MITNet: GAN Enhanced Magnetic Induction Tomography Based on Complex CNN

Zuohui Chen, Cheng Chen, Chongyang Shao, Chang Cai, Xujie Song, Cheng Chen, Yun Xiang, Ruigang Liu, and Qi Xuan, Senior Member, IEEE

Abstract—Magnetic induction tomography (MIT) is an efficient solution for long-term brain disease monitoring. It focuses on reconstructing the brain’s bio-impedance distribution through nonintrusive electromagnetic fields. However, high-quality reconstruction of brain images remains a significant challenge, as reconstructing images from weak and noisy signals is a highly nonlinear and ill-conditioned problem. In this work, we propose a generative adversarial network (GAN) enhanced MIT technique, named MITNet, based on a complex convolutional neural network (CNN). MITNet takes complex-valued signals as input and outputs a discretized conductivity distribution map. Our approach leverages the power of GANs to eliminate artifacts and enhance the reconstruction of object shapes. The experimental results on the real-world dataset validate the performance of our technique. The F1 score of MITNet surpasses the state-of-the-art stacked auto-encoder (SAE) method by 5.33% on the agar data.

Index Terms—Deep neural network (DNN), electromagnetic inversion, electromagnetic tomography, generative adversarial networks (GANs), magnetic induction tomography (MIT).

I. INTRODUCTION

CEREBROVASCULAR diseases seriously threaten human health. They can cause diseases such as half-body dysfunction, speech disorders, and even death. Thus, early prevention and treatment are crucial. However, existing diagnostic methods such as computed tomography (CT), positron emission tomography (PET), and magnetic resonance imaging (MRI) are often expensive, time-consuming, and may even be hazardous due to radiation exposure. Magnetic induction tomography (MIT), also known as electromagnetic tomography, is a nonintrusive alternative. It uses a harmless magnetic field to induce eddy currents in the target object, and sensing coils detect the magnetic field strength indices. The magnetic field can penetrate the brain skull and convey information about the inner conductivity distribution. Cerebrovascular diseases, such as acute stroke, often induce significant conductivity variations in the lesion, which can be captured by magnetic field sensors. Patients with cerebrovascular diseases are often monitored and treated on a long-term basis. Compared with the existing techniques, MIT is low-cost, harmless, and thus more suitable for continuous long-term monitoring. In this work, we propose a deep learning enhanced MIT technique based on a complex convolutional neural network (CNN) and generative adversarial network (GAN) to improve the accuracy of the imaging results.

The reconstruction of the brain conductivity map, also known as the MIT inverse problem, poses a significant challenge. Linear inversion algorithms, such as back-projection and the Landweber iteration method [1], and regularization methods such as Tikhonov regularization [2] and truncated singular value decomposition (TSVD) [3] have been proposed to...
address this issue. However, since the MIT system is inherently nonlinear and ill-conditioned [4], these methods are generally insufficient to meet expectations [5]. The linearization and approximation of nonlinear problems can cause significant reconstruction errors [6].

Recently, deep-learning-based techniques have gained widespread use in medical imaging [7], including region segmentation [8], [9], cancer classification [10], disease diagnosis [11], and image reconstruction [12], [13]. These techniques possess powerful learning and generalization abilities [14], [15], [16]. However, the existing methods are typically designed to process images or real-valued 1-D data, whereas the MIT system measures excitation signal changes in both amplitude and phase, producing complex-valued data. Therefore, direct application of these methods to reconstruct the impedance distribution map from MIT signals is not feasible.

In this work, we develop an MIT-based system and propose a complex CNN and GAN-based algorithm to predict object location and shape simultaneously. This method is applied directly to complex-valued data, rather than solely on the amplitude of the signal data. We collect a dataset based on actual objects using our MIT system and evaluate our method on a real-world dataset. The experimental results confirm that our algorithm significantly outperforms the state-of-the-art methods.

Specifically, we make the following contributions.

1) We introduce a novel impedance reconstruction algorithm called MITNet, which is enhanced by a GAN and based on a complex CNN. Unlike the existing methods, MITNet can directly process complex-valued data without rounding it off.

2) We also develop an MIT system that uses a mixed excitation and measurement coil structure to acquire induced magnetic field signals and reconstruct impedance distribution.

3) Finally, we evaluate the performance of our algorithm using a real-world dataset. Our experimental results demonstrate that MITNet outperforms the state-of-the-art methods significantly.

The rest of this article is organized as follows. Section II presents the related work. Section III introduces the mechanism and details of our system and dataset. Section IV describes the MITNet. Section V gives the experimental setup and results. Section VI concludes the article.

II. RELATED WORK

In this section, we generalize the related work into three categories, including MIT, deep complex networks, and GANs, respectively.

A. Magnetic Induction Tomography

In recent years, deep learning and neural-network-based algorithms have gained popularity in solving the MIT problem. For instance, Xiang et al. [12] proposed a multifrequency electromagnetic tomography (mfEMT) method based on deep learning for the initial diagnosis of acute stroke. Their method uses a sensor array comprising 12 gradiometer coils with multifrequency sine waves to excite and sense the sensing region.

They use frequency-constrained sparse Bayesian learning to derive the distribution from a sequence of measurements. Although their method performs well in numerical simulations, the image quality is compromised in the phantom experiment. Chen et al. [6] used a stacked autoencoder (SAE) neural network consisting of a multilayer automatic encoder to reconstruct the distribution of electrical characteristics. They calculated the phase difference values through the forward process using the given conductivity distribution and used them to train the neural network. However, their experiment was conducted solely on simulation data. Wang et al. [17] used the variational Born iterative method to reconstruct the arbitrary anisotropic parameters of the object. Their focus was on the forward and inverse process of arbitrary anisotropic objects embedded in multilayered arbitrary anisotropic media, and they validated their method using simulations. Li et al. [18] found that there is a fundamental connection between a deep neural network (DNN) architecture and an iterative method of the MIT problem. Inspired by this connection, they proposed a DNN architecture DeepNIS, which is a complex-valued residual CNN cascaded by multiple layers. They used the MNIST [19] dataset to simulate numbers placed in the sensing region. A full-wave solver to Maxwell equations is used to derive images. Then, they train a network with the derived images as input and ground-truth images as labels. Moreover, their method is still validated only on simulation data.

