 20% noise for circles and

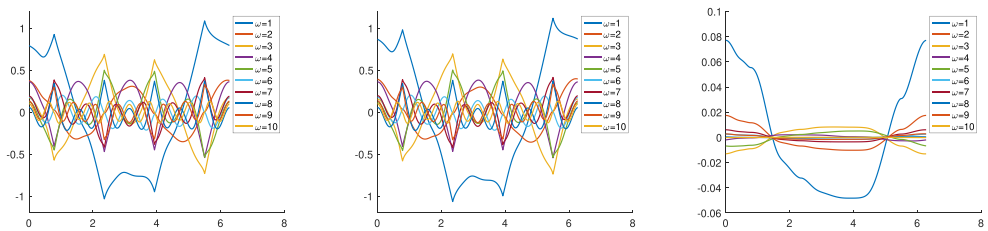

FIG. 14. *Plots of $u^c_{1,\omega}|_{\partial\Omega}$, $u^c_{2,\omega}|_{\partial\Omega}$ and $u^c_{1,\omega}|_{\partial\Omega} - u^c_{2,\omega}|_{\partial\Omega}$ versus the polar angle θ (of points on the boundary).*

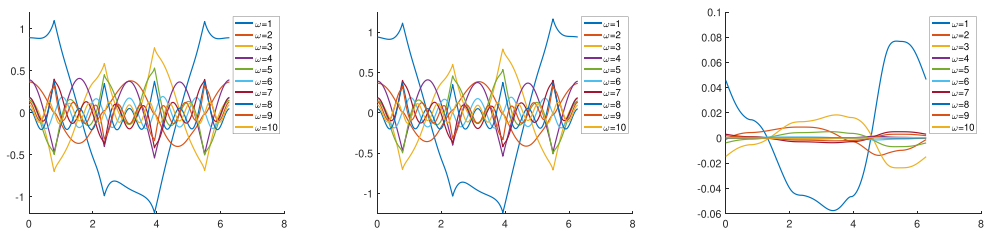

FIG. 15. *Plots of $u^e_{1,\omega}|_{\partial\Omega}$, $u^e_{2,\omega}|_{\partial\Omega}$ and $u^e_{1,\omega}|_{\partial\Omega} - u^e_{2,\omega}|_{\partial\Omega}$ versus the polar angle θ (of points on the boundary).*

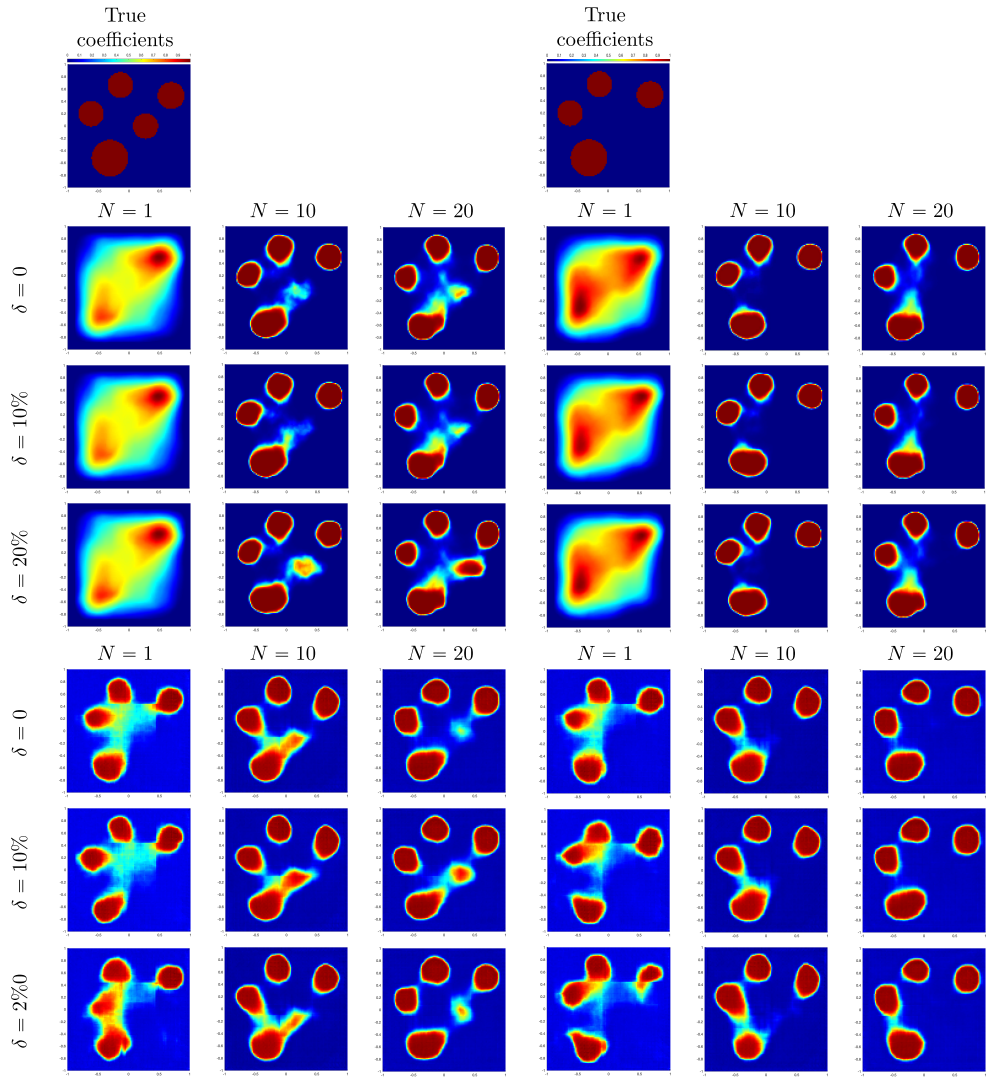

FIG. 16. *FNN-DDSM (the top three rows) and CNN-DDSM (the bottom three rows) reconstruction for random circles: one circle is located closed to the center of domain and blocked from the boundary by other* 4 *circles (left), and this central circle is removed (right).*

10% noise for ellipses. More interestingly, it seems sometimes that the noise actually enhances the reconstruction of the center inclusions instead of undermining it. For example, in Figure 16 for FNN, the center inclusion reconstructed with 10% or 20% noise is certainly more clear than with no noise.

