

# Machine Learning Target Count Prediction in Electromagnetics Using Neural Networks

Mohsen Sabbaghi<sup>1</sup>, Graduate Student Member, IEEE, Jun Zhang<sup>2</sup>, Senior Member, IEEE, and George W. Hanson<sup>3</sup>, Fellow, IEEE

**Abstract**—In this article, we showcase an application of neural networks (NNs) to solve an inverse problem in electromagnetics (EMs). Wires are randomly distributed into an area of known dimensions. The wires are then illuminated with a monochromatic plane wave (PW) at a certain angle of incidence, and the EM field measured at a finite number of uniformly spaced points along the perimeter of the area is then fed into a convolutional neural network (CNN) designed to predict the number of wires. Counting the wires is posed as a supervised classification problem with a known upper limit to the number of wires, and accuracy of 96% has been achieved for the case where the number of the wires is known to be ten or less. A number of approaches have been taken to improve the network performance including frequency variation analysis and illuminating the wire distributions with additional PW angles of incidence. We conclude with an analysis of the network capability to resolve objects based on its performance on known wire distributions, which suggests the existence of a characteristic resolution limit corresponding to the CNN topology.

**Index Terms**—Artificial intelligence, HF antennas, inverse problems, neural networks (NNs), wire scatterers.

## I. INTRODUCTION

**R**EAL-WORLD problems in engineering often require the recovery of model variables from observations, which is an inverse problem. This recovery can be formally represented by a mapping from the space of observables into the space of model variables [1].

An example of inverse problems in electromagnetics (EMs), which is of great significance in biomedical engineering, is the detection and identification of tissue anomalies, such as malignant tumors. In [2], electrical contrast maps of breast tissue have been extracted from the scattering parameters measured at 800 MHz using transceivers surrounding the breast tissue. Another example, which is of special interest in an aerial target recognition, is the microwave imaging of flying aircraft that was achieved in [3] and [4] by processing the localized radar cross section (RCS) measurements acquired at 9.6 GHz at distances varied from 3 to 20 km.

Manuscript received 17 May 2021; revised 12 September 2021; accepted 21 September 2021. Date of publication 14 October 2021; date of current version 8 September 2022. (Corresponding author: Mohsen Sabbaghi.)

The authors are with the Department of Electrical Engineering and Computer Science, University of Wisconsin–Milwaukee, Wisconsin, WI 53211 USA (e-mail: sabbagh2@uwm.edu; junzhang@uwm.edu; george@uwm.edu).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TAP.2021.3118799>.

Digital Object Identifier 10.1109/TAP.2021.3118799

## A. Neural Networks (NNs) and Inverse EM Problems

Since inverse problems are mostly ill-posed, i.e., a single set of observations corresponds to several possible sets of model variables, conventional inversion methods are not always applicable [1]. A remedy to ill-posedness is to impose constraints on the inverse problems. NNs have been demonstrated to be very effective tools in the machine learning (ML) toolbox, which makes it possible to impose constraints on inverse problems by leveraging enough samples of training data. Therefore, NNs offer a data-driven tool to estimate the inverse mapping in ill-posed inverse problems in EMs.

Convolutional neural networks (CNNs) have been demonstrated to be essential in recovery problems, such as MRI image reconstruction [5]. Recent applications of ML in optics include: 1) the inverse design of integrated photonic power splitters [6], metasurfaces [7], and nanophotonic particles [8], [9]; 2) the optimization of photonic structures [10] and dielectric metasurfaces [11]; and 3) the prediction of the optical properties of plasmonic structures [12]. Another example is electrical impedance tomography (EIT) in two and three dimensions, which was achieved in [13], by feeding the electrostatic voltage measurements acquired from the electrodes surrounding the area into a combination of BCR-Net [14], [15] and CNNs. In addition, permittivity contrast maps have been recovered by feeding surrounding receiver antenna readings at 400 MHz, 3 GHz, and 4 GHz into CNNs [16]–[20].

## B. Inverse Problem of Counting in EMs

Counting objects based on the EM field measurements is another example of inverse problems in EMs. In [21], the data acquired from postdetection processing of high-resolution Doppler radar signals have been utilized to count aircraft targets in a moving cluster, and correct-count probabilities of 93% and 87% have been achieved respectively on single- and double-target data. In [22], the signals acquired by measuring the EM field backscattered off passive RFID tags at 924 MHz were processed to count the number of people present in a room, which was achieved by establishing a correlation between the variations in the signal strength and the number of people.

In [23], range-velocity radar images acquired at 57–64 GHz were fed into a deep residual U-net architecture [24] to detect and count human targets. Moreover, in [25], the counting

of one to six moving human targets was performed with an accuracy of 91% by feeding range-time radar images acquired at 1.6–2.2 GHz into a residual NN, and similar counting of one to three moving human targets was performed with an accuracy of 97%.

In this work, we study the problem of counting identical wires using EM field values at  $f = 10$  MHz (equivalent to a free-space wavelength of  $\lambda \cong 30$  m) using a CNN. For the purpose of fast generation of synthetic training/testing data, the objects are assumed to be vertically oriented thin wires. The counting problem is posed as a supervised classification problem, and a number of approaches are discussed to improve network performance. This study provides a proof of concept for the application of NNs in counting identical objects in EMs.

The CNN that is trained on the synthetically generated wire data should not be applied to other counting problems, even to the same wire counting setup in the laboratory, because the computed EM field data might not be an accurate representation of the actual antenna measurements. Still, the choice of wires allows us to synthetically and rapidly generate as many samples as needed for parameter-sweep ML experiments to obtain estimates of the number of antennas, the number of training samples, the source illumination wavelength, the number and placement of illumination sources, and so on, which yields maximal learning before moving to the “laboratory” stage where the identical objects are not necessarily wires, and the EM field due to each camera-recorded object distribution is measured using the surrounding monopole antennas.

Of particular interest to this work is understanding the advantages that could be gained by using ML in recognizing two objects that are separated by a distance smaller than  $\lambda$ , which is motivated by previous studies in optics, acoustics, and EMs. To name a few, in [26], 1-D CNNs were utilized to process the information retrieved from measuring the EM field scattered off subwavelength silicon nanostructures, which, subsequently, led to pushing the limits of optical information storage in the visible frequency range [27]. In [28], subwavelength acoustic imaging has been experimentally achieved by feeding the acoustic signals recorded by a microphone array placed in the far-field into a “U-net-type” CNN. Moreover, in [29] and [30], the imaging of subwavelength quasi-periodic structures of circular rods was performed by feeding transient scattered field data acquired at a  $\sim 3$  GHz into CNNs and RNNs to identify the location of the missing rod. For the first time, to the best of our knowledge, our work demonstrates the existence of a resolution threshold of NNs in the context of counting objects based on the measured EM field (for the network described here, we find a resolution to be limited to be approximately  $0.1\lambda$ ).

This article is structured as follows. The specifics of the inverse problem are detailed in Section II. The generation of training data and the network structure are described in Section III. Next, a number of approaches are discussed in Section IV to improve the performance of the CNN. Failure analysis of the samples for which the CNN makes incorrect predictions is included in Section V, which will be followed by conclusions in Section VI.

## II. FORWARD AND INVERSE PROBLEMS

### A. General Framework for Identical Objects

In general, we consider the forward problem to be the computation of the EM field in the presence of  $N$  identical objects when illuminated with a monochromatic plane wave (PW) of frequency  $f$  at an angle of incidence  $\theta$ . Therefore, in the forward problem, the total EM field is a function of: 1)  $f$ ; 2)  $\theta$ ; 3) the location of the point where the EM field is measured,  $\mathbf{r}$ ; and 4) the distribution of objects. Here, the term “object distribution” refers to the positions of all  $N$  objects denoted by  $\mathbf{R} = [\mathbf{R}_1, \dots, \mathbf{R}_N]$ . Therefore, the forward problem can be formulated as

$$\mathbf{E} = \mathbf{E}^{\text{sc}} + \mathbf{E}^{\text{inc}} = \mathbf{F}(\mathbf{R}, \mathbf{r}, f, \theta) \quad (1)$$

where the unsubscripted vector  $\mathbf{E}$  denotes the total electric field, the superscripts “sc” and “inc,” respectively, stand for scattered and incident, and  $\mathbf{F}$  is a vector-valued function that represents the forward mapping from the  $(\mathbf{R}, \mathbf{r}, f, \theta)$  space into the space of the observable total EM field. Here, the terms “total” and “incident” are used to refer to the EM field measured, respectively, in the presence and the absence of the objects, and the subtraction of the latter from the former results in what is referred to as the “scattered” EM field. The inverse problem constitutes the computation of the number of objects from the EM field measured at  $P$  points whose position are denoted by  $\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \dots, \mathbf{r}_P$ . As it can be seen from (1), the dependence of the total EM field on the number of objects is implicit. The inverse problem can be formulated as

$$N = \mathcal{G}(\mathbf{v}_{\text{in}}) \quad (2)$$

where  $\mathcal{G}$  denotes the (inverse) mapping of the total EM field measurements,  $\mathbf{v}_{\text{in}}$ , into the number of objects

$$\mathbf{v}_{\text{in}} = \begin{bmatrix} \mathbf{E}(\mathbf{R}, \mathbf{r}_1, f, \theta) \\ \mathbf{E}(\mathbf{R}, \mathbf{r}_2, f, \theta) \\ \vdots \\ \mathbf{E}(\mathbf{R}, \mathbf{r}_P, f, \theta) \end{bmatrix}. \quad (3)$$

### B. ML Approach in Counting Wires

In this work, ML, in particular NNs, has been utilized as a data-driven tool to estimate the mapping  $\mathcal{G}$  from the field measurements to the number of scatterers. Since the EM field measurements of a general  $N$ -object system could contain a large number of features, the training dataset is required to contain a large number of samples [31]–[33].