Many existing studies evaluate their methods using simulated data, which fall short in demonstrating their potential for generalization to real-world datasets. Furthermore, prevalent approaches are constrained to using either amplitude or phase as input, leading to inevitable information loss. The present deep-learning-based models are characterized by simplistic structures with limited fitting capabilities. In contrast, our proposed method capitalizes on the comprehensive utilization of complex-valued signals. It also applies GANs to enhance the reconstruction results.

B. Deep Complex Networks

The sensing coils in our system detect both the signal amplitude and the phase of the excitation signal, resulting in complex-valued measured data with rich representational capacity [20].

While most of the mathematics in deep learning models are currently real-valued, complex-valued data are widely used in audio processing and signal processing fields. In fact, using complex-valued models can achieve comparable or even superior performance compared with real-valued models [21]. For example, Arjovsky et al. [20] propose unitary evolution recurrent neural networks (unitary evolution RNNs) to solve the problem of long-term dependencies. They use orthogonal and unitary matrices in RNNs as building blocks and introduce complex-valued matrices and parameters, providing a richer representation in the complex domain. Danihelka et al. [22] use holographic reduced representations (HRRs) to model complex vectors using real vectors. By combining HRRs with long short-term memory (LSTM) networks, they propose associative LSTM networks and show that using complex-valued
vectors is numerically more stable and efficient than real-valued matrices.

The existing deep-learning-based conductivity reconstruction methods, e.g., DNN [23], FCN [24], and SAE [6], exploit only partial information in complex-valued signal since they usually rely on the traditional real-valued models. In addition, given that the inverse process is inherently nonlinear and vulnerable to noise, the reconstructed outputs frequently manifest unwanted artifacts. To mitigate these challenges, our approach uses the capabilities of complex CNNs and GANs.

C. Generative Adversarial Networks

GANs consist of two networks: a generator network that produces outputs similar to the distribution of training data and a discriminator network that scores the generated output. GAN is an unsupervised learning model, which allows the generator to automatically learn the distribution with unlabeled data, resulting in a generator indistinguishable from real samples.

Goodfellow et al. [25] first proposed GANs, leveraging the adversarial relationship between the generator and the discriminator. However, training of GANs can still be challenging [26]. Li et al. [27] improved the original design with deep convolutional GANs (DCGANs), which use batch normalization [28], transposed convolution, and leaky rectified linear unit (ReLU) [29]. Mirza and Osindero [30] introduced supervised learning into GAN training, proposing conditional GANs (CGANs) to enhance controllability and limit the extent of output.

GANs have exhibited remarkable success across various image and video enhancement applications, including image super-resolution and image enhancement [31]. Jiang et al. [32] introduce an edge-enhanced GAN tailored for remote sensing image super-resolution. Leveraging GAN’s adversarial framework, this approach enhances the spatial details of remote sensing images. They [33] also use GAN as super-resolvers to recognize tiny faces under complex degradation. In video processing [34], Yi et al. [35] propose a progressive fusion GAN for temporal information fusion and consistent video super-resolution.

In medical image analysis, GANs provide a new solution option for challenging problems, such as medical image denoising, reconstruction, and data simulation [36]. Mandija et al. [37] modeled the electrical properties’ reconstruction problem in MRI as a supervised deep learning task, using two CGANs to generate binary masks and MRI images, respectively. The mask distinguishes tissue from air, while the image provides tissue contrast information. Frid-Adar et al. [38] used GANs to generate synthesized medical images to augment data for CNN training. Pradhan et al. [39] presented a GAN-based method to transform 2-D medical images into multiple dimensions. In general, GAN can be used to repair the shape distortion and significantly improve the IoU.

In our work, images reconstructed by complex CNN may suffer from artifacts and distortion of shape. We use CGAN to eliminate artifacts and enhance the reconstruction of object shapes, which proved to be effective in our experiments.

III. PRELIMINARIES

In this section, we introduce the preliminary MIT system in detail.

A. MIT Measurement System

1) Problem Definition: The MIT system comprises a series of excitation coils and sensing coils that encircle the object. Alternating current is used to energize the excitation coils, generating eddy currents and the corresponding magnetic fields. The conductivity and dielectric constant distribution of the object can modify the eddy current intensity [40]. By measuring the induced voltage captured by the sensing coils, we can reconstruct the image of the object.

To solve for the boundary voltage and phase, we typically use the time-harmonic notation of Maxwell’s equations in the eddy current region [41], [42] given the conductivity distribution

\[ \nabla \times H = J_s \]
\[ \nabla \times E = -\frac{d B}{d t} \]
\[ \nabla \cdot B = 0 \]

and nonconducting region

\[ \nabla \times H = J_s \]
\[ \nabla \cdot B = 0 \]

where \( H \) is the magnetic field strength (A m\(^{-1}\)), \( E \) is the electric intensity (V m\(^{-1}\)), \( J_s \) and \( J_e \) are the source current density and eddy current density (A m\(^{-2}\)), respectively, and \( B \) is the magnetic induction (T). The eddy current density and the electrical characteristics of the object satisfy the following equation [43]:

\[ J_e = (\sigma + j \omega \varepsilon_r \varepsilon_0) E \]