Due to the severe instability of the EIT problem, a small perturbation of the data may yield a completely wrong reconstruction. In our opinion, a good algorithm should be, on one hand, sensitive to the true (correct) data perturbation, i.e., recognize the inclusion information hidden in the Cauchy data as much as possible, and, on the other hand, insensitive to the noise interruption, i.e., the reconstruction is not affected too much by noise. According to the numerical experiments, we believe the proposed DDSMs have this kind of feature.

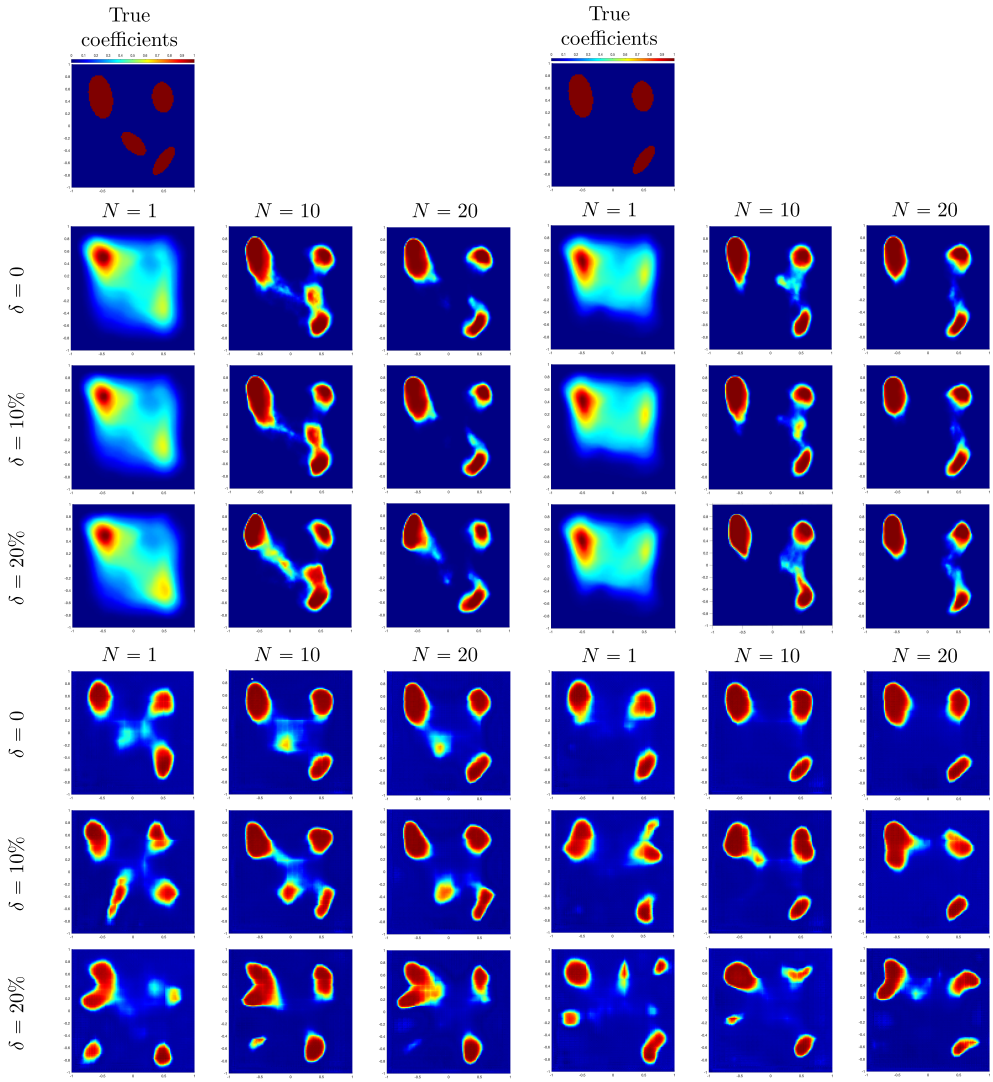

FIG. 17. *FNN-DDSM (the top three rows) and CNN-DDSM (the bottom three rows) reconstruction for random ellipses: one ellipse is located closed to the center of domain and blocked from the boundary by other 3 ellipses (left), and this center ellipse is removed (right).*

5.4. Application to more general cases. In this section, we apply the neural networks trained above to some more general situations. First of all, we consider the case that the values of σ may not be known or may be known inaccurately in advance. As discussed in section 4, the index functions defined from the data manifold $\{x, \phi\}$ to the inclusion distribution should be independent of the values σ. Note that this is also true for the original DSM [16]. We highlight that the index function learned from vast data actually inherits this nice property. To show this, we focus on the second case in the scenario with 5 circles shown in Figures 9 and 10 and apply the networks to the case that $\sigma_0 = 1$ but $\sigma = 2, 5, 20$ in the left inclusion denoted by σ_1^- and $\sigma = 100$ in the right inclusion denoted by σ_1^+. Note that it is different from $\sigma = 10$ inside

