

# Physics-informed Deep Learning to Solve 2D Electromagnetic Scattering Problems

Ji-Yuan Wang<sup>1</sup> and Xiao-Min Pan<sup>2\*</sup>

<sup>1</sup>School of Integrated Circuit and Electronics  
Beijing Institute of Technology, Beijing, 100081, China  
3120210673@bit.edu.cn

<sup>2</sup>School of Cyberspace Science and Technology  
Beijing Institute of Technology, Beijing, 100081, China  
xmpan@bit.edu.cn

\*Corresponding Author

**Abstract** – The utilization of physics-informed deep learning (PI-DL) methodologies provides an approach to augment the predictive capabilities of deep learning (DL) models by constraining them with known physical principles. We utilize a PI-DL model called the deep operator network (DeepONet) to solve two-dimensional (2D) electromagnetic (EM) scattering problems. Numerical results demonstrate that the discrepancy between the DeepONet and conventional method of moments (MoM) is small, while maintaining computational efficiency.

**Index Terms** – Electromagnetic scattering, physics-informed deep learning, the method of moments.

## I. INTRODUCTION

Machine learning (ML) has emerged as a promising alternative for solving electromagnetic (EM) scattering problems [1–4] through an offline training - online prediction pattern. In the current landscape, two primary categories of ML approaches have been identified for EM scattering problems. The first approach treats ML as a “black box” and employs it as an alternative to traditional solvers [5–7]. Alternatively, the second approach involves replacing a module of a traditional solver with an ML-based solution [8–10]. Among these methodologies, physics-informed ML (PI-ML) [11] has attracted significant attention since it can gracefully incorporate both empirical data and prior physical knowledge. This integration of physical laws and data-driven approaches has been shown to improve the accuracy, reliability, and interpretability of predictions in various applications related to physical mechanisms. In [12], the physics-informed neural network (PINN) [13] was developed to solve time domain EM problems, where initial conditions, boundary conditions, as well as Maxwell’s equations, were encoded as the constraints during the training process of the network.

Among many deep learning (DL) networks, the deep operator network (DeepONet) has the attractive ability to break the “curse of dimensionality” [14]. For the first time, to the best of our knowledge, this work utilizes the DeepONet to solve dynamic EM scattering problems. The performance of the DeepONet is investigated in terms of different configurations.

The rest of the paper is as follows. In Section II, we review the process of EM scattering, using a two-dimensional (2D) dielectric scatterer as an illustrative example. In Section III, we show how to use the DeepONet to solve EM scattering problems. In Section IV, several numerical results are given to verify the good performance of the DeepONet. Conclusions are given in Section V.

## II. PROBLEM STATEMENT

In this section, we consider a 2D nonmagnetic transverse magnetic (TM) case, as shown in Fig. 1. The unknown scatterers are positioned in a domain  $D \in \mathbb{R}^2$  with homogeneous background, i.e., the free space. Harmonic incident waves are excited by the transmitter with the time-factor being  $e^{j\omega t}$ . The scattered fields can be measured by the receivers in the domain  $S \in \mathbb{R}^2$ .

The EM scattering problem aims to determine the scattered fields generated by the illumination of incident fields on scatterers. The method of moments (MoM) [1, 4] is widely adopted to solve this problem. After discretizing the computational domain  $D$  into  $M = M_1 \times M_2$  sub-domains, we can obtain the matrix system

$$\bar{\mathbf{J}} = \bar{\chi} \cdot (\bar{\mathbf{E}}^i + \bar{\mathbf{G}}_D \cdot \bar{\mathbf{J}}), \quad (1)$$

where  $\bar{\mathbf{J}}$  and  $\bar{\mathbf{E}}^i$  are vectors of the equivalent electric currents and incident fields,  $\bar{\mathbf{G}}_D$  is the matrix of the free space Green’s function in the domain  $D$ , and  $\bar{\chi}$  is the diagonal matrix storing the contrast of each sub-domain.

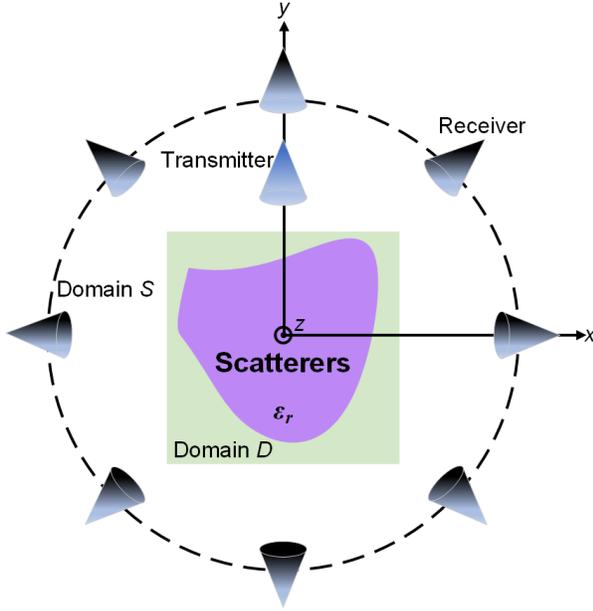


Fig. 1. Setup of EM scattering problems in the 2D case.