where \( \sigma \) is the conductivity, \( \omega \) is the angular frequency, and \( \varepsilon_r \) and \( \varepsilon_0 \) are the relative permittivity and vacuum permittivity, respectively. By introducing the vector magnetic potential \( A \), \( B = \nabla \times A \), we can derive the differential equation to derive boundary parameters

\[ \frac{1}{\mu} \nabla^2 A - j \omega \sigma A = -J \]

where \( \mu \) is the magnetic permeability of the object. A simplified statement of the problem can be expressed using the following equation [12]:

\[ \varphi = F(\sigma) + v \]

where \( F \) is the mapping of conductivity to the coil voltages, \( \sigma \in \mathbb{R}^M \) is the conductivity distribution, \( \varphi \in \mathbb{R}^N \) is the measurements with noise \( (N \ll M) \), and \( v \) is the noise vector. Equation (5) defines the mapping relationship between the magnetic fields and the boundary voltages [44], which is also known as the forward problem of MIT.
The goal of MIT is to reconstruct a map of the conductivity distribution using boundary voltage measurements. However, this is a challenging task since it is both nonlinear and ill-posed [45]. A small variation in the measurement can cause a dramatic change in the solution to the inverse problem. The problem can be formulated using the following equation:

$$\sigma = Q(\varphi - v).$$  \hspace{1cm} (6)

The existing solutions include TSVD [3], Tikhonov regularization [46], iterative methods [5], and other direct reconstruction methods. These methods use assumptions and approximations to solve (6), which can result in inaccurate and poor reconstruction results. To address this issue, we propose using a complex CNN that can automatically learn from the data.

2) System Composition:
The data acquisition system is completed in cooperation with Hangzhou Utron Technologies Company Ltd. [47]. The system includes four parts: a multiplexed coil array, a signal generation and control module, a signal detection unit, and an automatic gain module. The object field is a circle with a diameter of 20 cm with 16 coils distributed at equal angles on the periphery. The coil array consists of 16 multiplexed coils, which have the same configuration, all with 13 turns, and the coil diameter is 18 mm. The signal acquisition process is shown in Fig. 1. During measurement, each coil excites the magnetic field in turn, while the rest receive the magnetic field interfered with by the object. The automatic gain module amplifies detected signals to achieve high-precision phase discrimination, and then the AD module converts them into digital data.

The sensitivity and stability are key metrics for evaluating the effectiveness of an MIT system [47]. To assess the system’s performance, we consider two important factors: the signal-to-noise ratio (SNR) and phase drift. SNR is measured when no object is present in the coil array. Since our system consists of 16 multiplexed coils, we measure the SNR for every possible combination using the following equation:

$$\text{SNR} = 10 \log_{10} \frac{m^2}{v}$$  \hspace{1cm} (7)

where $m$ is the mean and $v$ is the variance of the measured data. There are a total of 240 combinations of emitter pairs, resulting in an average SNR of 62 dB, with a maximum of 71 dB and a minimum of 51 dB. It should be noted that the secondary signal is typically in-quadrature to the primary field and much smaller [48]. Therefore, the measurement noise in our system should be kept minimal to recover the conductivity distribution. Fig. 2 illustrates the mean phase variation for 1 h after the startup. The measured data phase becomes stable after approximately 40 min with an average variation of less than 0.05°.

IV. MITNET: MIT PROCESSING NETWORK
In this section, we introduce the structure of our system and the MITNet.

A. MITNet Overview
The overall architecture of the image reconstruction system is illustrated in Fig. 3. The data acquisition equipment collects the excitation signals and digitizes them before they are input to the MITNet deep learning structure. MITNet is composed of a complex CNN and a GAN. The objective is to derive a high-dimensional impedance distribution from low-dimensional signals, wherein the input comprises a 16 × 16 complex matrix representing signals captured by 16 sensing coils. Due to the inherently non-linear and ill-conditioned characteristics of the MIT inverse process, directly reconstructing the fine-grained distribution proves challenging. To address this challenge, we use a strategy involving the division of the measurement area into coarse-grained grids, facilitated by the widely adopted finite element method (FEM). Specifically, the area is discretized into 512 triangles [49], and the model’s output takes the form of a real-valued vector. Each entry in this vector corresponds to the conductivity values of the respective triangles.

The complex CNN takes a 16 × 16 complex matrix (differential frame) as input and is responsible for reconstructing the locations and shapes of objects. To achieve this, the network uses both downsampling and upsampling structures. On the downsampling side, several max-pooling layers are added between the complex convolutional layers. The input
data are a $16 \times 32$ real-valued vector, which is then converted into a $16 \times 16 \times 2$ complex-valued input for the network. On the upsampling side, the outputs from different layers are first converted into real numbers (C2R), concatenated, and then converted back to complex values (R2C) for convolution, to generalize and process features from different layers.

The GAN is used to enhance the reconstruction results. After passing through the complex CNN, the reconstructed images are further processed by the GAN to improve their quality.

B. Complex CNN

A complex-valued black CNN has unique building components [21], including complex number representation, complex convolution, and complex-valued activations. However, current deep learning frameworks can only compute real numbers. Thus, we need to represent complex-valued tensors and implement complex number operations by defining new arithmetic rules on real numbers.

1) Complex Number Representation: To represent a complex number $c = a + bi$, where $a$ is the real part and $b$ is the imaginary part, we need to use two real-valued channels. For instance, if the input has $N_{in}$ complex numbers, the kernel size is $m \times m$, and the output has $N_{out}$ feature maps, then the complex tensor $T$ has a size defined as

$$\text{Size}_T = N_{in} \times N_{out} \times m \times m \times 2. \quad (8)$$

2) Complex Convolution: To process complex tensors, we need to define complex convolution. A complex convolutional filter is determined by its weight matrix, which is expressed in the same way as a complex tensor $T$, i.e., using two real-valued channels to receive the real part and imaginary part. We define a complex weight matrix as follows:

$$W = A + iD \quad (9)$$

where $A$ and $D$ are the real number matrices. Assuming that the input is a complex vector $h = x + iy$, the convolution operator, to make complex convolution equivalent to a real-valued 2-D convolution, we have

$$W * h = (A * x - D * y) + i(D * x + A * y). \quad (10)$$

In (10), according to the complex number arithmetic rules, we calculate the tensor real part and imaginary part separately on two channels. The size of output tensor follows (8).