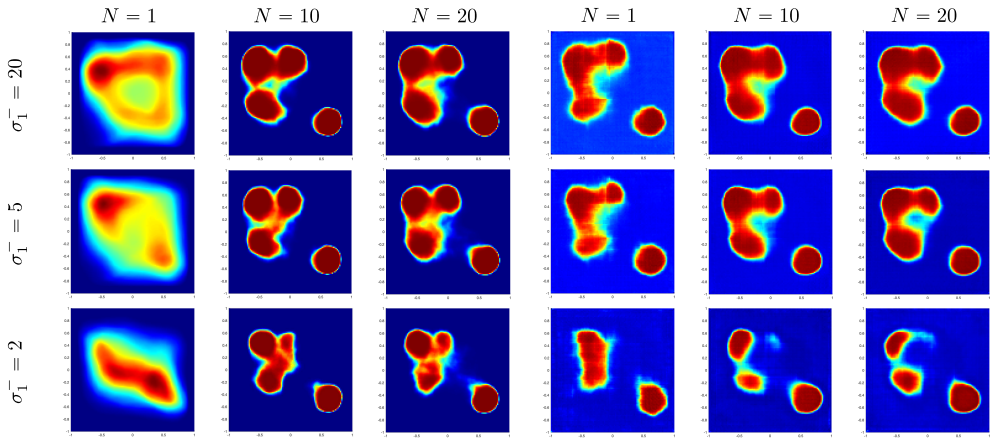

FIG. 18. *FNN-DDSM (the left three columns) and CNN-DDSM (the right three columns) reconstruction for the second case in the first scenario:* $\sigma_0 = 1$, $\sigma = 50$ *in the left inclusion and* $\sigma = 100$ *in the right inclusion.*

all the inclusions set to generate data for training. Here we set $\sigma_1^+ = 2$ very close to the background conductivity $\sigma_0 = 1$ such that the boundary data may be very insensitive to the inclusion shapes, and we intentionally want to test our algorithm for this extreme case. The reconstructed distributions are presented in Figure 18, and the result for the circular inclusion with $\sigma_1^+ = 100$ is always good. As for the inclusion on the right with $\sigma_1^- = 5, 20$, the results are also quite satisfactory and comparable with those in Figures 9 and 10. For the extreme case that $\sigma_1^- = 2$, the algorithm still gives the reasonable reconstruction. We believe the results validate our expectation that the index function is independent of conductivity values to a certain extent. We emphasize it can be a very useful property in practice, since the reconstruction of the inclusion shape does not rely on the correct knowledge of media property. We observe that DDSMs are capable to make a very rough guess about the conductivity value and still obtain quite accurate reconstruction of the inclusion shape. If more accurate reconstruction of the conductivity values is requested, based on the reconstructed inclusion shapes, one may only need to apply an optimization algorithm [18] with fixed shape and several unknowns for the conductivity.

We remark that the inclusion shapes are never known exactly in practice and the a priori knowledge of the shapes may not be always correct. It is important for a DNN-based algorithm to have stable performance on a large variety of inclusion shapes no matter whether they are within the training set-up (library). Therefore, in the second group of experiments, we study the performance of the DDSMs on some typical inclusion shapes which are out of the scope of the training sample library. It is more challenging than predicting the inclusions just in the test set since it requires that the DNNs truly learn and fit the nonlinear mapping from the Cauchy data to the coefficient distribution instead of just a certain projection of the mapping on some low-dimensional data manifold.

For this purpose, we focus on the FNN-DDSM and CNN-DDSM trained by the data of 4 ellipses and show the reconstruction on some typical inclusion shapes: a triangle, two rectangular bars, and a rectangular annulus in Figures 19 and 20, where the shape features are far away from the features of the training data in which each inclusion is a union of ellipses. Some similar ones were also used in [16, 15] for the

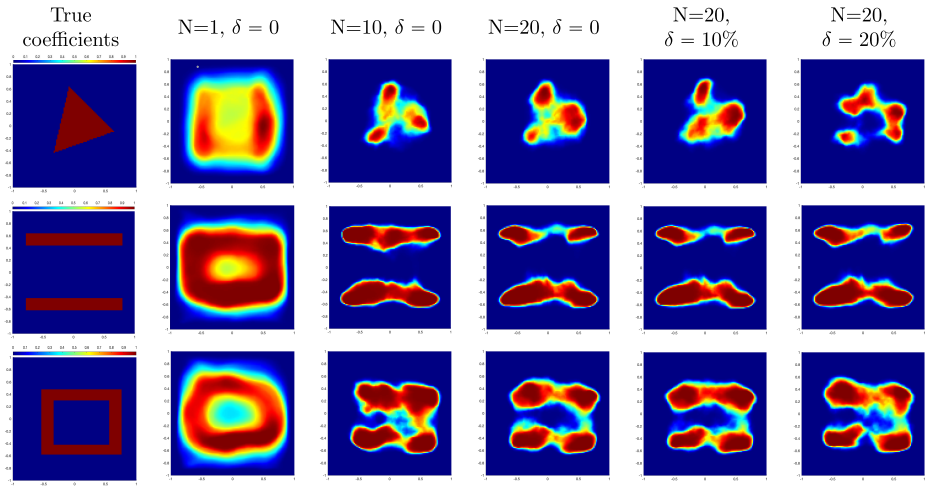

FIG. 19. *FNN-DDSM reconstruction for* 3 *special inclusion shapes: one triangle (top), two long rectangular bars (middle), and a rectangular annulus (top).*

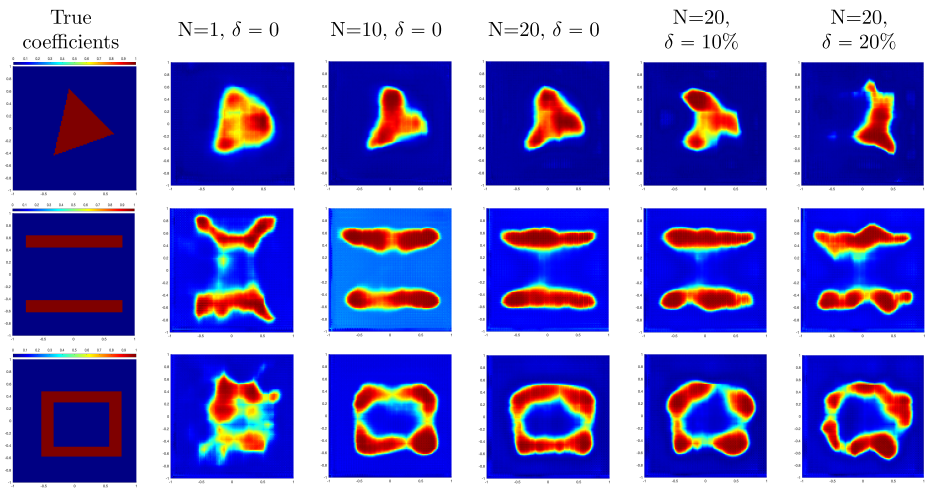

FIG. 20. *CNN-DDSM reconstruction for* 3 *special inclusion shapes: one triangle (top), two long rectangular bars (middle), and a rectangular annulus (top).*