The geometry of the problem of interest is depicted in Fig. 1, showing a square region, 100 m  $\times$  100 m in area, containing ten wires and illuminated by a PW (see figure caption for details). In a real-world problem, the training data can be acquired by measuring the EM field from antennas located at the perimeter of the box for a large number of wire configurations. However, here, we use synthetic generation of training data, allowing us to control the number of “antennas,” control the amount of data necessary for training the network, and acquire data without noise so that the impact of noise can be studied separately. Depending on the type and shape

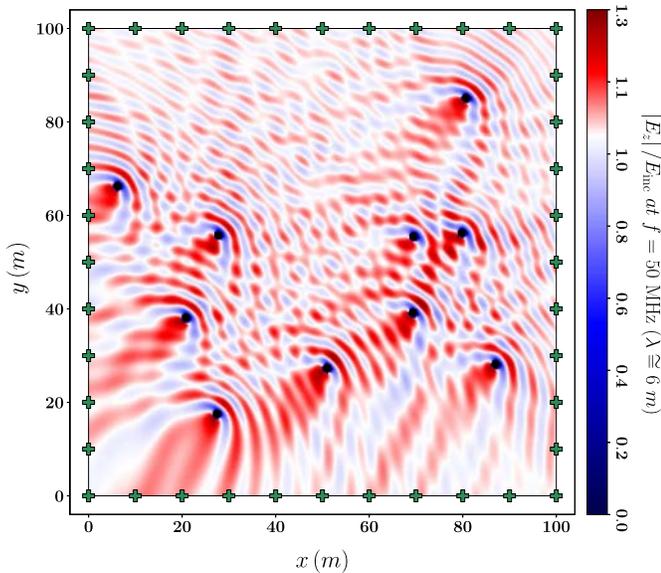


Fig. 1. Amplitude of a  $z$ -polarized electric field for a system of  $N = 10$  randomly distributed and  $z$ -directed thin wires illuminated with a PW of frequency  $f = 50$  MHz ( $\lambda \cong 6$  m) at an angle of  $\theta = 45^\circ$ . The field values are generated over a  $300 \times 300$  meshgrid using the formalism provided in Appendix I. The wire length and radius are set to be  $l = 3$  m and  $a \cong 0.0498l \cong 14.9$  cm, respectively, and for graphical reasons, the EM field data are excluded from circular regions of radius  $7a \cong 106$  cm centered at each wire. These exclusion areas are mostly covered by black-filled circles that represent wires but with an exaggerated radius. The box sides are  $L = 100$  m long, and the plus symbols represent the  $P = 40$  antennas that are located at the perimeter of the box for EM field measurement. The EM field measurements at the location of antennas are to be fed into our NN, which will be discussed in detail in Section III-A.

of the object, the generation of training data by computing the EM field for a large number of randomized instances, of the order of  $10^5$ , could lead to relatively large run times. Under this constraint, we chose the identical objects in our study to be thin wires because each wire can be approximated by an electric dipole, which leads to tractable run times for data generation. Following the formulation in [34], the set of electric dipole moments corresponding to a given wire geometry can be obtained self-consistently by solving a matrix equation.

To maximize the electric dipole moment induced in each wire due to the electric field at the location of the wire, the incident EM field is set to have a purely  $z$ -directed electric field. To reduce the degrees of freedom in this problem, we assume the wires to be thin, long,<sup>1</sup> and centered at  $z = 0$ , the plane at which the EM field data are to be acquired. This problem is well-studied [34]–[36], and the analytic formalism for the scattered EM field taken from [34] can be found in Appendix I. In this geometry, the  $z$ -component of the scattered magnetic field and the  $x$ - and  $y$ -components of the scattered electric field, i.e.,  $H_z^{sc}$ ,  $E_x^{sc}$ , and  $E_y^{sc}$ , are zero, and we construct the input vector only from the  $z$ -component of the electric

<sup>1</sup>The assumption of the wires being long and thin is to rule out the possibility of rotational (Eddy) currents being induced in each wire. Otherwise, magnetic dipole moment and other higher order terms will be needed in addition to the electric dipole moment term to compute the scattered EM field due to each wire.

field. For a case when ten  $z$ -directed wires are illuminated with a PW at  $f = 50$  MHz at an angle of incidence of  $\theta = 45^\circ$ , the values for  $E_z$  are obtained using (12) and are shown in Fig. 1.

### III. DESIGN OF THE NN

In this work, the problem of counting the wires is approached as a classification problem. In this approach, samples containing the same number of wires belong to the same class regardless of the location of wires. Since the number of classes has to be finite, we assume an upper bound to the number of wires in each sample denoted by  $N_{\max}$ . This implies our knowledge of the number of wires in all samples being less than or equal to  $N_{\max}$ . Constructing the desired output vector will be discussed in more detail in Section III-A.

#### A. Generation of Training Data

All the datasets used for training and validation of our NNs consist of 300 000 samples, each of which corresponds to a randomized wire distribution (RWD). An RWD is generated by choosing a random number,  $N$ , from  $\{1, \dots, N_{\max}\}$  as the number of the  $z$ -directed wires, and then for the  $i$ th of those  $N$  wires, two random numbers,  $x_i$  and  $y_i$ , are chosen from the range of  $[\delta, L - \delta]$  as the wire location within a square box of length  $L = 100$  m in the  $xy$  plane. The parameter  $\delta$  leaves a margin between the perimeter of the area where the wires are distributed and the sides of the square box at which the antennas are located, and throughout this article, we set  $\delta = 0.02 L$ . The length  $l$  and radius  $a$  of each wire are assumed to be 10 m and 1 cm, respectively. Since the problem aims at predicting the number of wires in each sample, each sample should be labeled by its desired output. The desired output corresponding to the class of samples containing  $N$  wires is constructed using one-hot encoding, which results in an  $N_{\max}$ -dimensional sparse vector with  $N_{\max} - 1$  of its components being equal to zero and the  $N$ th component being the only nonzero component and equal to 1, that is,

$$\mathbf{v}_{\text{out}} = \left[ 0, 0, 0, \dots, \underbrace{1}_{N\text{-th}}, \dots, 0, 0 \right]^T. \quad (4)$$

The input vector, on the other hand, is a 40-D vector containing the  $E_z$  values computed at  $P = 40$  uniformly spaced points located at the edges of the box. Since the calculated EM field is complex-valued, the input vector is two-folded with each fold containing the real and imaginary parts, similar to the way the intensity of the red, green, and blue components of each of the pixels of an RGB image is stored in three separate arrays [37].

#### B. From Fully Connected to CNNs

In the earlier stages of this study, we made use of fully connected NNs, and evaluating the trained networks on samples outside the training dataset indicated that the networks memorized the training data. That is, training the network on a certain dataset led to accuracies as high as 98%, but the

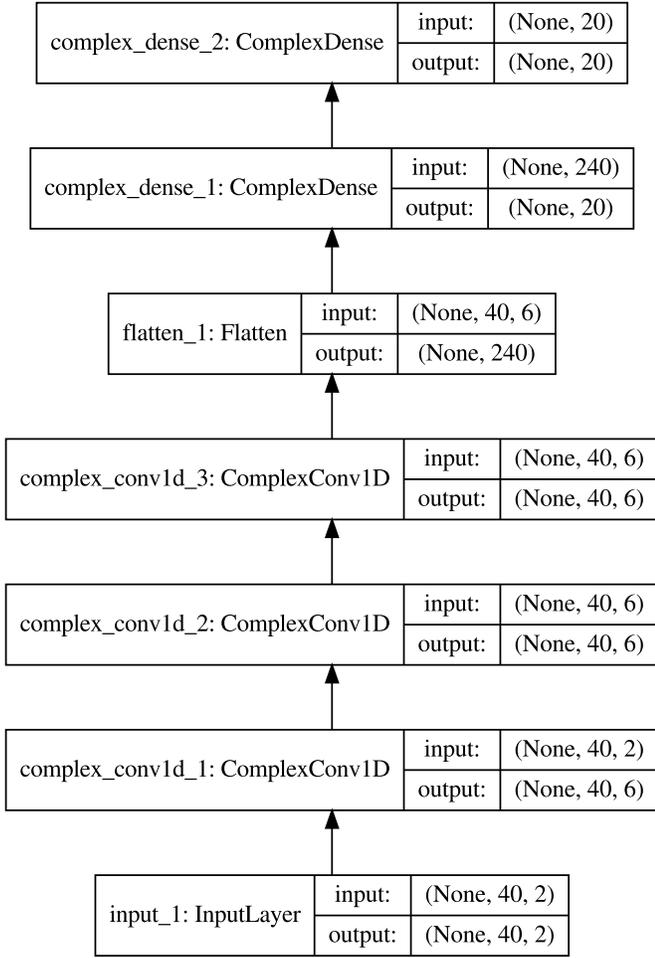


Fig. 2. Topology of the complex-valued CNN utilized in our work. For the first three consecutive 1-D convolution layers, the intermediate dense layer, and the last dense layer, the activation function is, respectively, set to be exponential linear unit (ELu), rectified exponential linear unit (RELu), and softmax. Each of the first three convolution layers contains three convolution filters of length 3, which moves over each channel with steps of 1. The output of a convolution is zero-padded such that the length of each channel is not affected by the convolution. This CNN contains 2784 trainable parameters (weights and biases). The dimension labeled as “None” corresponds to indexing the samples within the same batch. Since the batch size is set to be 500, this index can take values from 0 to 499.

trained network made poor predictions for samples outside the training dataset. This can be understood by the fact that those fully connected NNs contained too many weights ( $\sim 300\,000$ ) and are, therefore, prone to overfitting [38].

To reduce the number of weights, we replaced the first three fully connected layers with 1-D convolutional layers with three filters each with a convolution window size of 3. All three weights in each of the convolution windows are accompanied by a bias. Since the input vector consists of two folds, that is, the real fold and the imaginary fold, operating three convolution filters along with each fold results in a six-fold output of the first convolutional layer. In this case, each convolutional layer consists of only  $6 \times (3 + 1) = 24$  trainable parameters, which is considerably smaller comparing to the case where the first three layers were fully connected.

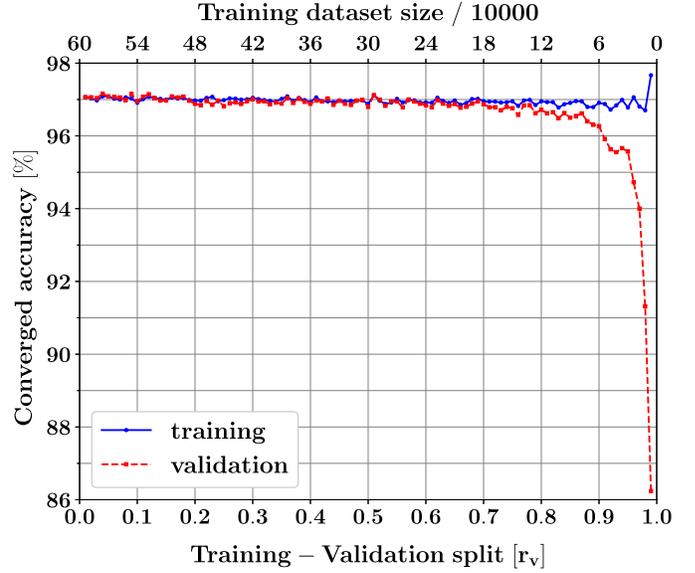


Fig. 3. Converged accuracy and converged validation accuracy achieved from training the CNN described in Fig. 2 on a training set containing  $3 \times 10^5 \times (1 - r_v)$  for  $r_v \in \{0.01, 0.02, \dots, 0.98, 0.99\}$ . For each value of  $r_v$ , the CNN is trained for 500 epochs, and the average of the (validation) accuracy achieved in the last ten epochs is shown by a data point. It is clear that, as the size of the training dataset drops below  $10^5$ , the generalization capability of the CNN worsens even though its learning from the samples within the training dataset improves. This experiment determines the minimum size of the training dataset from which the CNN is able to learn the most. Therefore, 270 000 of samples are set aside for training the CNN, i.e.,  $r_v = 0.1$ , and another separate and independent set of 30 000 samples is used for validating the model in each epoch of training.