3) Complex-Valued Activations: To process the complex-valued representations, several activation functions are proposed. In our experiment, we use modReLU [20]

$$\text{modReLU}(z) = \text{ReLU}(|z| + b)e^{i\theta_z}$$

$$= \begin{cases} |z| + b & \text{if } |z| + b \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

where $z \in \mathbb{C}$, $\theta_z$ is the phase of $z$, and $b \in \mathbb{R}$ is a learned parameter. ReLU prohibits neurons with the output of less than 0. In the complex domain, modReLU uses the modulus instead. If $z$ is within the circle with radius $b$ around the origin 0, it is set to 0. Thus, the phase $\theta_z$, which is an important feature, can be represented.

To design our complex classification network, we use the existing complex CNN components [21] and the U-net [50] network structure. The input is first converted into complex-valued features and then processed through layers of complex convolution and pooling. As the output resolution is higher than the input, we use upsampling. We concatenate low-level features and high-level features of the same scale using
skip-connections, and then upsample the extracted features. The final layer converts complex-valued numbers into real-valued ones. Finally, we apply a densely connected layer to obtain the real-valued $1 \times 512$ vector that matches the triangulation field. We use binary cross-entropy (BCE) loss to train the model, which is shown as follows:

$$\text{loss}_{\text{CNN}}(o, t) = \frac{1}{n} \sum_{i} w_i (t_i \log(o_i) + (1-t_i)\log(1-o_i))$$

(12)

where $o$ is the output vector, $t$ is the ground-truth vector from real-world measurements, and $w$ is the weight labels. The conductivities of the object and the surrounding area are assumed uniform in our setup.

C. GAN-Based Network

Since the conductivity distribution field is discretized with a resolution of $1 \times 512$, the output of complex CNN is coarse-grained. To obtain the accurate location and shape of the object, we use CGAN [30] to enhance the output. Our model is based on the pixel2pixel design [51]. A traditional GAN consists of a generator and a discriminator, which are described by the following objective function:

$$L_{\text{GAN}} = E_{d \sim \mathcal{D}}[\log D(x_{\text{in}})] + E_{z \sim \mathcal{N}}[\log(1 - D(G(z)))]$$

(13)

where $D$ maximizes $\log D(x_{\text{in}})$, while $G$ minimizes $\log(1 - D(G(z)))$. Here, $x_{\text{in}}$ is the label image, $z_{\text{in}}$ is random noise, and $G(z)$ is the candidate image with the probability distribution of $x_{\text{in}}$. The discriminator outputs $D(x_{\text{in}})$ or $D(G(z))$, which is a scalar scoring how close the input is, defined as the probability of the input belonging to $x_{\text{in}}$.

In our case, we use a CGAN to reconstruct the conductivity image. It adds extra information $y_{\text{e}}$ to penalize generator outputs that are beyond the given information. The extra information $y_{\text{e}}$ can be labels or any other limitations. Here, $y_{\text{e}}$ is set as the complex CNN results, forcing the GAN output to be more accurate. In practice, $y_{\text{e}}$ is fed into both the discriminator and generator as an additional input layer. The objective function can then be rewritten as

$$L_{\text{CGAN}} = E_{y_{\text{e}},x_{\text{in}}}[\log D(x_{\text{in}}|y_{\text{e}})] + E_{y_{\text{e}},z_{\text{in}}}[\log(1 - D(G(z_{\text{in}}|y_{\text{e}})))]$$

(14)

Moreover, the L2 distance of the output is also penalized using the ground-truth data while the discriminator remains unchanged [52]. After adding the penalty, our objective function becomes

$$L^* = \arg \min_{\theta_g} \max_{\theta_d} L_{\text{CGAN}} + \lambda L_{\ell1}$$

(15)

where $\theta_g$ and $\theta_d$ are the parameters of $G$ and $D$, respectively. A normal generator transforms a white Gaussian noise or random dropout [51] to the target distribution. Noise $n$ induces diversity in the output of GAN. However, in the MIT inverse problem, we expect the output of GAN reflects the actual distribution. Thus, our generator takes the images as input without random noise or random dropout. In this work, we use the encoder-decoder structure [53].

We use a total of ten convolutional layers for feature extraction, with batch normalization and ReLU activation applied after each convolution layer. Max-pooling is applied after every two convolution layers, and the upsampling process includes four transpose convolution layers that concatenate with symmetrical low-level features in the structure. The final output resolution is $256 \times 256$. For the discriminator, we use the PatchGAN design [51]. In a regular GAN, the discriminator outputs a scalar representing the corresponding confidence. However, PatchGAN outputs a $N \times N$ matrix where each element corresponds to a patch of the input image, indicating whether that patch is real or fake. This is achieved by sending concatenated image pairs into a series of convolution layers. The patches represent different perspective fields of the discriminator that are sensitive to image details at different textures. In the MIT problem, this loss penalizes the generator output, making it more distinguishable between different conductivity objects.

Algorithm 1 Training Complex CNN

Input: differential measurement frames $d = \{d^1, d^2, \ldots, d^n\}$, ground-truth triangulation map vectors $y = \{y^1, y^2, \ldots, y^n\}$. Output: triangulation maps $\{\tilde{x}^1|y^1, \tilde{x}^2|y^2, \ldots\}$.