conventional DSM. For the triangle shown in the first row in Figures 19 and 20, we can clearly see that the CNN-DDSM reconstructs the triangular shape, while the FNN-DDSM is only able to capture the three angles even with zero noise and 20 pairs of Cauchy data. It might be due to the facts that FNN-DDSM only includes the information of x and $\{\nabla\phi^\omega\}_{\omega=1}^N$ and this information is not enough to capture the structure of the true index function. The reconstructions by CNN-DDSM with zero noise are all quite satisfactory. Even with 10% noise the reconstruction still provides some rough information about the triangular shape. The case of two rectangular bars is provided in the second row in Figures 19 and 20. For the CNN-DDSM, except the reconstruction by only a single Cauchy data pair, all the other reconstructions recover the shape and position very well. The FNN-DDSN with 10 pairs of Cauchy data and no noise also results in a good performance, but the reconstructions with

other settings are relatively worse. As mentioned in [16], the most challenging case is the rectangular annulus since there is a hole in the inclusion which is hardly detected by the boundary data shown in the last row in Figures 19 and 20. For the CNN-DDSM, the reconstruction becomes much more accurate as the number of Cauchy data pairs increases. Even with 20% noise, we can still obtain an annulus at the right location although its shape deteriorates. Contrary to the CNN-DDSM, the FNN-DDSM captures the feature of the annulus accurately with a single Cauchy data pair but loses two edges in other settings.

To summarize our findings for predicting inclusions different from the training samples, the CNN-DDSM performs quite satisfactorily and much better than the FNN-DDSM, which might be due to the more general structure of its index function. We believe that the behavior of DDSMs can be improved if inclusions with more kinds of shapes such as the triangles and rectangles are added to the training set (users' library) for training. Another interesting observation is that the reconstruction is not symmetric even if the inclusion is symmetric, which is related to the nonlinearity and nonsmoothness of the optimization problem in the neural network.

Finally, we notice that in the previous experiments, the boundary data points we employed are all the mesh points on boundary, i.e., the total 800 data points for a 200×200 mesh. Note that this is not a practical assumption since there may be rather limited electrodes placed on the boundary to emit and receive data. So in this group of experiments, we consider the case that only a few data points are available on the boundary. Here, instead of retraining the networks by the reduced boundary data, we still use the networks trained above by the data available at every mesh point, which is very beneficial in practice. The main reason is that the electrodes may be placed at different locations around the boundary at different situations that yield various data point locations. In such cases, we interpolate the data collected at the electrodes to generate boundary data functions which can be used to further generate the Cauchy difference functions ϕ. According to our extensive numerical experiments, as long as the number of electrodes is sufficient to capture the wave shapes of Fourier modes on the boundary, the obtained reconstruction can be comparable to those obtained by full data at mesh points. To illustrate the behavior, we herein also focus on the second case in the scenario of 5 circles of Figures 9 and 10 and consider the following two situations: 20 data points with $N = 10$ and 8 data points with $N = 1$ which are all equally placed around the boundary, as shown in the first row of Figure 21. As we can observe from the figures, even if the boundary data are limited, the reconstructed inclusion distribution is still satisfactory to a certain extent and almost comparable to those in Figures 9 and 10.

5.5. The choice of γ. In this subsection, we discuss the choice of γ in (2.6) to generate the Cauchy difference functions ϕ. Note that for the original DSM [16], the empirical choice is $\gamma = 2$ which is used together with the norm $|\cdot|_Y = |\cdot|_{H^{3/2}(\partial\Omega)}$ for probing functions in (2.7). For all the numerical experiments presented above, we choose $\gamma = 0$, and based on our experience there are two main reasons for this choice: (i) Since there are only (potentially few) discrete data points on the boundary, computing the graph Laplacian operator $(-\triangle_{\partial\Omega})^\gamma$ will be costly and, more importantly, hard to achieve a good accuracy. (ii) Computing $(-\triangle_{\partial\Omega})^\gamma$, $\gamma > 0$, with only discrete data points will make the data functions ϕ and the reconstructed results more sensitive to noise. Both the accuracy and noise issues will become very severe for the case that only a few data points are available on the boundary, for example, those in Figure 21. However, as discussed in [16], using certain positive γ can indeed significantly

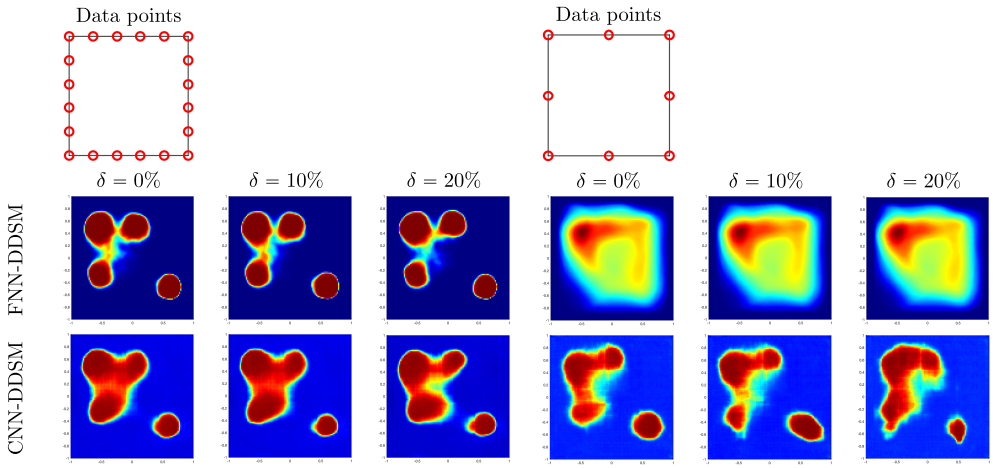

FIG. 21. *FNN-DDSM (the top three rows) and CNN-DDSM (the bottom three rows) reconstruction for random ellipses: one ellipse is located closed to the center of domain and blocked from the boundary by other 3 ellipses (left), and this center ellipse is removed (right).*