Ideally, the CNN is expected to output  $N_{\max}$ -dimensional one-hot vectors when fed with input vectors of the measured total field. However, in reality, the CNN outputs a normalized vector with  $N_{\max}$  nonzero components, whose  $i$ th component represents the probability that the sample corresponding to the input vector belongs to the  $i$ th class, i.e., contains  $i$  wires. The output vector of the CNN is denoted by  $\hat{\mathbf{v}}_{\text{out}}$ . This requires that the activation function of the last layer is set to “softmax,” which is described as follows:

$$\sigma(u_1, \dots, u_{N_{\max}})_k = \frac{e^{u_k}}{\sum_{k=1}^{N_{\max}} e^{u_k}} \quad (5)$$

where  $k = 1, 2, 3, \dots, N_{\max}$  and  $\mathbf{u} = [u_1, \dots, u_{N_{\max}}]$  is the input to the last layer. This dense layer is fully connected to the last convolution layer and is fed with the flattened output of the last convolution layer. The CNN was constructed using the complexnn PYTHON library [39], [40] with TensorFlow backend, and the corresponding network topology is shown in Fig. 2. It is worthy to note that, unlike the approach in [5], this treatment of complex-valued input data requires the separation of real and imaginary components as two separate channels. Reference [41] is an example of a recent use of the complexnn library; however, since the authors did not prepare their complex-valued input data into two separate real and imaginary channels, their complex-valued CNN could only work by replacing the first ComplexConv1D layer with a regular one, i.e., Conv1D, to avoid any input shape mismatch. The learning process of the CNN involves

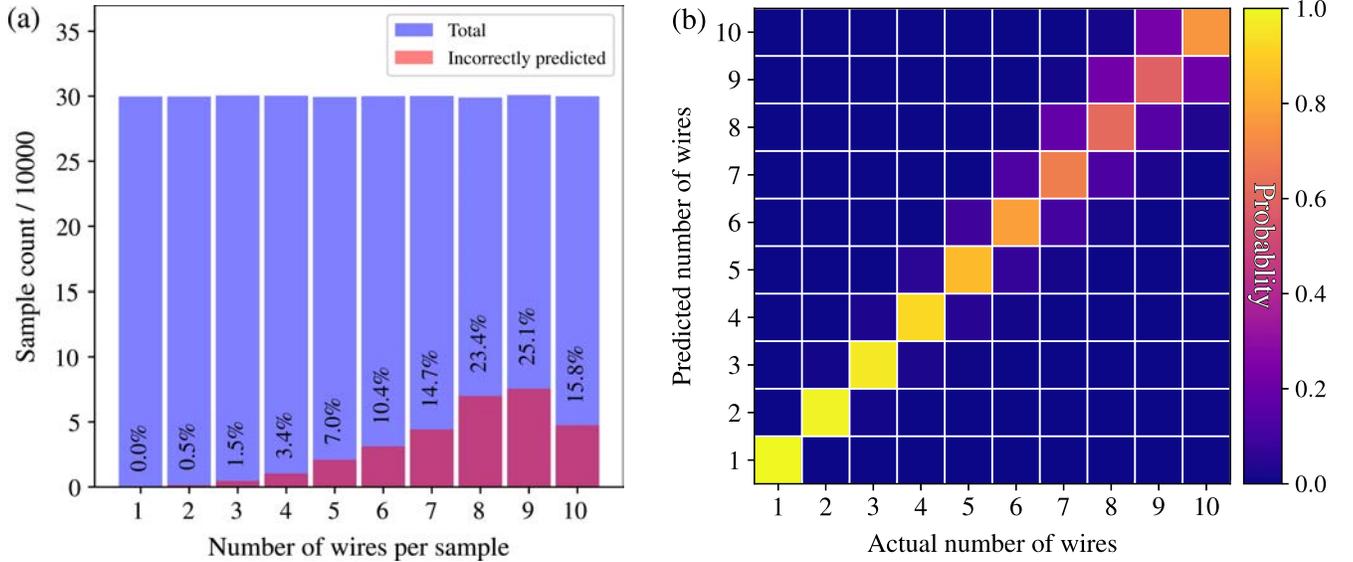


Fig. 4. Evaluation of the CNN performance trained on 270 000 samples containing up to  $N_{\max} = 10$  wires. All RWDs are illuminated with a PW at  $f = 10$  MHz and at an angle of incidence of  $\theta = 45^\circ$ . The training process involved online-testing (validation) of the CNN on another independent 30 000 samples to monitor the generalization capability of the CNN. After the training and the online-testing processes are completed, a set of  $3 \times 10^6$  new samples were generated for final performance evaluation. Operating the trained CNN on the  $\mathbf{v}_{\text{in}}$  of each sample returns an output vector  $\hat{\mathbf{v}}_{\text{out}}$  from which the predicted number of wires can be obtained. Within the set of  $3 \times 10^6$  new samples, an overall accuracy of 89% is achieved. (a) Accuracies achieved when the set of  $3 \times 10^6$  new samples is divided into  $N_{\max} = 10$  subsets each containing samples with a given number of wires. The total number of samples within each subset that is fed into the CNN is shown with blue bars, while the red bars show the number of incorrect predictions. The error percentage, which is the ratio of the number of incorrect predictions to the total number of predictions within each subset, is shown on the corresponding bar. (b) Average of the probability vectors returned by the CNN for the samples within the  $N$ th subset for  $N \in \{1, \dots, N_{\max}\}$ .

minimizing the categorical cross entropy (CCE) loss function given by

$$\text{CCE} = - \sum_{k=1}^{N_{\max}} v_{\text{out},k} \times \log_{10} [\hat{v}_{\text{out},k}] \quad (6)$$

with  $v_{\text{out},k}$  and  $\hat{v}_{\text{out},k}$  being the  $k$ th components of the desired output vector and the CNN output vector, respectively. Since  $\mathbf{v}_{\text{out}}$  is one-hot encoded, the loss function corresponding to a sample containing  $N$  wires is simply given by  $\text{CCE} = -\log_{10} [\hat{v}_{\text{out},N}]$ . Upon feeding the input vector corresponding to each sample into the NN, the loss function and its partial derivative with respect to each of the trainable parameters are evaluated, which are then used to adjust the trainable parameters in the next iteration to lower the loss function. This adjustment could be done based on the *averaged* loss function and its partial derivatives calculated from the values obtained upon propagating each of the samples of a given unclassified subset of the training dataset [42]. This subset and its size are, respectively, referred to as “batch” and “batch size.” In principle, each batch could contain at least one sample and at most the whole samples within the training dataset. For all instances of the training of the CNN, a batch size of 500 has been used.

To verify whether the CNN is capable of generalization, i.e., to make correct predictions for samples that do not belong to the training dataset, a fraction of the samples is set aside from the dataset for validation purposes. The subset containing these samples is referred to as the “validation”

dataset. The ratio of the number of the samples within the validation dataset to the total number of samples is referred to as the “training–validation split” and denoted by  $r_v$ . In each epoch, after the CNN is trained on all samples within the training dataset, the accuracy of the predictions made by the CNN on the samples belonging to the validation dataset is evaluated. If the accuracies achieved in training and validation processes are reasonably close, then the CNN proves to be capable of generalization. Inspired by [31]–[33], we carried out an analysis in which  $r_v$  was varied from 0.01 to 0.99 in increments of 0.01 to determine the size of the validation subset. As shown in Fig. 3, a training–validation split of  $r_v = 0.1$ , which corresponds to a validation subset size of  $3 \times 10^4$ , yields fairly similar training and validation accuracies while leaving enough samples for training. Moreover, the flat behavior of both the training and validation accuracy curves versus  $r_v$  in Fig. 3 suggests that adding more samples to the training subset does not enhance the learning of the CNN from the data, and therefore, a training subset size of  $27 \times 10^4$  is adequate.

Since the training subset contains  $27 \times 10^4$  samples, and a batch size of 500 is used, each “epoch” involves  $27 \times 10^4 / 500 = 540$  adjustments of the trainable parameters, i.e., it takes 540 iterations to propagate all of the samples of the training subset through the CNN. If the training process involves  $N_E$  epochs, the full training of the CNN involves propagating all samples of the training subset through the CNN for  $N_E$  times. For the training processes in this article, we set  $N_E = 100, 500, \text{ and } 1000$ .

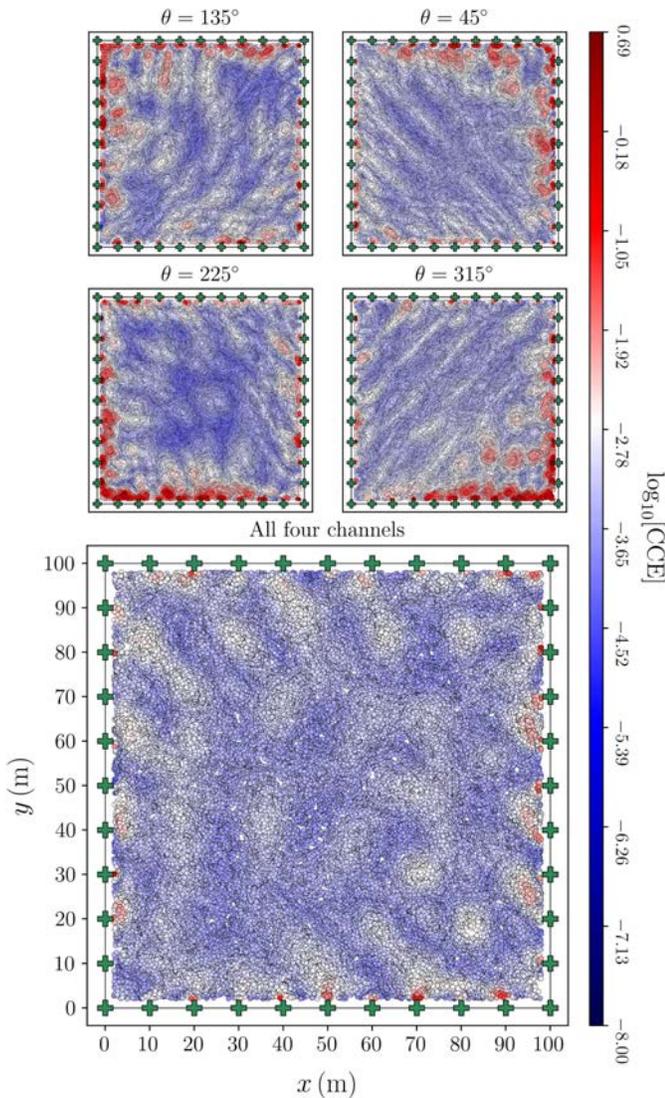


Fig. 5. In each of the five panels, each point represents the wire position in single-wire samples and is colored with the prediction error made by the CNN in predicting the number of wires of the corresponding sample. Four cases where the same RWD is illuminated with PWs at 10 MHz at only one angle of incidence ( $45^\circ$ ,  $135^\circ$ ,  $225^\circ$ , or  $315^\circ$ ) compared with a case where the input vector consists of all the four channels acquired by those separate illuminations. For all five cases, the CNN is trained for 100 epochs with a training-validation split of 0.5. To achieve a visual contrast in the prediction error map, a logarithmic scale is used to color-map the calculated CCE values.