1: repeat
2: Sample $m$ examples $\{d^1|y^1, d^2|y^2, \ldots, d^m|m\}$ from $d$ and $y$, correspondingly.
3: Input examples into the complex CNN model and calculate the forward propagation result $\tilde{y}^1, \tilde{y}^2, \ldots$.
4: Update complex CNN model parameters $\theta_{\text{CNN}}$ to minimize:
5:   $\text{loss}_{\text{CNN}} = \frac{1}{N} \sum_i (w_i \log(\tilde{y}_i) + (1-y_i)\log(1-\tilde{y}_i))$.
6: until convergence.
7: Observe triangulation map vectors $\{\tilde{d}^1|y^1, \tilde{d}^2|y^2, \ldots\}$,
8: Fill triangulation maps $\{\tilde{x}^1|y^1, \tilde{x}^2|y^2, \ldots\}$ according to triangulation map vectors $\{\tilde{d}^1|y^1, \tilde{d}^2|y^2, \ldots\}$.
9: Return $\{\tilde{x}^1|y^1, \tilde{x}^2|y^2, \ldots\}$.

D. Algorithms

The training processes for the complex CNN and GAN are outlined in Algorithms 1 and 2, respectively. Our algorithm includes several modifications compared with the original algorithms. First, for training the complex CNN, we replace the original loss function with the BCE loss as shown in (12). Since the output vector is binarized to indicate the presence or absence of the object, the reconstruction process is essentially a binary classification problem. Therefore, we use the BCE loss function, which does not require a softmax layer to normalize the forward propagation. In addition, to generate a stable output image, we initialize the GAN with preprocessed images. Prior to training the complex CNN, we preprocess the data to obtain the input differential frame and ground-truth map vector.

V. EXPERIMENTAL RESULTS

In this section, we evaluate our algorithms based on the collected real-world dataset.
Algorithm 2 Training GAN

Input: triangulation maps \( x = \{x^1, x^2, \ldots, x^n\} \), ground truth images \( t = \{t^1, t^2, \ldots, t^n\} \).
Output: reconstructed images \( \{\tilde{x}^1|t^1, \tilde{x}^2|t^2, \ldots\} \)

1: repeat
2: Sample \( m \) examples \( \{x^1|t^1, x^2|t^2, \ldots, x^m|t^m\} \) from \( x \) and \( t \), correspondingly.
3: Obtain \( \{\tilde{x}^1|t^1, \tilde{x}^2|t^2, \ldots, \tilde{x}^m|t^m\} \) by adding labels to the generated images.
4: Obtain reconstruct images \( \{\tilde{x}^1|t^1, \tilde{x}^2|t^2, \ldots, \tilde{x}^m|t^m\} \) by adding labels to the generated images.
5: Update discriminator parameters \( \theta_d \) to maximize:
\[
\begin{align*}
\psi = \frac{1}{m} \sum_{i=1}^{m} \log D(x^i|t^i) + \frac{1}{m} \sum_{i=1}^{m} \log (1 - D(\tilde{x}^i|t^i)). \\
\theta_d &\leftarrow \theta_d + \eta \nabla \psi(\theta_d).
\end{align*}
\]
6: Obtain \( \{\tilde{x}^1|t^1, \tilde{x}^2|t^2, \ldots, \tilde{x}^m|t^m\} \) by adding labels to the generated images.
7: Update generator parameters \( \theta_g \) to minimize:
\[
\begin{align*}
\psi &= \frac{1}{m} \sum_{i=1}^{m} \log(1 - D(G(z|t^i)|y^i)) + \lambda \sum_{i=1}^{m} \sum_{t^i} (\tilde{x}^i - t^i)^2. \\
\theta_g &\leftarrow \theta_g - \eta \nabla \psi(\theta_g).
\end{align*}
\]
8: until convergence.
9: Obtain generated images \( \{\tilde{x}^1|t^1, \tilde{x}^2|t^2, \ldots\} \) by putting labels to generator: \( \tilde{x}^i|t^i = G(z|t^i)|t^i. \)
10: Return \( \{\tilde{x}^1|t^1, \tilde{x}^2|t^2, \ldots\} \).

A. Experimental Setup

The evaluation of the cylinder and cube objects is performed separately, and we divide the training and testing sets with a ratio of 5:1. We collect three frames at each location and combine them into one data entry. Our model is trained on a server equipped with a Tesla V100 graphics processing unit (GPU), an Intel Xeon CPU running at 2.30 GHz, and PyTorch as the deep learning framework. The Adam optimizer is used for all the networks, with a learning rate of 0.001. The training process is performed for 200 epochs, and the batch size is set to 16. For the complex CNN and GAN training, we adopt the BCE loss and mean square error loss, respectively. Furthermore, after discretization, we apply a smoothing operation to the data by averaging the values of each triangular region with its adjacent ones, as given by the following equation:

\[
\alpha_i = \frac{1}{m} \sum_{e=1}^{m} \alpha_i^e
\]

where \( m \) is the number of surrounding regions. This smoothing process is applied to all the comparing algorithms.

1) Evaluation Metrics: Intersection over union (IoU) is a commonly used metric for evaluating reconstructed images. It is defined as follows:

\[
\text{IoU} = \frac{\text{area}(R_{\text{pred}}) \cap \text{area}(R_{\text{true}})}{\text{area}(R_{\text{pred}}) \cup \text{area}(R_{\text{true}})} \times 100\%
\]

where \( \text{area}(\cdot) \) measures the number of pixels, \( R_{\text{pred}} \) represents the predicted object in the reconstructed image, and \( R_{\text{true}} \) corresponds to the object in the ground-truth map. A higher IoU value indicates a better reconstruction of the image. When the IoU equals 1, it means the reconstructed image is completely consistent with the ground truth.