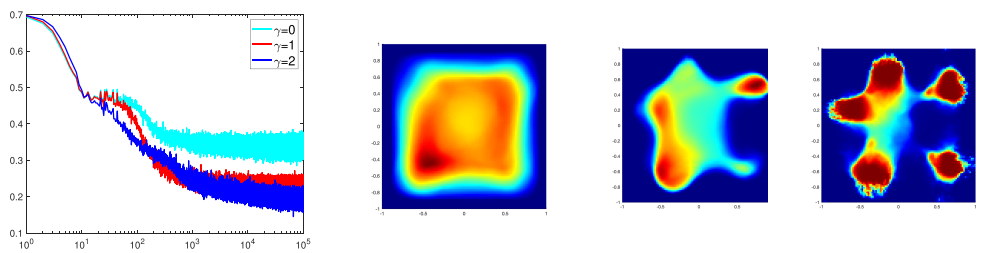

FIG. 22. *Left: evolution of cross entropy loss values versus training iterations of the FNN-DDSM with* $\gamma = 0, 1, 2$ *and* $N = 1$. *Right three: the reconstruction results for Case* 1 *in Scenario* 2 *(no noise).*

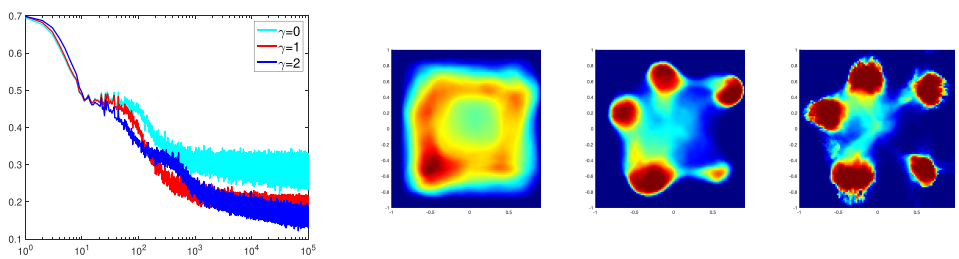

FIG. 23. *Left: evolution of cross entropy loss values versus training iterations of the FNN-DDSM with* $\gamma = 0, 1, 2$ *and* $N = 2$. *Right three: the reconstruction results for Case* 1 *in Scenario* 2 *(no noise).*

enhance the sensitivity of inclusion distribution to the boundary data. To see this, we apply $\gamma = 1, 2$ to generate ϕ which are then used to train the networks.

First of all, notice that the performance of FNN with $\gamma = 0$ and small numbers of N is not very satisfactory as shown by Figure 13. But using a little higher γ can significantly improve the performance as shown by Figures 22 and 23 where the values

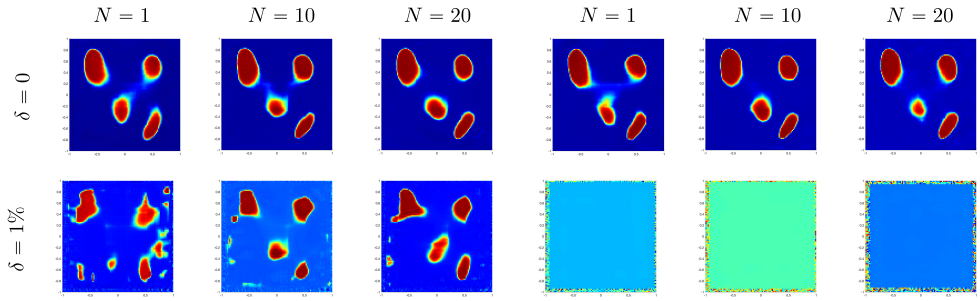

FIG. 24. *CNN-DDSM reconstruction for the case that one ellipse is located closed to the center of domain and blocked from the boundary by other* 3 *ellipses:* $\gamma = 1$ *(the left* 3 *columns) and* $\gamma = 2$ *(the right* 3 *columns).*

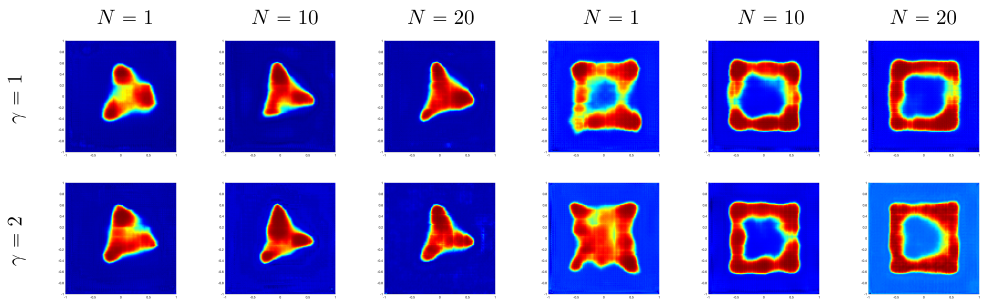

FIG. 25. *CNN-DDSM reconstruction for* 2 *special inclusion shapes: one triangle (the left* 3 *columns) and one annulus (the right* 3 *columns) with* $\gamma = 1$ *(top) and* $\gamma = 2$ *(bottom).*

of loss functions are decreased a lot and much better reconstructions are obtained. We also note that this improvement will deteriorate as N and γ increase. In addition, for $\gamma = 2$ there is a clear checkerboard phenomenon around the inclusion boundary which, we expect, may be due to the inaccuracy of $(-\triangle_{\partial\Omega})^{\gamma}$ for higher γ. Comparing the results in Figure 22 and those in [16] we highlight that FNN-DDSM provides quite comparable reconstructions with the original DSM for the case of a single Cauchy data pair and nonzero γ. So it is interesting to note that, in this case, the index functions derived by mathematical intuition in the original DSM and obtained by the FNN-DDSM might be close to each other.