### C. Learning From Total EM Field Measurements and the Requirement for Nontrainable Parameters

Initially, the CNN was trained on the input vectors containing the scattered EM field measurements, which led to predictions made with high accuracies on samples outside the training subset. However, the CNN was not capable of learning from the total field measurements at frequencies as low as 10 MHz. This can be understood by the fact that the scattered field signal due to a given RWD becomes much smaller compared to the incident field signal at 10 MHz, and as a result, the scattered field signal gets “buried” in the incident field signal.

By definition, the scattered field signal can be retrieved by subtracting the EM field measurements obtained from antennas

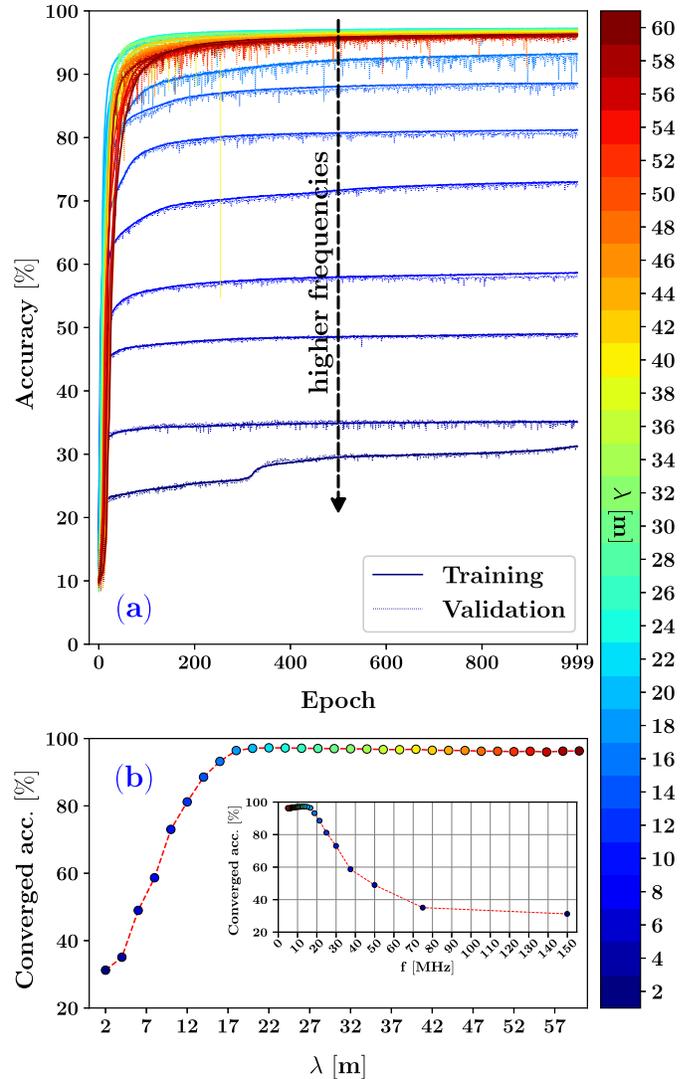


Fig. 6. Convergence of the CNN of the topology described in Section IV-A1 for the  $N_{\max} = 10$  case. (a) Learning curves of the CNN when trained on 30 datasets each containing the  $E_z$  field data generated at 30 wavelengths ranging from 2 to 60 m. For all 30 cases, the same set of hyperparameters are used, which includes a learning rate of 0.001, a momentum rate of 0.90, and a batch size of 500. In each case, the network is trained on a subset of 270 000 samples and validated on a separate subset of 30 000 samples. (b) Dependence of the accuracy to which the CNN converges on the wavelength at which the training data are generated. Inset: dependence of the converged accuracy values on the frequency at which the training data are generated.

with no wire(s) present from their counterparts measured in the presence of wire(s). Assuming that we have no knowledge of the source that transmits the incident field and measuring the EM field in the absence of wire(s) not being possible, we suggest a similar numerical remedy to this situation. This remedy involves the subtraction of a “constant”  $P$ -dimensional vector, which we refer to as the background vector and denote it by  $\mathbf{v}_{\text{BG}}$ , from all the  $P$ -dimensional vectors containing the total field measurements before being fed into the CNN. Since all the measured total field vectors corresponding to either training or testing/validation samples should be subtracted by the very same  $\mathbf{v}_{\text{BG}}$  before being fed into the CNN, the  $P$  components of  $\mathbf{v}_{\text{BG}}$  are implied to be the *nontrainable* parameters

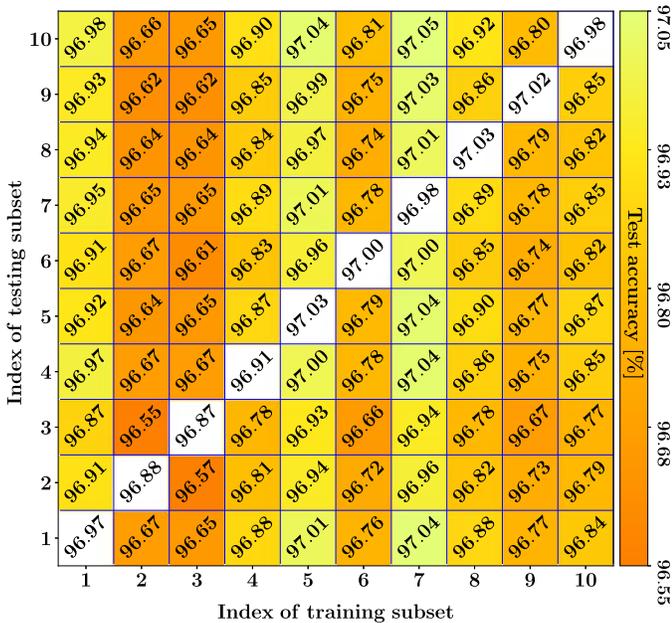


Fig. 7. Tabulated results for the tenfold cross-validation procedure that was carried out by: 1) grouping  $3 \times 10^6$  samples into ten independent datasets of equal size; 2) training the CNN on each of the ten independent subsets of  $3 \times 10^5$  samples; and 3) testing the trained CNN on each of the remaining nine subsets. The cell in the  $p$ th column and the  $q$ th row ( $p \neq q$ ) shows the accuracy obtained from testing the CNN on the  $q$ th subset after being trained on the  $p$ th subset. The white cell in the  $p$ th column and the  $p$ th row merely shows the accuracy obtained from training the CNN on the  $p$ th subset. Each of the off-diagonal cells is color-mapped with the testing accuracy value that it contains.

of the CNN. Here, we propose two methods to obtain  $\mathbf{v}_{BG}$ . In the first method,  $\mathbf{v}_{BG}$  is obtained by taking the average of all the total EM field vectors corresponding to samples in the training subset, which involves collective processing of the training data known as “centering.” We tested this method that proved to be effective.

In situations where the collective processing of the training data is not desired,  $\mathbf{v}_{BG}$  can simply be the total field measurements corresponding to a given RWD. Since we consider this method to be the most applicable to practical situations, we chose to implement this recipe in generating the input vectors fed into our CNN, that is,

$$\mathbf{v}_{in}^{[n]} = \begin{bmatrix} E_z(\mathbf{R}^{[n]}, \mathbf{r}_1, f, \theta) \\ E_z(\mathbf{R}^{[n]}, \mathbf{r}_2, f, \theta) \\ \vdots \\ E_z(\mathbf{R}^{[n]}, \mathbf{r}_P, f, \theta) \end{bmatrix} - \begin{bmatrix} E_z(\mathbf{R}^{[n_0]}, \mathbf{r}_1, f, \theta) \\ E_z(\mathbf{R}^{[n_0]}, \mathbf{r}_2, f, \theta) \\ \vdots \\ E_z(\mathbf{R}^{[n_0]}, \mathbf{r}_P, f, \theta) \end{bmatrix} \quad (7)$$

with  $\mathbf{R}^{[n]}$  and  $\mathbf{R}^{[n_0]}$  describing the wire distribution of the  $n$ th and  $n_0$ th RWDs, respectively. To generate our training data, the total field measurements due to the first RWD is taken as  $\mathbf{v}_{BG}$ , i.e.,  $n_0 = 1$ . This was proven to be effective because this subtraction yields a signal with an amplitude comparable to that of the scattered field signal, and it does not break the one-to-one correspondence between a given RWD and its associated field measurement. At frequencies as low as 10 MHz, the difference between the total field

signal and the incident field signal is negligible, but, at higher frequencies, say 50 MHz, since the scattered field signal amplitude becomes comparable to that of the incident field signal, such subtraction is not necessary. However, the converged accuracy will be unaffected by this subtraction because the aforementioned one-to-one correspondence is not broken by such subtraction.

#### IV. CNN PERFORMANCE ANALYSIS

Since the problem of predicting the number of wire(s) in a given sample is formulated as a classification problem, the CNN output should be mapped into a class. This is performed by simply mapping the maximum component of the normalized output vector into 1 and the rest of the components into 0. The resulting one-hot vector will then be compared to the desired output, which is also a one-hot vector. Based on this comparison, the prediction is labeled as either “accurate” or “inaccurate.”

The overall performance of CNN can be analyzed by evaluating the accuracy with which the CNN makes predictions on samples outside the training dataset. This can be done by dividing the number of correct predictions by the total number of predictions. For a testing dataset containing 300 000 samples containing up to  $N_{\max} = 10$  wires and generated at  $f = 10$  MHz at an angle of incidence of  $\theta = 45^\circ$ , an accuracy of 89% was achieved.