Another metric to consider is the centroid distance (CD), which measures the accuracy of object localization. The centroid represents the center of mass of a geometric object. The CD is the Euclidean distance between the centroid of the ground truth and the centroid of the reconstructed object. It can be calculated using the following formula:

\[
\text{CD} = \sqrt{(d^x_x - d^x_y)^2 + (d^y_y - d^y_y)^2}
\]

where \( d^x_x \) and \( d^y_y \) is the centroid of ground truth and \( d^x_y \) is the centroid of reconstructed image. In (18), \( d^x_x \) and \( d^y_y \) are calculated using

\[
d^x_x = \frac{1}{k} \sum_{i=1}^{k} d_x^i \quad \text{and} \quad d^y_y = \frac{1}{k} \sum_{i=1}^{k} d_y^i.
\]

The CD between the ground truth and reconstructed image should be minimized, and a value of zero indicates that the two centroids overlap with each other.

We also use several metrics commonly used in computer vision areas, including precision (P), recall (R), and F1 score (F1). Precision measures the percentage of true positive predictions among all the positive predictions, while recall indicates the proportion of actual positive instances that are correctly identified by the model out of all the positive instances. F1-score combines precision and recall to provide a comprehensive performance measure. The definitions of these metrics are as follows:

\[
\begin{align*}
\text{P} &= \frac{\text{area}(R_{\text{TP}})}{\text{area}(R_{\text{TP}}) \cup \text{area}(R_{\text{FP}})} \times 100\% \\
\text{R} &= \frac{\text{area}(R_{\text{TP}})}{\text{area}(R_{\text{TP}}) \cup \text{area}(R_{\text{FN}})} \times 100\% \\
\text{F1} &= \frac{2 \times \text{P} \times \text{R}}{\text{P} + \text{R}} \times 100\%
\end{align*}
\]

where \( R_{\text{TP}} \) are the objects correctly classified, \( R_{\text{FP}} \) are the objects falsely classified, \( R_{\text{FN}} \) is the correctly classified nonobject regions, and \( R_{\text{FN}} \) is the wrongly classified nonobject regions.

2) Baseline Methods: We compare our MITNet with the Newton–Raphson (NR) method [54], VDDNet [55], optimized fully connected network (FCN) [24], SAE [56], pCNN, mCNN, and simple machine learning techniques.

FCN: For FCN [24], as it is real-valued, we use the magnitude of complex-valued data as input. The output is the same as that of the complex CNN and NR algorithms. The input is a \( 1 \times 256 \) vector, and the output is a \( 1 \times 512 \). The entire network consists of two hidden layers of size \( 1 \times 360 \), with batch normalization applied between each layer. We fill the triangulation map with the corresponding output value.

VDDNet: VDDNet [55] consists of a preimaging module and a CNN module, where the preimaging module initially reconstructs the conductivity via the conjugate gradient method and CNN module is used to filter artifacts and generate fine-grained conductivity distribution. In our experiments, we modified the input and output size of VDDNet according to our datasetting, where the input is the \( 1 \times 256 \) phase and the output is a \( 1 \times 512 \) vector.
SAE: The SAE [56] uses a stacked structure with two layers of autoencoders. We set the output layers of the two autoencoders to 512, the input layers to 256 and 128, and the hidden layers to 128 and 64, respectively. The two autoencoders are first trained using $1 \times 512$ ground-truth vectors, and the hidden layer of the previous autoencoder is used as the input of the next one. The final SAE consists of two layers of autoencoders together with a decoder layer of output size 512. The pretrained autoencoders fit the structure of the training data, making the initial value of the entire network in a suitable state and speeding up training convergence.

pCNN and mCNN: To evaluate the efficiency of cplxCNN, we design two traditional CNNs using identical architectures as cplxCNN. The first one, pCNN, uses the phase information as input. The other one, mCNN, uses the signal magnitude as input. We keep the other settings of mCNN and pCNN consistent with those of cplxCNN.

Simple Machine Learning Techniques: We also evaluate our methods against simpler machine learning techniques on our data. Specifically, we implement and compare $K$-neighbor regression (KNR), linear regression (LR), and ridge regression (RR). The input of all the methods is the measured signal and the output is a $1 \times 512$ vector. We then fill the triangulation map with the corresponding output value.

B. MIT Dataset

In this work, we aim to bridge the gap between simulation and real-world applications by collecting a real-world dataset using our MIT system. To achieve this, we used two different methods: automatic collection using slide rails and manual collection.

As shown in Fig. 4, our MIT device comprises 16 coils along with their respective sensor modules. These coils are capable of both the receiving and transmitting signals. The sensor module consists of a direct digital synthesis (DDS) signal generator and a multiplexer. The 16 coils form our sensor array, which is evenly distributed along a circular arrangement with a diameter of 20 cm.

We use the DDS AD9958 module to generate sinusoidal signals. Specifically, we use a 32-bit tuning word capable of producing excitation signals with frequencies of 25, 28, and 31 MHz. The LTC2145-14 module is a two-channel 14-bit analog-to-digital converter (ADC). To drive the output current, we use the TSH3491 module, which is a high-speed, high-voltage, and low-distortion current-feedback amplifier. It has a bandwidth of 900 MHz and can provide an output current of 420 mA.

For the automated collection process, we design a device as shown in Fig. 5. This device consists of a controller, two screw slide rails equipped with servo motors, and the MIT device. The object is attached to the rail and positioned it at the starting point first. As shown in Fig. 5(c), each rail is driven by an independent servo motor, enabling object automatic movement on a 2-D plane.

To evaluate our method, it requires a diverse dataset consisting of real-world object measurements that differ in shapes, conductivity, and locations. Therefore, we use three containers containing salt water in our data collection, including two distinct cylinders with diameters of 30 mm (C-30) and 35 mm (C-35), and a triangular prism (PR). Their heights are identical at 8 cm. The dataset comprises a total of 3687 samples for C-30 and C-35, and 2247 samples for the triangular prism. We position the objects at 1229 and 749 locations for the cylinders and prism, respectively. Three samples are taken at each location.