Second, the center inclusions in the examples of Figures 16 and 17 cannot be reconstructed very accurately since the boundary data are extremely unsensitive to the center inclusions as shown in Figures 14 and 15. Still, using larger γ can enhance this kind of sensitivity and thus improve the reconstruction. To avoid redundancy, we only show the results of CNN for ellipses in Figure 24. We can clearly observe that the reconstruction of the center inclusions is much more accurate. In this example, we also show the results with noise at the same time, and we note that they are indeed much more sensitive to the noise which is the price paid for more accurate reconstruction. In particular, for $\gamma = 2$, even 1% noise can totally destroy the reconstruction.

At last, we show larger γ can also yield better reconstruction for the out-of-scope inclusions such as those in Figures 19 and 20. Again, to avoid redundancy, we only show the results of CNN for the triangle and the rectangular annulus in Figure 25 which are indeed much better than those in Figure 20. Especially for the rectangular

annulus, the reconstruction is quite accurate. However, the tradeoff is still the high sensitivity to the noise for which we omit the numerical results. In addition, we also notice that the reconstruction with $\gamma = 2$ may not be always more accurate than the one with $\gamma = 1$, for example, the rectangular annulus with $N = 1$. We think it may be due to the fact that computing $(-\triangle_{\partial\Omega})^2$ is less accurate than $-\triangle_{\partial\Omega}$ with finite difference techniques.

6. Concluding remarks. In this work, based on the DSM invented in [16] we propose two approaches to construct DNNs for solving EIT problems and the DNN-based DSM, called DDSMs. Our basic idea is to use a large amount of inclusion samples together with boundary measurements to learn the index function in the DSM of which the construction is unknown by classical mathematical derivation for multiple Cauchy data pairs and general-shaped domain. The first DNN we have proposed is an FNN directly approximating the index function pointwise by taking the spatial variable x and the gradient of Cauchy difference functions $\nabla\phi^{\omega}(x)$ as the input, which is called FNN-DDSM. A remarkable feature is that its output has a very clear probabilistic meaning, i.e., the chance of x inside or outside the inclusions, and we hope it can motivate research on the relationship between probability and the DSM. The second one is a CNN approximating a so-called index functional from a stack of Cauchy difference functions to the inclusion distribution, which is called CNN-DDSM. It is worth mentioning that the CNN-DDSM is a further generalization of the conventional DSM [16] since it relaxes the assumption that the location of each point x only lies on the data at this point. Additionally, it has a strong connection to the image segmentation problems. The proposed two DDSMs inherit the features of both classical optimization-type methods (accuracy) and the DSM [16] (efficiency) based on their offline-online decomposition structure, that is, the costly optimization procedure only needs to be done once and in offline phase, and the prediction is done by fast direct evaluation and in online phase. Another difference between the DDSMs and the DSM is that the index function (functional) in DDSMs is implicit and optimal in a certain sense with fixed design of neural network and given data set, while the index function in the DSM is an explicit theoretically developed function.

We have also carried out extensive numerical experiments to show that the proposed DDSMs are effective, accurate, and robust even for very complicated geometry. In particular, the DDSMs are highly stable with respect to large noise which, we believe, is due to the noise smoothing procedure in the generation of the Cauchy difference functions (2.7). Comparing these two methods on the testing set within the scope of geometry set-up of the training set, we conclude that the predictions of these two DDSMs are almost comparable. But the CNN-DDSM is slightly more sensitive to the center inclusion than FNN-DDSM, while the FNN-DDSM seems more stable with respect to noise. When applying them to the inclusions in which the geometry is out of the scope of the training set, the performance of the CNN-DDSM is much better than the FNN-DDSM, which may be due to the fact that the CNN-DDSM uses more neighborhood information to predict the location of each point. The observations from the numerical experiments suggest that our DDSMs have great potential to improve if more types of inclusions are included in the training set and that the index function (functional) in DDSMs is a better approximation of the ideal index function in (2.1), especially the index functional in CNN-DDSM, compared with the DSM.

Acknowledgments. The authors would like to express their sincere gratitude to the referees for their constructive and valuable comments which have led to improvements to the paper.

REFERENCES

[1] A. Adler and R. Guardo, *A neural network image reconstruction technique for electrical impedance tomography*, IEEE Trans. Med. Imaging, 13 (1994), pp. 594–600.

[2] H. Ammari and H. Kang, *Reconstruction of small inhomogeneities from boundary measurements*, Springer, Cham, 2004.

[3] H. Ammari and H. Kang, *Polarization and Moment Tensors: With Applications to Inverse Problems and Effective Medium Theory*, Springer Science & Business Media, New York, 2007.

[4] K. Astala and L. Päivärinta, *Calderón's inverse conductivity problem in the plane*, Ann. of Math., 163 (2006), pp. 265–299.

[5] A. R. Barron, *Universal approximation bounds for superpositions of a sigmoidal function*, IEEE Trans. Inform. Theory, 39 (1993), pp. 930–945, https://doi.org/10.1109/18.256500.

[6] M. Bonnet, *Higher-order topological sensitivity for 2-D potential problems. Application to fast identification of inclusions*, Int J Solids Structures, 46 (2009), pp. 2275–2292.

[7] B. H. Brown, J. A. Tidy, K. Boston, A. D. Blackett, R. H. Smallwood, and F. Sharp, *Relation between tissue structure and imposed electrical current flow in cervical neoplasia*, Lancet, 355 (2000), pp. 892–895.