This performance analysis can be done in more detail by evaluating the accuracy within the subset of testing data, which contains only samples of a given number of wires. This is shown in Fig. 4(a), where the total number of samples and incorrect predictions is shown for each subset. In this way, the performance of the CNN is determined by the  $N_{\max}$  accuracies corresponding to the  $N_{\max}$  subsets.

Furthermore, this performance analysis can be done in more detail by looking at the predictions that were made by the CNN for samples in a subset containing  $N$  wires, where  $1 \leq N \leq N_{\max}$ . Averaging the output vectors returned for all samples within each subset yields  $N_{\max}$  numbers that describe the averaged probabilities that the samples within that subset are predicted to have 1, 2, 3, ...,  $N_{\max}$  wires. The matrix constructed from concatenating the resulting  $N_{\max}$  averaged probability vectors is shown in Fig. 4(b).

##### A. Improvement of Accuracy

The performance of the CNN can be improved by the following.

1) *Mapping Prediction Error Into Space and Detecting the “Blind Spots” of the CNN:* Another approach to evaluate the performance of the CNN described in Section III-B is to search for its “blind spots.” This can be achieved by looking at the subset of samples containing only one wire. This allows us to study the error calculated for the prediction made by the CNN as a function of wire location. The prediction error can be calculated via plugging the nonprojected output vector into the CCE function defined by (6). Consider a sample that actually contains one wire, with the output of the CNN for that sample being projected into the one-hot vector corresponding

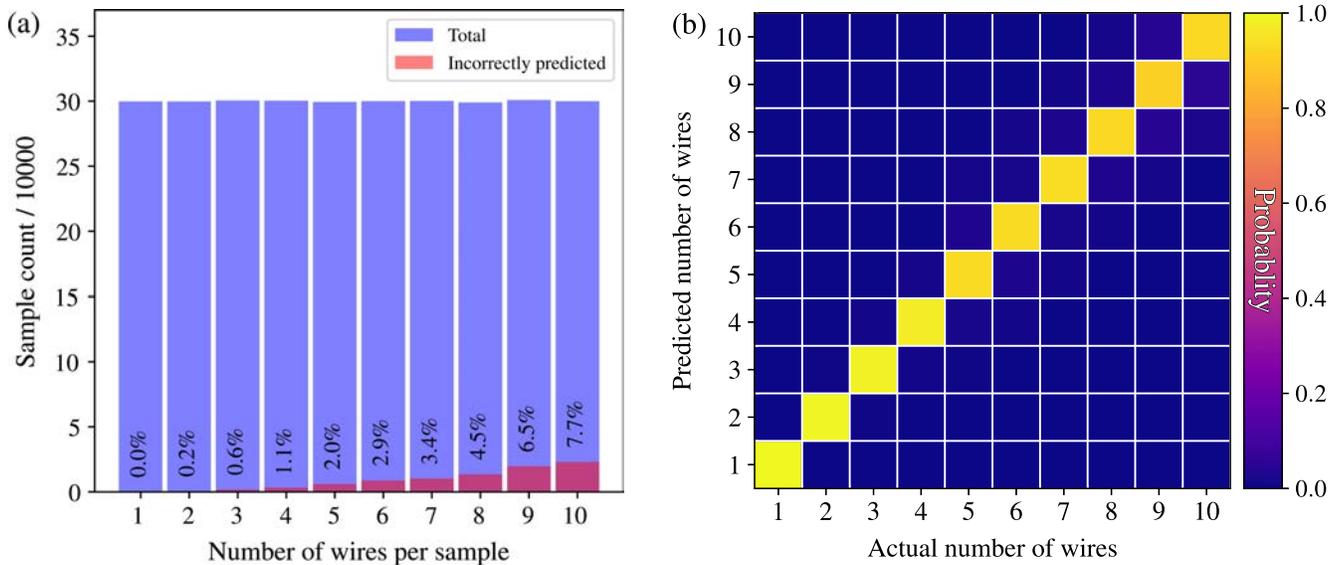


Fig. 8. Repeat of Fig. 4 for the case where each RWD is illuminated with a PW of  $f = 10$  MHz four separate times each time at a different angle of incidence. In this case, the CNN is fed with a four-channel input vector, each channel corresponding to  $\theta = 45^\circ, 135^\circ, 225^\circ,$  and  $315^\circ$ . Each channel is a twofold vector with each fold containing the real and imaginary parts of  $E_z$ . Therefore, in this case, the CNN is fed with an eightfold input vector containing the real and imaginary parts of  $E_z$  measured upon illumination of the wires with a PW at four different angles of incidence. Operating the trained CNN on a separate set of  $3 \times 10^6$  independent samples led to predictions with an overall accuracy of 97%.

to a single-wire sample, i.e.,  $[1, 0, 0, \dots, 0, 0]^T$ . Even if the prediction based on this projection is accurate, the prediction error obtained from the CCE function is still nonzero. These nonzero values for the prediction error provide insight into the blind spots of CNN. As implied in Section III-B, the prediction error for the  $n$ th sample that actually contains one wire is given by  $CCE_n = -\log_{10}[p_1^{[n]}]$  with  $p_1^{[n]}$  being the first component of the CNN output vector that gives the probability with which the  $n$ th sample is predicted to contain one wire.

As it can be seen in the four top panels of Fig. 5, illuminating the single-wire RWD with a PW only at one angle of incidence results in a higher prediction error if the wire is located at the corner of the box toward which the PW phase velocity is directed (see Fig. 1). This suggests that appending additional channels obtained from additional illumination angles to the input vector should reduce the prediction error for one-wire samples located in the formerly identified blind spot of the CNN, and this is confirmed by the bottom panel of Fig. 5.

The CNN that is to be fed with this eight-fold input vector has a similar topology to the one shown in Fig. 2 except that the input dimension of the first convolution layer, and of course, the input-output dimensions of the input layer, should be changed to (None, 40, 8). The CNN into which this eightfold input vector is fed contains 2838 trainable parameters (weights and biases). This points out one of the advantages of utilizing the convolutional layers because a 400% increase in the size of the input vector led to only 54 additional trainable parameters.

2) *Wavelength Variation*: Since the box side is chosen to be  $L = 100$  m long, the working wavelength of the dataset on which the CNN is trained is selected from the range

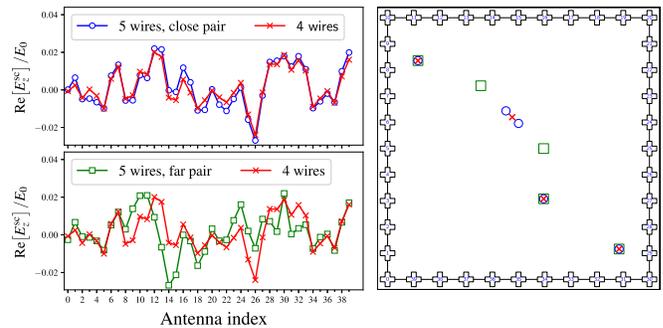


Fig. 9. Three RWDs with three wire locations in common. The wire locations of each RWD are marked with blue circles, red x's, and green squares on the right panel, and each of these three RWDs contains five, four, and five wires, respectively. Two wires in the RWD marked with circles are intentionally placed close to each other. Replacing these two wires with a single wire in their midpoint results in the second RWD marked with x's. Taking the two wires farther apart results in the third RWD marked with squares. The  $E_z^{sc}$  fields due to the illumination of these three RWDs with a PW incident at  $\theta = 45^\circ$  at  $f = 10$  MHz are graphed in the top-left and bottom-left panels. Although the location of three wires remains intact, as the remaining two wires get farther apart, replacing them with a single wire in their midpoint leads to field measurements with lesser similarity.

of 2–60 m, which extends over the HF and VHF frequency bands. For a given RWD being separately illuminated with PWs incident at  $\theta = 45^\circ, 135^\circ, 225^\circ,$  and  $315^\circ$ , four sets of 40-D input channels are generated all of which containing the  $E_z$  field computed at 40 sites at a given wavelength. This was repeated for all testing and training RWDs, and 30 different datasets were prepared by varying the illumination wavelength from 2 to 60 m in steps of 2 m. As shown in Fig. 6, such frequency analysis indicates that training the network on datasets containing the  $E_z$  field generated at larger wavelengths leads to higher accuracies.