To simulate realistic biological tissues, we also use NaCl-doped gels, i.e., agar, to build our dataset. Specifically, a cylinder agar block with the same diameter and height as C-30 is made by mixing with a solution of NaCl with a conductivity of $11.5 \text{ m}^{-1}$. The block was moved at a step of 5 mm in different radial directions. In total, we collected 240 locations for each agar block, and for each location, we acquired three frames.

To evaluate MITNet in practical and general scenarios [12], [57], we conduct additional data collection to fine-tune and evaluate our model. The performance of MITNet is tested using real-world objects, specifically carrots. In scenarios involving multiple objects, we use two agar cylinders, resulting in 20 combinations arranged at various locations. For the real-world experiment, two carrots are used to create five combinations. To prevent overgeneralization, it is important to note that the carrot data were excluded from the training set. Instead, we directly applied the network trained on agar data.

C. Results and Analysis

Fig. 6 shows the reconstructed images. Our model successfully restores the shape and location of objects while minimizing the noises from the measured signal. The “cplxCNN” column shows the result prior to GAN enhancement. cplxCNN can already significantly reduce noises and improve localization precision compared with other existing methods. For example, the traditional NR method shows significant noises around the object and close to the periphery electrode. The SAE and FCN methods improve on localization but still are inaccurate on the shapes. Our MITNet technique achieves the best performance overall.

In Fig. 7, it can be observed that MITNet achieves the best performance, with the highest IoU (81.08%) and lowest CD (3.59) on average, producing the reconstructed image that is closest to the ground truth. The GAN architecture, in addition to removing artifacts, also aids in reconstructing
the shapes of objects by leveraging the knowledge of the distribution of object data. Unlike other methods that use approximations during calculation and are easily influenced by noise, our MITNet model uses the complete information of the complex-valued signals, leading to better results. Conversely, the SAE and FCN methods use only the magnitude of data as input, leading to information loss in the training and testing processes.

We evaluate different methods on agar data. As shown in Fig. 8, MITNet achieves the highest IoU and F1 score, which is 10.4% and 5.3% better than the second-best technique, respectively. VDDNet is worse than MITNet, but better than pCNN and other simpler machine learning techniques such as LR. While SAE and mCNN exhibit higher precision than cplxCNN, their recall is lower by 12.4% and 14.7%, respectively. This indicates a higher rate of false negativity. Moreover, pCNN and mCNN are much worse than cplxCNN, which shows the necessity of using complex-valued data.

Simple machine learning techniques are generally slightly worse than the NR method, with F1 score of 44.92%, 21.92%, and 53.85% for KNR, LR, and RR, respectively, while our method achieves 88%, which is 34.15% better than the second best on F1 score. In general, for all the comparing metrics, our complex CNN technique outperforms the second best machine learning technique significantly, as shown in Fig. 8.

In Table I, we also show the training and inference time costs. All the methods are trained for 200 epochs, except for NR, which does not require training. In general, the training time increases with the complexity of the network. Our MITNet requires 5934 s to finish training. For inference time, we use a batch size of 8 and calculate the average time per image. Although MITNet needs longer inference time compared with other baselines, it is also much more accurate. In many applications, such as medical imaging, 3-s (23.57/8) processing time is sufficient while accuracy is much more important.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
MITNet also performs well when the diameter of the object changes, achieving an IoU of 88.61% and 86.68% when the object’s diameter changes from 35 to 30 mm, respectively.

On the triangular prism data, MITNet outperforms other algorithms, achieving the highest IoU of 71.47%. The original output resolution becomes coarse-grained after finite element division, and even after smoothing, it is challenging to recognize the shape and size of the object. For instance, in Fig. 6, the reconstructed images of SAE on cylinder and prism are very similar, but MITNet can differentiate between the prism and cylinder images, producing sharp edges in the prism images.

To demonstrate the effectiveness of artifact removal, we apply the same GAN technique to enhance the output of the NR, FCN, and SAE algorithms. In Table II, SAE and NR show significant performance improvement on both 35 and 30 mm cylinder data, achieving IoU of 90.12%/85.05% and 86.46%/82.48%, respectively, while also reducing the CD. However, they perform poorly on the triangular prism dataset, indicating a limited ability to recognize object shapes. As shown in Table III, the F1 score of VDDNet increased from 30.07% to 50.20%, indicating the effectiveness of GAN enhancement. However, its performance still falls behind our MITNet technique. The performance of simple machine learning techniques can also be improved by 3.46% on average in terms of F1 score after GAN enhancement. However, our deep-learning-based technique still outperforms the machine-learning-based methods by 44.27% on average. These results demonstrate that the traditional simpler machine learning techniques are inadequate for such complex and highly nonlinear MIT reconstruction tasks. The results validate the effectiveness of GAN in MIT image reconstruction. Overall, MITNet remains the best algorithm on all the datasets, achieving the highest IoU and the lowest CD on average.
We fine-tune MITNet using multiobject data and evaluate its performance. As shown in Fig. 9, cplxCNN demonstrates an accurate prediction of carrot locations with little noise. After GAN enhancement, MITNet records excellent performance, with IoU, precision, recall, and F1 scores of 77.14%, 92.78%, 75.43%, and 82.32% respectively.

We also conduct tests on MITNet under limited conditions. Specifically, we consider scenarios where only a portion of the coils are available, i.e., half of the coils cannot be accessed. In that case, we make adjustments to the structure of our model to accommodate the new inputs. The results in Table IV indicate almost identical performance for the MITNet, with a 1.8% decrease in precision and a 1.0% decrease in the F1 score. Moreover, we show examples of the prediction result in Fig. 10. Therefore, the reduction in performance is caused by the information loss of the cplxCNN.