[8] M. Brühl, *Explicit characterization of inclusions in electrical impedance tomography*, SIAM J. Math. Anal., 32 (2001), pp. 1327–1341, https://doi.org/10.1137/S003614100036656X.

[9] M. Brühl and M. Hanke, *Numerical implementation of two noniterative methods for locating inclusions by impedance tomography*, Inverse Problems, 16 (2000), pp. 1029–1042, https://doi.org/10.1088/0266-5611/16/4/310.

[10] A. Calderón, *On an inverse boundary value problem*, Seminar on Numerical Analysis and Its Applications to Continuum Physics, Sociedade Rio de Brasileira de Mathematica, Janiero, 1980.

[11] D. Cedio-Fengya, S. Moskow, and M. Vogelius, *Identification of conductivity imperfections of small diameter by boundary measurements. Continuous dependence and computational reconstruction*, Inverse Problems, 14 (1998), p. 553.

[12] T. F. Chan and X.-C. Tai, *Level set and total variation regularization for elliptic inverse problems with discontinuous coefficients*, J. Comput. Phys., 193 (2004), pp. 40–66.

[13] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, *DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs*, IEEE Trans. Pattern Anal. Mach. Intell., 40 (2017), pp. 834–848.

[14] Y. T. Chow, F. Han, and J. Zou, *A Direct Sampling Method for Simultaneously Recovering Inhomogeneous Inclusions of Different Nature*, preprint, arXiv:2005.05499 [math. NA], 2020.

[15] Y. T. Chow, K. Ito, K. Liu, and J. Zou, *Direct sampling method for diffusive optical tomography*, SIAM J. Sci. Comput., 37 (2015), pp. A1658–A1684.

[16] Y. T. Chow, K. Ito, and J. Zou, *A direct sampling method for electrical impedance tomography*, Inverse Problems, 30 (2014), 095003.

[17] Y. T. Chow, K. Ito, and J. Zou, *A time-dependent direct sampling method for recovering moving potentials in a heat equation*, SIAM J. Sci. Comput., 40 (2018), pp. A2720–A2748.

[18] E. T. Chung, T. F. Chan, and X.-C. Tai, *Electrical impedance tomography using level set representation and total variational regularization*, J. Comput. Phys., 205 (2005), pp. 357–372.

[19] D. P. Kingma and J. Ba, Adam: *A Method for Stochastic Optimization*, arXiv preprint, arXiv:1412.6980 [cs. LG], 2014.

[20] A. El Badia and T. Ha-Duong, *An inverse source problem in potential analysis*, Inverse Problems, 16 (2000), p. 651.

[21] H. W. Engl, M. Hanke, and A. Neubauer, *Regularization of Inverse Problems*, Springer, Cham, 1996.

[22] Y. Fan and L. Ying, *Solving electrical impedance tomography with deep learning*, J. Comput. Phys., 404 (2020), 109119.

[23] I. Frerichs, G. Hahn, T. Schröder, and G. Heilige, *Electrical impedance tomography in monitoring experimental lung injury*, Intensive Care med., 24 (1998), pp. 829–836.

[24] D. G. Gisser, D. Isaacson, and J. C. Newell, *Electric current computed tomography and eigenvalues*, SIAM J. Appl. Math., 50 (1990), pp. 1623–1634, https://doi.org/10.1137/0150096.

[25] D. Griffiths, *Introduction to Electrodynamics, 3rd ed., Prentice-Hall, Englewood Cliffs, NJ, hall*, 2007.

[26] R. Guo and T. Lin, *A group of immersed finite element spaces for elliptic interface problems*, IMA J. Numer. Anal., 39 (2017), pp. 482–511.