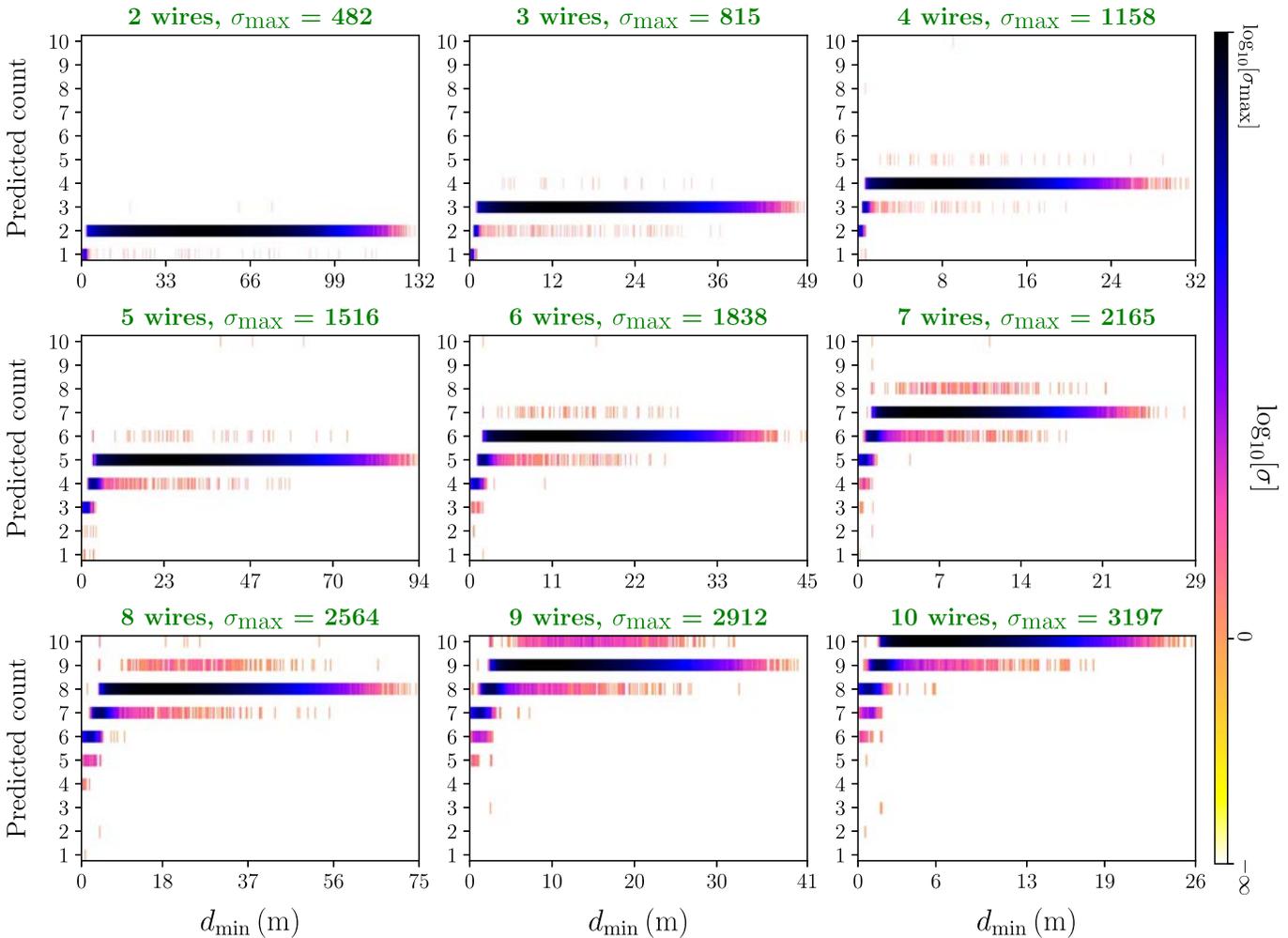


Fig. 10. CNN described in Section IV-A1 was tested on  $3 \times 10^6$  additional independent test samples containing up to  $N_{\max} = 10$  wires and generated at  $f = 10$  MHz. This test led to predictions with an overall accuracy of 97%. The density of predictions,  $\sigma$ , was then measured per predicted wire count for samples of given actual wire numbers. Within the subset of  $N$ -wire samples (where  $2 \leq N \leq 10$ ),  $\sigma$  is simply measured by counting the number of samples that are predicted to have a certain wire number whose  $d_{\min}$  value falls within  $[m\Delta, (m+1)\Delta)$ , where  $m \in \{0, 1, \dots, m_{\max}\}$ . For visualization purposes, a logarithmic scale has been utilized to color-map the measured  $\sigma$  values.

As the final test of the generalization capability of the CNN, we carried out tenfold cross validation. First,  $3 \times 10^6$  samples were randomly grouped into ten subsets, each containing  $3 \times 10^5$  samples. After training the CNN on the  $p$ th subset, it was tested on the  $q$ th subset, where  $q \neq p$  and  $1 \leq p, q \leq 10$ . No internal validation was performed during any of these training processes, and so the CNN was trained on  $3 \times 10^5$  samples. The 90 accuracy values obtained from testing the CNNs are shown as off-diagonal components of a  $10 \times 10$  matrix in Fig. 7, while the diagonal components show the ten training accuracy values. All the 100 accuracy values range from 96% to 97%, indicating that the CNN is capable of generalizing what it learns from any given dataset into any other given subset.

As suggested by the spatial and frequency analyses, the training data were generated by illuminating each RWD with a PW at  $f = 10$  MHz and separately at four different angles of incidence, namely,  $\theta = 45^\circ, 135^\circ, 225^\circ$ , and  $315^\circ$ . An evaluation of the CNN performance on  $3 \times 10^5$  additional independent test samples is shown in Fig. 8. A comparison

between Figs. 4 and 8 indicates the effectiveness of the analysis described in Section IV-A1. Moreover, Fig. 8(a) suggests that generating the training data with additional illumination angles not only improves the CNN performance for single-wire samples but also for samples with multiple wires.

## V. FAILURE ANALYSIS

### A. Small Interwire Separation

As shown in Fig. 9, in a sample with two wires close enough to each other, replacing the pair with a single wire makes negligible changes to the EM field measured at the 40 antennas. This suggests that the CNN would undercount the wires if two of them are too close to each other. To study the resolution capability of the CNN described in Section IV-A1, we measure the minimum interwire distance for samples that contain more than one wire, and we denote this quantity by  $d_{\min}$ . Then, within the subset of samples that contain only a given number of wires,  $N$ , the predictions to all possible wire counts are studied versus  $d_{\min}$ . More specifically,

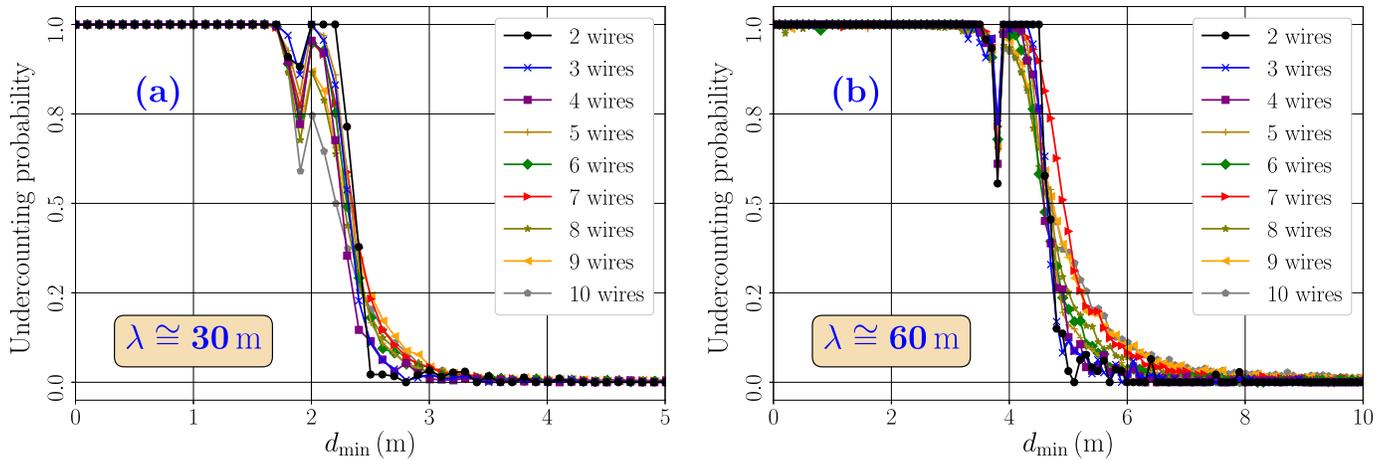


Fig. 11. (a) Undercounting probability curves calculated from the prediction density ( $\sigma$ ) values shown in Fig. 10 corresponding to nine different subsets each containing only samples with  $N$  wires, where  $2 \leq N \leq 10$ , generated at an illumination wavelength of  $\lambda \cong 30$  m. (b) Same curves generated at an illumination wavelength of  $\lambda \cong 60$  m whose corresponding nine-panel figure for prediction densities is not shown here.

this is done by counting the number of predictions to given wire count,  $C$ , whose  $d_{\min}$  value falls within  $[m\Delta, (m+1)\Delta)$  with  $m \in \{0, 1, \dots, m_{\max}\}$ , a quantity known as density of predictions and denoted by  $\sigma_{N,C}(m)$ . To measure  $\sigma_{N,C}(m)$  per subset of samples with  $N$  wires, per predicted count,  $C$ , the CNN whose training results are shown in Fig. 8 was tested on  $3 \times 10^6$  additional independent test samples, and the  $\sigma$  values measured for  $\Delta = 10$  cm are shown in Fig. 10. For samples containing  $N$  wires, where  $2 \leq N \leq N_{\max}$ , whose  $d_{\min}$  value falls within the range of  $[m\Delta, (m+1)\Delta)$ , the undercounting probability is defined as follows:

$$P_N(m) = \frac{\sum_{2 \leq C < N} \sigma_{N,C}(m)}{\sum_{2 \leq C \leq N_{\max}} \sigma_{N,C}(m)} \quad (8)$$

with  $m$  being any possible integer that covers the maximum measured  $d_{\min}$  within a given subset. The undercounting probability was computed from the prediction density values shown in Fig. 10 for nine different subsets each containing only samples with  $N$  wires, where  $2 \leq N \leq 10$  and shown in Fig. 11(a). As suggested by the step-like behavior shown in Fig. 11(a), the CNN makes erroneous predictions for samples containing a pair of wires whose distance falls under a certain threshold. That is, if two wires in a sample are separated by less than a certain threshold, the CNN would count those two wires as one. For samples with two pairs of wires with separation under a certain threshold, the CNN would undercount the number of wires twice and so on. This suggests the existence of a resolution limit for the CNN used to predict the number of wires in each RWD. As shown in Fig. 11(a), the resolution limit for the CNN is estimated to be 3 m, which is 10% of the illumination wavelength ( $\lambda \cong 30$  m). To understand the correlation between the illumination wavelength and the resolution limit of the CNN, we generated the undercounting probability curves by repeating the same process for an illumination wavelength of  $\lambda \cong 60$  m, and as it can be seen in Fig. 11(b), the resolution limit

of the CNN is estimated to be around 6 m, which again turns out to be around 10% of the illumination wavelength.

### B. Small Clearance From Antennas

On the other hand, as it can be seen in the bottom panel in Fig. 5, even after covering the blind spots of the CNN by additional illumination angles, the prediction error is larger for RWDs that contain a wire located too close to any of the antennas. This can be explained by the fact that the formalism provided by (12)–(14) generates diverging EM field values if the wire is too close to the measurement point, and the resulting 40-D vector of EM field measurements would contain a component much larger than the rest. As a result, the input vector resulting from such measurement becomes an outlier compared with the input vectors for which the CNN makes correct predictions. As mentioned in Section III-A, throughout this work, all the generated datasets contain RWDs with a minimum clearance of  $\delta = 0.02 L = 2$  m from the antennas. Increasing the minimum clearance does not improve the predictions; however, decreasing the minimum clearance from 2 m to 10 cm reduces the overall accuracy from 97% to 93%. This source of error is expected to be eliminated when using a formalism that produces correct EM field values at points in proximity to the wires. An alternative method to generate the training EM field data, which goes beyond the interacting dipole approximation, is utilizing numerical EMs packages, such as pyGDM [43], which could be considered for our future works.

## VI. CONCLUSION AND FUTURE WORKS

We studied solving an inverse problem in EMs, namely, counting the number of wires from the EM measurements acquired from antennas surrounding the wires, using NNs. Since fully connected NNs require too many weights to learn from the training data, they are incapable of generalization, and therefore, their use was ruled out. As a result, we made use of CNNs that inherently require much fewer trainable

parameters and were proven to have excellent generalization capability. This was achieved by moving a 1-D convolution window of length 3 over each of the channels of the input, in each of the three consecutive convolution layers.