### VI. Conclusion

In this article, we design a bio-impedance-based imaging system and propose a novel deep learning framework, MITNet, for image reconstruction. The system uses a set of excitation-sensing coils to induce eddy currents in the internal object field with a fixed frequency current. The device is radiation-free and has the potential for application in the monitoring and diagnosis of cerebrovascular diseases. We also collect a real-world dataset based on our system and evaluate our method with the existing algorithms. The experimental results show that the images reconstructed by our MITNet have the highest quality, indicating that our system has great potential for future biomedical imaging studies.

### REFERENCES

[1] H.-Y. Wei, L. Ma, and M. Soleimani, “Volumetric magnetic induction tomography,” Meas. Sci. Technol., vol. 23, no. 5, May 2012, Art. no. 055401.

[2] H.-Y. Wei and M. Soleimani, “Four dimensional reconstruction using magnetic induction tomography: Experimental study,” Prog. Electromagn. Res., vol. 129, pp. 17–32, 2012.

[3] J. Shi, X. Cao, F. Liu, B. Zhang, J. Luo, and J. Bai, “Greedy reconstruction algorithm for fluorescence molecular tomography by means of truncated singular value decomposition conversion,” J. Opt. Soc. Amer. A, Opt. Image Sci., vol. 30, no. 3, p. 437, 2013.

[4] J. Zhao, L. Ye, F. Han, L. Xiong, and Q. H. Liu, “Multiparametric electromagnetic inversion of 3-D biaxial anisotropic objects embedded in layered uniaxial media using VBM enhanced by structural consistency constraint,” IEEE Trans. Antennas Propag., vol. 68, no. 6, pp. 4774–4785, Jun. 2020.

[5] M. Soleimani and W. R. B. Lionheart, “Absolute conductivity reconstruction in magnetic induction tomography using a nonlinear method,” IEEE Trans. Med. Imag., vol. 25, no. 12, pp. 1521–1530, Dec. 2006.

[6] R. Chen et al., “A novel algorithm for high-resolution magnetic induction tomography based on stacked auto-encoder for biological tissue imaging,” IEEE Access, vol. 7, pp. 185597–185606, 2019.

[7] J. Xiang, Y. Dong, and Y. Yang, “FISTA-Net: Learning a fast iterative shrinkage thresholding network for inverse problems in imaging,” IEEE Trans. Med. Imag., vol. 40, no. 5, pp. 1329–1339, May 2021.

[8] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, “Unet++: Redesigning skip connections to exploit multiscale features in image segmentation,” IEEE Trans. Med. Imag., vol. 39, no. 6, pp. 1856–1867, Jun. 2020.

[9] D.-P. Fan et al., “Inf-Net: Automatic COVID-19 lung infection segmentation from CT images,” IEEE Trans. Med. Imag., vol. 39, no. 8, pp. 2626–2637, Aug. 2020.

[10] R. K. Samala, H.-P. Chan, L. Hadjiiski, M. A. Helvie, C. D. Richter, and K. H. Cha, “Breast cancer diagnosis in digital breast tomosynthesis: Effects of training sample size on multi-stage transfer learning using deep neural nets,” IEEE Trans. Med. Imag., vol. 38, no. 3, pp. 686–696, Mar. 2019.

[11] H. Kang et al., “Diagnosis of coronavirus disease 2019 (COVID-19) with structured latent multi-view representation learning,” IEEE Trans. Med. Imag., vol. 39, no. 8, pp. 2606–2614, Aug. 2020.

[12] J. Xiang, Y. Dong, and Y. Yang, “Multi-frequency electromagnetic tomography for acute stroke detection using frequency-constrained sparse Bayesian learning,” IEEE Trans. Med. Imag., vol. 39, no. 12, pp. 4102–4112, Dec. 2020.

[13] C. Cai, M. Diwakar, D. Chen, K. Sekihara, and S. S. Nagarajan, “Robust empirical Bayesian reconstruction of distributed sources for electromagnetic brain imaging,” IEEE Trans. Med. Imag., vol. 39, no. 3, pp. 567–577, Mar. 2020.

[14] Q. Xuan et al., “Automatic pearl classification machine based on a multistream convolutional neural network,” IEEE Trans. Ind. Electron., vol. 65, no. 8, pp. 6538–6547, Aug. 2018.

[15] Z. Chen et al., “SigNet: A novel deep learning framework for radio signal classification,” IEEE Trans. Cognit. Commun. Netw., vol. 8, no. 2, pp. 529–541, Jun. 2022.

[16] Q. Xuan, Z. Chen, Y. Liu, H. Huang, G. Bao, and D. Zhang, “Multiview generative adversarial network and its application in pearl classification,” IEEE Trans. Ind. Electron., vol. 66, no. 10, pp. 8244–8252, Oct. 2019.

[17] J. Wang, J. Li, Y. Chen, F. Han, and Q. H. Liu, “Simulation of 3-D electromagnetic scattering and inverse scattering by arbitrary anisotropic dielectric objects embedded in layered arbitrary anisotropic media,” IEEE Trans. Antennas Propag., vol. 68, no. 8, pp. 6473–6478, Aug. 2020.

[18] L. Li, L. G. Wang, F. L. Teixeira, C. Liu, A. Nehorai, and T. J. Cui, “DeepNIS: Deep neural network for nonlinear electromagnetic inverse scattering,” IEEE Trans. Antennas Propag., vol. 67, no. 4, pp. 1819–1825, Mar. 2019.

[19] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” Proc. IEEE, vol. 86, no. 11, pp. 2278–2324, Oct. 1998.

[20] M. Arjovsky, A. Shah, and Y. Bengio, “Unitary evolution recurrent neural networks,” in Proc. Int. Conf. Mach. Learn., 2016, pp. 1120–1128.