[27] R. Guo, T. Lin, and Y. Lin, *A fixed mesh method with immersed finite elements for solving interface inverse problems*, J. Sci. Comput., 79 (2018), pp. 148–175.
[28] R. Guo, T. Lin, and Q. Zhuang, *Improved error estimation for the partially penalized immersed finite element methods for elliptic interface problems*, Int. J. Numer. Anal. Model., 16 (2018), pp. 575–589.
[29] S. J. Hamilton and A. Hauptmann, *Deep D-bar: Real-time electrical impedance tomography imaging with deep neural networks*, IEEE Trans. Med. Imaging, 37 (2018), pp. 2367–2377.
[30] D. S. Holder, *Electrical Impedance Tomography: Methods, History and Applications*, CRC Press, Boca Raton, FL, 2004.
[31] S. Hong, H. Noh, and B. Han, *Decoupled deep neural network for semi-supervised semantic segmentation*, in Advances in Neural Information Processing Systems, 2015, pp. 1495–1503.
[32] S. Ioffe and C. Szegedy, *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*, arXiv preprint, arXiv:1502.03167 [cs LG], 2015.
[33] V. Isakov and J. Powell, *On the inverse conductivity problem with one measurement*, Inverse Problems, 6 (1990), p. 311.
[34] K. Ito, B. Jin, and J. Zou, *A direct sampling method to an inverse medium scattering problem*, Inverse Problems, 28 (2012), 025003.
[35] K. Ito, B. Jin, and J. Zou, *A direct sampling method for inverse electromagnetic medium scattering*, Inverse Problems, 29 (2013), 095018.
[36] K. Ito, K. Kunisch, and Z. Li, *Level-set function approach to an inverse interface problem*, Inverse Problems, 17 (2001), p. 1225.
[37] B. Jin, T. Khan, and P. Maass, *A reconstruction algorithm for electrical impedance tomography based on sparsity regularization*, Internat. J. Numer. Methods Engrg., 89 (2012), pp. 337–353.
[38] B. Jin, Y. Xu, and J. Zou, *A convergent adaptive finite element method for electrical impedance tomography*, IMA J. Numer. Anal., 37 (2017), pp. 1520–1550.
[39] J. Jordana, M. Gasulla, and R. Pallàs-Areny, *Electrical resistance tomography to detect leaks from buried pipes*, Measurement Sci. Tech., 12 (2001), p. 1061.
[40] A. Kirsch and N. Grinberg, *The Factorization Method for Inverse Problems*, Oxford University Press, Oxford, UK, 2008.
[41] G. Kłosowski and T. Rymarczyk, *Using neural networks and deep learning algorithms in electrical impedance tomography*, Informatyka, Automatyka, Pomiary w Gospodarce i Ochronie Środowiska, 7 (2017) pp. 99–102.
[42] J. Lampinen, A. Vehtari, and K. Leinonen, *Using Bayesian neural network to solve the inverse problem in electrical impedance tomography*, in Proceedings of 11th Scandinavian Conference on Image Analysis SCIA'99, 1999.
[43] X. Li, Y. Zhou, J. Wang, Q. Wang, Y. Lu, X. Duan, Y. Sun, J. Zhang, and Z. Liu, *A novel deep neural network method for electrical impedance tomography*, Trans. Inst. Measurement Control, 41 (2019), pp. 4035–4049, https://doi.org/10.1177/0142331219845037.
[44] T. Lidy and A. Schindler, *Parallel convolutional neural networks for music genre and mood classification*, in Proceedings of MIREX2016, 2016.
[45] S. Martin and C. T. Choi, *Electrical impedance tomography: A reconstruction method based on neural networks and particle swarm optimization*, in 1st Global Conference on Biomedical Engineering & 9th Asian-Pacific Conference on Medical and Biological Engineering, Springer, Cham, 2015, pp. 177–179.
[46] S. Martin and C. T. Choi, *Nonlinear electrical impedance tomography reconstruction using artificial neural networks and particle swarm optimization*, IEEE Trans. Magn., 52 (2015), pp. 1–4.
[47] S. Martin and C. T. M. Choi, *A novel post-processing scheme for two-dimensional electrical impedance tomography based on artificial neural networks*, PLOS One, 12 (2017), pp. 1–27, https://doi.org/10.1371/journal.pone.0188993.
[48] T. Martin and J. Idier, *A FEM-based nonlinear map estimator in electrical impedance tomography*, in Proceedings of the International Conference on Image Processing, IEEE, 1997, pp. 684–687.
[49] H. Mhaskar, *A direct approach for function approximation on data defined manifolds*, Neural Networks, 132 (2020), pp. 253–268.
[50] H. N. Mhaskar, *Neural networks for optimal approximation of smooth and analytic functions*, Neural Comput., 8 (1996), pp. 164–177, https://doi.org/10.1162/neco.1996.8.1.164.
[51] M. Michalikova, R. Abed, M. Prauzek, and J. Koziorek, *Image reconstruction in electrical impedance tomography using neural network*, in Proceedings of the 2014 Cairo International Biomedical Engineering Conference (CIBEC), IEEE, 2014, pp. 39–42.

[52] A. Miller, B. Blott, and T. Hames, *Neural networks for electrical impedance tomography image characterisation*, Clinical Phys. Physiol. Measurement, 13 (1992), p. 119.

[53] M. K. Pidcock, M. Kuzuoglu, and K. Leblebicioglu, *Analytic and semi-analytic solutions in electrical impedance tomography.* i. *Two-dimensional problems*, Physiol. Measurement, 16 (1995), pp. 77–90, https://doi.org/10.1088/0967-3334/16/2/001.

[54] L. Rondi and F. Santosa, *Enhanced electrical impedance tomography via the Mumford–Shah functional*, ESAIM Control Optim. Calc. Var., 6 (2001), pp. 517–538.

[55] L. K. Saul and S. T. Roweis, *Think globally, fit locally: Unsupervised learning of low dimensional manifolds*, J. Mach. Learn. Res., 4 (2003), pp. 119–155.

[56] J. Schmidt-Hieber, *Deep ReLY Network Approximation of Functions on a Manifold*, preprint, arXiv:1908.00695 [stat. ML], 2020.

[57] U. Shaham, A. Cloninger, and R. R. Coifman, *Provable approximation properties for deep neural networks*, Appl. Comput. Harmon. Anal., 44 (2018), pp. 537–557.

[58] J. W. Siegel and J. Xu, *Approximation rates for neural networks with general activation functions*, Neural Networks, 128 (2020), pp. 313–321, https://doi.org/https://doi.org/10.1016/j.neunet.2020.05.019.

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

[60] P. J. Vauhkonen, M. Vauhkonen, T. Savolainen, and J. P. Kaipio, *Three-dimensional electrical impedance tomography based on the complete electrode model*, IEEE. Trans. Biomed. Eng., 46 (1999), pp. 1150–1160.

[61] J. A. Victorino, J. B. Borges, V. N. Okamoto, G. F. J. Matos, M. R. Tucci, M. P. R. Caramez, H. Tanaka, F. S. Sipmann, D. C. B. Santos, C. S. V. Basbas, C. R. R. Carvalho, and M. B. P. Amato, *Imbalances in regional lung ventilation: A validation study on electrical impedance tomography*, Am. J. Respir. Crit. Care Med., 169 (2004), pp. 791–800.

[62] J. Wang, Y. Yang, J. Mao, Z. Huang, C. Huang, and W. Xu, *CNN-RNN: A unified framework for multi-label image classification*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2285–2294.

[63] P. Wang, H.-l. Li, L.-l. Xie, and Y.-c. Sun, *The implementation of FEM and RBF neural network in EIT*, in Proceedings of the 2009 Second International Conference on Intelligent Networks and Intelligent Systems, IEEE, 2009, pp. 66–69.

[64] H. Yao, X. Zhang, X. Zhou, and S. Liu, *Parallel structure deep neural network using CNN and RNN with an attention mechanism for breast cancer histology image classification*, Cancers, 11 (2019), p. 1901.

[65] M. S. Zhdanov and G. V. Keller, *The geoelectrical methods in geophysical exploration*, Methods Geochem. Geophys., 31 (1994), pp. I–IX.

[66] Y. Zou and Z. Guo, *A review of electrical impedance techniques for breast cancer detection*, Med. engrg. phys., 25 (2003), pp. 79–90.