We then studied the prediction error made by the CNN for single-wire samples and mapped the prediction error into space by analyzing the prediction error versus wire location. As a result of this analysis, the blind spots of the CNN were identified, and those blind spots were covered by additional channels of the input vector acquired by illuminating the RWDs with additional illumination angles. This remedy led to enhancing the CNN performance, which is suggested by the increase in accuracy from 88% to 97%.

At the end, the distribution of predictions was studied versus the minimum interwire separation within  $N_{\max} - 1$  subsets of  $N$ -wire samples, where  $2 \leq N \leq N_{\max}$ . This analysis indicated the existence of a resolution limit for the CNN described in Section IV-A1. That is, the CNN will predict the number of wires of a sample less than their actual number if at least one of the interwire separations measured for that sample falls under a certain threshold. This characteristic threshold was inferred from the step-like behavior of the undercounting probability and estimated to be around 10% of the illumination wavelength.

Our focus is to demonstrate the potential of the NNs to be utilized in counting identical objects and to introduce analytic methods that allow us to get estimates of the relevant parameters before moving to the laboratory stage. In addition, our analytic approach enables us to understand the capabilities of the network, such as resolution, and whether or not tuning the network topology can help in breaking the fundamental physics-related limits, such as the resolution limit.

Our future works include: 1) predicting the position of multiple identical targets as a classification or regression problem; 2) utilizing numerical packages, such as pyGDM [43], to generate the training data; 3) pulsed illumination of targets to include more illumination frequencies and applying convolution windows to the frequency dimension of the input; 4) the use of antenna transfer function [44]–[47] for more realistic modeling of antennas; and 5) introducing noise to the measured EM field to assure a more robust network performance.

## APPENDIX I

### FORMALISM TO GENERATE THE SCATTERED EM FIELD FOR A SYSTEM OF $N$ COUPLED WIRES

The approach discussed in [34] represents the  $i$ th object with a set of electric and magnetic dipoles denoted by  $\mathbf{d}_{2 \times 1}^{(i)} = [\mathbf{p}^{(i)}, \mathbf{m}^{(i)}]^T$ . The first step is to self-consistently determine the electric and magnetic dipole moments,  $\mathbf{d}_{2 \times 1}^{(i)} = [\epsilon_0 \mathbf{P}^{(i)} \cdot \mathbf{E}(\mathbf{r}_i), \mathbf{M}^{(i)} \cdot \mathbf{H}(\mathbf{r}_i)]^T$ , from the incident EM field at the location of the  $i$ th dipole,  $\mathbf{r}_i$ , using this matrix equation

$$\begin{bmatrix} \mathbf{d}_{2 \times 1}^{(1)} \\ \vdots \\ \mathbf{d}_{2 \times 1}^{(N)} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{2 \times 2} & \cdots & \mathbf{Q}_{2 \times 2}^{(1,N)} \\ \vdots & \ddots & \vdots \\ \mathbf{Q}_{2 \times 2}^{(N,1)} & \cdots & \mathbf{I}_{2 \times 2} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{F}_{2 \times 1}^{(1)} \\ \vdots \\ \mathbf{F}_{2 \times 1}^{(N)} \end{bmatrix} \quad (9)$$

where  $\mathbf{F}_{2 \times 1}^{(i)} = [\epsilon_0 \mathbf{P}^{(i)} \cdot \mathbf{E}^{(\text{inc})}(\mathbf{r}_i), \mathbf{M}^{(i)} \cdot \mathbf{H}^{(\text{inc})}(\mathbf{r}_i)]^T$ . Assuming that all the scatterers are z-directed thin wires of length  $l$  and radius  $a$ , the electric and magnetic polarizability tensors for all the wires,  $i = 1, 2, \dots, N$ , can be approximated with

$$\mathbf{P}^{(i)} \cong \frac{4}{3} \pi \left( \frac{l}{2} \right)^3 \left[ \ln \left( \frac{l}{a} \right) - 1 \right]^{-1} \hat{\mathbf{z}} \hat{\mathbf{z}} \quad \& \quad \mathbf{M}^{(i)} \cong \mathbf{0}. \quad (10)$$

Note that such approximation becomes less valid as the wire dimensions become comparable to the wavelength. For z-directed thin wires,  $\mathbf{Q}_{2 \times 2}^{(\alpha, \beta)}$  reduces to

$$\mathbf{Q}_{2 \times 2}^{(\alpha, \beta)} = \frac{l^3 e^{-jkR_{\alpha, \beta}}}{24 R_{\alpha, \beta} \left[ \ln \left( \frac{l}{a} \right) - 1 \right]} \begin{bmatrix} \frac{1}{R_{\alpha, \beta}^2} + \frac{jk}{R_{\alpha, \beta}} - k^2 & 0 \\ 0 & 0 \end{bmatrix} \quad (11)$$

where  $R_{\alpha, \beta} \equiv |\mathbf{r}_\alpha - \mathbf{r}_\beta|$  and  $k \equiv 2\pi(f/c)$ . Once the electric dipole moments,  $\{p_z^{(1)}, \dots, p_z^{(N)}\}$ , are obtained from (9), we can feed them into the following formalism to compute the scattered EM fields. The nonvanishing components of the scattered EM field are given by

$$E_z^{\text{sc}}(\mathbf{r}) = \sum_{\alpha=1}^N \frac{p_z^{(\alpha)} e^{-jkR_\alpha}}{4\pi \epsilon_0 R_\alpha} \left[ k^2 - \frac{1}{R_\alpha^2} - \frac{jk}{R_\alpha} \right] \quad (12)$$

$$H_x^{\text{sc}}(\mathbf{r}) = \frac{f}{2} \sum_{\alpha=1}^N p_z^{(\alpha)} e^{-jkR_\alpha} \frac{y - y_\alpha}{R_\alpha^2} \left[ \frac{j}{R_\alpha} - k \right] \quad (13)$$

$$H_y^{\text{sc}}(\mathbf{r}) = \frac{f}{2} \sum_{\alpha=1}^N p_z^{(\alpha)} e^{-jkR_\alpha} \frac{x - x_\alpha}{R_\alpha^2} \left[ \frac{j}{R_\alpha} + k \right] \quad (14)$$

where  $R_\alpha \equiv |\mathbf{r} - \mathbf{r}_\alpha|$ . It is worthwhile to mention that, as we get closer to the wires, i.e.,  $\mathbf{r} \rightarrow \mathbf{r}_n$  for  $n = 1, \dots, N$ , the formalism given by (12)–(14) becomes less accurate. Therefore, in addition to these dipole terms, higher order terms would be needed to correctly produce the scattered EM field in the vicinity of the wires. The thin wires are illuminated with a PW whose electric field vector is purely z-polarized, and the nonzero components of the incident EM field are given by

$$E_z^{\text{inc}}(\mathbf{r}) = E_0 e^{-jk[x \cos \theta + y \sin \theta]} \quad (15)$$

$$H_x^{\text{inc}}(\mathbf{r}) = -\frac{E_0}{Z_0} e^{-jk[x \cos \theta + y \sin \theta]} \sin \theta \quad (16)$$

$$H_y^{\text{inc}}(\mathbf{r}) = \frac{E_0}{Z_0} e^{-jk[x \cos \theta + y \sin \theta]} \cos \theta \quad (17)$$

with  $\theta$  and  $Z_0 = \sqrt{\mu_0/\epsilon_0} \cong 376.73 \Omega$ , respectively, being the PW angle of incidence and the impedance of free space. As a result, the total fields can be computed by adding the scattered fields obtained from (12)–(14) to the incident fields obtained from (15)–(17). The preceding equations assume an  $e^{j\omega t}$  time dependence.

## ACKNOWLEDGMENT

Mohsen Sabbaghi would like to thank Elizabeth K. Cole for providing detail on her codes for the end-to-end complex-valued CNNs. Also, he would like to thank Peter R. Wiecha for his feedback on using the pyGDM package in the HF and VHF frequency bands.

## REFERENCES

- [1] S. Arridge, P. Maass, O. Öktem, and C.-B. Schönlieb, "Solving inverse problems using data-driven models," *Acta Numerica*, vol. 28, pp. 1–174, May 2019.
- [2] Z. Q. Zhang, Q. H. Liu, C. Xiao, E. Ward, G. Ybarra, and W. T. Joines, "Microwave breast imaging: 3-D forward scattering simulation," *IEEE Trans. Biomed. Eng.*, vol. 50, no. 10, pp. 1180–1189, Oct. 2003.
- [3] B. D. Steinberg, "Microwave imaging of aircraft," *Proc. IEEE*, vol. 76, no. 12, pp. 1578–1592, Dec. 1988.
- [4] B. D. Steinberg, D. L. Carlson, and W. Lee, "Experimental localized radar cross sections of aircraft," *Proc. IEEE*, vol. 77, no. 5, pp. 663–669, May 1989.
- [5] E. Cole, J. Cheng, J. Pauly, and S. Vasanawala, "Analysis of deep complex-valued convolutional neural networks for MRI reconstruction and phase-focused applications," *Magn. Reson. Med.*, vol. 86, no. 2, pp. 1093–1109, Aug. 2021.
- [6] M. H. Tahersima *et al.*, "Deep neural network inverse design of integrated photonic power splitters," *Sci. Rep.*, vol. 9, no. 1, p. 1368, Feb. 2019.
- [7] Z. Liu, D. Zhu, S. P. Rodrigues, K.-T. Lee, and W. Cai, "Generative model for the inverse design of metasurfaces," *Nano Lett.*, vol. 18, no. 10, pp. 6570–6576, Oct. 2018.
- [8] J. Peurifoy *et al.*, "Nanophotonic particle simulation and inverse design using artificial neural networks," *Sci. Adv.*, vol. 4, no. 6, Jun. 2018, Art. no. eaar4206.
- [9] A. Sheverdin, F. Monticone, and C. Valagiannopoulos, "Photonic inverse design with neural networks: The case of invisibility in the visible," *Phys. Rev. A, Gen. Phys.*, vol. 14, no. 2, Aug. 2020, Art. no. 024054.
- [10] Z. A. Kudyshev, A. V. Kildishev, V. M. Shalaev, and A. Boltasseva, "Machine learning-assisted global optimization of photonic devices," *Nanophotonics*, vol. 10, no. 1, pp. 371–383, 2021.
- [11] J. Jiang and J. A. Fan, "Global optimization of dielectric metasurfaces using a physics-driven neural network," *Nano Lett.*, vol. 19, no. 8, pp. 5366–5372, Aug. 2019.
- [12] I. Sajedian, J. Kim, and J. Rho, "Finding the optical properties of plasmonic structures by image processing using a combination of convolutional neural networks and recurrent neural networks," *Microsystems Nanoengineering*, vol. 5, no. 1, p. 27, Jun. 2019.
- [13] Y. Fan and L. Ying, "Solving electrical impedance tomography with deep learning," *J. Comput. Phys.*, vol. 404, Mar. 2020, Art. no. 109119.
- [14] G. Beylkin, R. Coifman, and V. Rokhlin, "Fast wavelet transforms and numerical algorithms I," *Commun. Pure Appl. Math.*, vol. 44, no. 2, pp. 141–183, 1991.
- [15] Y. Fan, C. O. Bohorquez, and L. Ying, "BCR-Net: A neural network based on the nonstandard wavelet form," *J. Comput. Phys.*, vol. 384, pp. 1–15, May 2019.
- [16] Z. Wei and X. Chen, "Physics-inspired convolutional neural network for solving full-wave inverse scattering problems," *IEEE Trans. Antennas Propag.*, vol. 67, no. 9, pp. 6138–6148, Sep. 2019.
- [17] X. Chen, Z. Wei, M. Li, and P. Rocca, "A review of deep learning approaches for inversescattering problems," *Prog. Electromagn. Res.*, vol. 167, pp. 67–81, 2020.
- [18] L. Li, L. G. Wang, F. L. Teixeira, C. Liu, A. Nehorai, and T. J. Cui, "Deep neural network for nonlinear electromagnetic inverse scattering," *IEEE Trans. Antennas Propag.*, vol. 67, no. 3, pp. 1819–1825, Mar. 2019.
- [19] A. I. Sandhu, S. A. Shaikat, A. Desmal, and H. Bagci, "ANN-assisted CoSaMP algorithm for linear electromagnetic imaging of spatially sparse domains," *IEEE Trans. Antennas Propag.*, vol. 69, no. 9, pp. 6093–6098, Sep. 2021.
- [20] H. Zhou, T. Ouyang, Y. Li, J. Liu, and Q. Liu, "Linear-model-inspired neural network for electromagnetic inverse scattering," *IEEE Antennas Wireless Propag. Lett.*, vol. 19, no. 9, pp. 1536–1540, Sep. 2020.
- [21] R. D. Freeman, "Method of counting multiple targets in the post detection processing of a radar," U.S. Patent 4536764A, Aug. 20, 1985.
- [22] H. Ding *et al.*, "Counting human objects using backscattered radio frequency signals," *IEEE Trans. Mobile Comput.*, vol. 18, no. 5, pp. 1054–1067, May 2019.
- [23] M. Stephan and A. Santra, "Radar-based human target detection using deep residual U-Net for smart home applications," in *Proc. 18th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2019, pp. 175–182.
- [24] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds. Cham, Switzerland: Springer, 2015, pp. 234–241.
- [25] Y. Jia *et al.*, "ResNet-based counting algorithm for moving targets in through-the-wall radar," *IEEE Geosci. Remote Sens. Lett.*, vol. 18, no. 6, pp. 1–5, Jun. 2020.
- [26] P. R. Wiecha, A. Lecestre, N. Mallet, and G. Larrieu, "Pushing the limits of optical information storage using deep learning," *Nature Nanotechnol.*, vol. 14, no. 3, pp. 237–244, Mar. 2019.
- [27] A. Miroshnichenko, "Deep learning beats the optical diffraction limit," *Nature Nanotechnol.*, vol. 14, no. 3, pp. 198–199, Mar. 2019.
- [28] B. Orazbayev and R. Fleury, "Far-field subwavelength acoustic imaging by deep learning," *Phys. Rev. X*, vol. 10, no. 3, Aug. 2020, Art. no. 031029.
- [29] P. Ran, Y. Qin, D. Lesselier, and M. Serhir, "Subwavelength microstructure probing by Binary-specialized methods: Contrast source and convolutional neural networks," *IEEE Trans. Antennas Propag.*, vol. 69, no. 2, pp. 1030–1039, Feb. 2021.
- [30] P. Ran, S. Chen, M. Serhir, and D. Lesselier, "Imaging of sub-wavelength micro-structures by time reversal and neural networks, from synthetic to laboratory-controlled data," *IEEE Trans. Antennas Propag.*, early access, Jun. 2, 2021, doi: [10.1109/TAP.2021.3083741](https://doi.org/10.1109/TAP.2021.3083741).
- [31] A. Jain and B. Chandrasekaran, "39 dimensionality and sample size considerations in pattern recognition practice," in *Classification Pattern Recognition and Reduction of Dimensionality* (Handbook of Statistics), vol. 2. Amsterdam, The Netherlands: Elsevier, 1982, pp. 835–855.
- [32] S. J. Raudys and A. K. Jain, "Small sample size effects in statistical pattern recognition: Recommendations for practitioners," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 3, pp. 252–264, Mar. 1991.
- [33] C. Beleites, U. Neugebauer, T. Bocklitz, C. Krafft, and J. Popp, "Sample size planning for classification models," *Anal. Chim. Acta*, vol. 760, pp. 25–33, Jan. 2013.
- [34] G. W. Hanson and C. E. Baum, "Asymptotic analysis of the natural system modes of coupled bodies in the large-separation low-frequency regime," *IEEE Trans. Antennas Propag.*, vol. 47, no. 1, pp. 101–111, Jan. 1999.
- [35] T. Shumpert and D. Galloway, "Finite length cylindrical scatterer near perfectly conducting ground—A transmission line mode approximation," *IEEE Trans. Electromagn. Compat.*, vol. EMC-20, no. 1, pp. 145–151, Feb. 1978.
- [36] L. S. Riggs and T. H. Shumpert, "Trajectories of the singularities of a thin wire scatterer parallel to lossy ground," *IEEE Trans. Antennas Propag.*, vol. AP-27, no. 6, pp. 864–868, Nov. 1979.
- [37] D. B. Judd, "Maxwell and mordem colorimetry," *J. Photographic Sci.*, vol. 9, no. 6, pp. 341–352, Nov. 1961.
- [38] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016. [Online]. Available: <http://www.deeplearningbook.org>
- [39] C. Trabelsi *et al.*, "Deep complex networks," 2017, *arXiv:1705.09792*. [Online]. Available: <https://arxiv.org/abs/1705.09792>
- [40] J. S. Dramsch and Contributors, "Complex-valued neural networks in keras with tensorflow," Figshare, Cambridge, MA, USA, Tech. Rep., 2019, doi: [10.6084/m9.figshare.9783773.v1](https://doi.org/10.6084/m9.figshare.9783773.v1).
- [41] L. Smith *et al.*, "Classifying WiFi 'physical fingerprints' using complex deep learning," *Proc. SPIE*, vol. 11394, pp. 82–96, Apr. 2020.
- [42] S. S. Haykin, *Neural Networks and Learning Machines*, 3rd ed. Upper Saddle River, NJ, USA: Pearson, 2009.
- [43] P. R. Wiecha, "PyGDM—A Python toolkit for full-field electro-dynamical simulations and evolutionary optimization of nanostructures," *Comput. Phys. Commun.*, vol. 233, pp. 167–192, Dec. 2018.
- [44] B. R. Mayo, P. W. Howells, and W. B. Adams, "Generalized linear radar analysis," *Microw. J.*, vol. 4, no. 8, pp. 79–84, 1961.
- [45] M. Kanda, "Transients in a resistively loaded linear antenna compared with those in a conical antenna and a TEM horn," *IEEE Trans. Antennas Propag.*, vol. AP-28, no. 1, pp. 132–136, Jan. 1980.
- [46] X. Qing, Z. N. Chen, and M. Y. W. Chia, "Characterization of ultrawideband antennas using transfer functions," *Radio Sci.*, vol. 41, no. 1, pp. 1–10, Feb. 2006.
- [47] M. Nel, J. Joubert, and J. W. Odendaal, "The measurement of complex antenna transfer functions for ultra-wideband antennas in a compact range [measurements corner]," *IEEE Antennas Propag. Mag.*, vol. 56, no. 6, pp. 163–170, Dec. 2014.



**Mohsen Sabbaghi** (Graduate Student Member, IEEE) was born in Qom, Iran, in 1985. He received the B.Sc. degree in physics from Isfahan University of Technology (IUT), Isfahan, Iran, in 2008, and the M.Sc. degree in physics from Pohang University of Science and Technology (POSTECH), Pohang, South Korea, in 2014. He is currently pursuing the Ph.D. degree with the Department of Electrical Engineering and Computer Science, University of Wisconsin–Milwaukee (UWM), Wisconsin, WI, USA. In his master's thesis published in 2014,

he independently predicted and proposed the dc-current-induced nonreciprocity in terahertz signal propagation in graphene.

He currently serves as a Research and Teaching Assistant with the University of Wisconsin–Milwaukee. His research interests include RF tomography, inverse scattering, antennas, machine learning, and the electromagnetic response of 2-D materials.



**Jun Zhang** (Senior Member, IEEE) received the M.S. and Ph.D. degrees in electrical engineering from Rensselaer Polytechnic Institute, Troy, NY, USA, in 1985 and 1988, respectively.

He joined the faculty of the Department of Electrical Engineering and Computer Science, University of Wisconsin–Milwaukee, Wisconsin, WI, USA, where he is currently a Professor. His research interests include image processing, signal processing, and machine learning.

Dr. Zhang has been an Associate Editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING, and his research has received funding from NSF, ONR, the State of Wisconsin, and industry.



**George W. Hanson** (Fellow, IEEE) was born in Glen Ridge, NJ, USA, in 1963. He received the B.S.E.E. degree from Lehigh University, Bethlehem, PA, USA, in 1986, the M.S.E.E. degree from Southern Methodist University, Dallas, TX, USA, in 1988, and the Ph.D. degree from Michigan State University, East Lansing, MI, USA, in 1991.

From 1986 to 1988, he was a Development Engineer with General Dynamics, Fort Worth, TX, USA, where he worked on radar simulators. From 1988 to 1991, he was a Research and Teaching Assistant with the Department of Electrical Engineering, Michigan State University. He is currently a Professor of electrical engineering and computer science at the University of Wisconsin–Milwaukee, Wisconsin, WI, USA. His research interests include nanoelectromagnetics, plasmonics, quantum optics, and mathematical methods in electromagnetics.

Dr. Hanson received the S.A. Schelkunoff Best Paper Award from the IEEE Antennas and Propagation Society in 2006. He is a member of the URSI Commission B, Sigma Xi, and Eta Kappa Nu, and was an Associate Editor of the IEEE TRANSACTIONS ON ANTENNAS AND PROPAGATION from 2002 to 2007. He has coauthored the book *Operator Theory for Electromagnetics: An Introduction* (Springer, New York, 2002) and authored *Fundamentals of Nanoelectronics* (Prentice-Hall, New Jersey, 2007).