After  $\bar{\mathbf{J}}$  is obtained by solving Eq. (1), the discrete total fields  $\bar{\mathbf{E}}^t$  in the domain  $D$  can be obtained:

$$\bar{\mathbf{E}}^t = \bar{\mathbf{E}}^i + \bar{\bar{\mathbf{G}}}_D \cdot \bar{\mathbf{J}}. \tag{2}$$

Similarly, the scattered fields of any observation region can be calculated:

$$\bar{\mathbf{E}}^s = \bar{\bar{\mathbf{G}}}_S \cdot \bar{\mathbf{J}}, \tag{3}$$

where  $\bar{\bar{\mathbf{G}}}_S$  is the matrix of the free space Green’s function in the domain  $S$ .

### III. PHYSICS-INFORMED DEEP LEARNING

The DeepONet [15] is a physics-informed DL (PIDL) model, learning a variety of explicit operators that map from one function space to another, which is based on the universal approximation theorem for operators [16]. The theorem portrays a structure that consists of two sub-networks: one is the branch network for encoding the discrete input function space while the other is the trunk network for encoding the output function space, and shows that this particular network structure is capable of approximating a class of physical operators with arbitrary accuracy, which suggests its significant implications for the accurate modeling of complex physical systems. Essentially, the underlying mathematical foundation distinguishes the DeepONet from other neural networks.

The overall flow of employing the DeepONet to solve EM scattering problems is shown in Fig. 2. Firstly, we approximate the unknown current generation opera-

tor, i.e., the map from  $\chi(\mathbf{r})$  to  $\mathbf{J}(\mathbf{r})$ , utilizing the DeepONet. Then the total and scattered fields are computed through a simple matrix-vector multiplication process using the predicted results acquired by the DeepONet. Specifically, the branch network of the DeepONet is a complex-valued (CV) ResNet [17] consisting of a convolutional input layer, followed by four residual blocks, and a fully connected output layer to generate modules of the real and imaginary parts. The setup of the CV network is similar to that in [18]. Here, we are dealing with images, or data matrices, of size  $64 \times 64$ . In the forward calculation, the number of channels is first changed to 64 through the input layer while the size of each channel is halved. The four residual blocks are then connected in turn. Assume that the input of one residual block has  $N_{in}$  channels and the size of each channel is  $M_3 \times M_3$ . After the forward calculation of this block, the number of channels becomes  $2N_{in}$ , and the size of each channel is halved to  $\frac{M_3}{2} \times \frac{M_3}{2}$ . After that, we flatten the  $1 \times 1$  matrix over 1024 channels and connect a fully connected layer to output the result. The trunk networks are two real-valued fully connected networks to encode respectively the real and imaginary parts of the equivalent electric current function. They both have six hidden layers.

The input of the branch network is  $\bar{\chi} \oplus \bar{\chi} \circ \bar{\mathbf{E}}^i$ , where  $\oplus$  means to put the matrices on both sides of the symbol into the two channels of the tensor, and  $\circ$  denotes the Hadamard product. The inputs to both trunk networks are the coordinates of one sub-domain, i.e.  $(x, y)$ , for the 2D problem in this paper.

The output of the DeepONet is the real and imaginary part of the equivalent electric current within the sub-domain based on the input coordinates of the trunk networks, derived by computing the dot product between the real and imaginary components of the branch network output with the corresponding elements of the two trunk network outputs.

Trainable biases  $\theta_{bias}$  are added to the final output of the DeepONet to improve generalization performance. The number and size of channels or the number of neurons is indicated below each layer.

According to the universal approximation theorem, the output layer of the branch network does not need to employ an activation function, while the output layer of the trunk network requires the use of it. Apparently, the background region with a contrast of zero does not contribute to equivalent electric currents. We incorporate this prior knowledge into the training process of the DeepONet, i.e., we only need to predict equivalent electric currents on sub-domains with non-zero contrasts.

The loss function of the DeepONet can be expressed in terms of the mean absolute error as

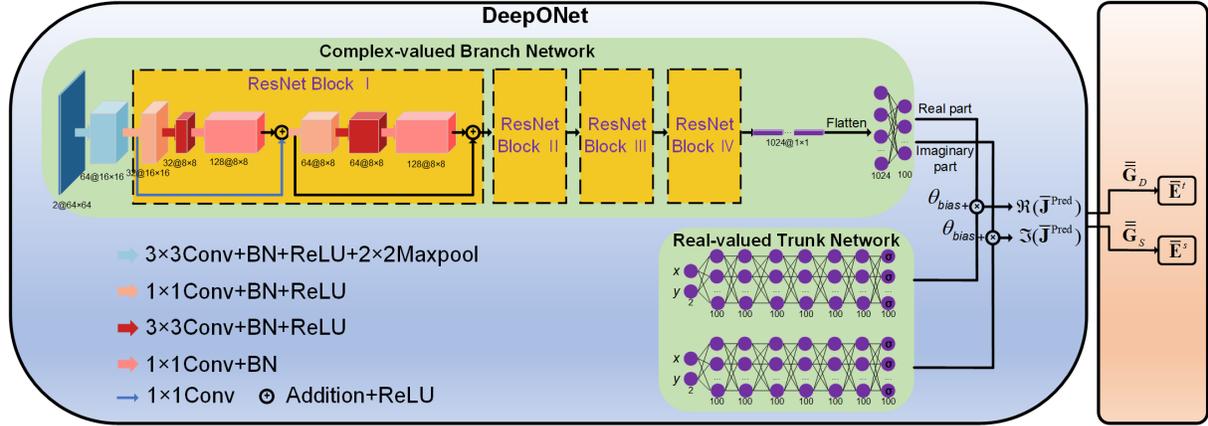


Fig. 2. The overall flow of employing the DeepONet to solve EM scattering problems.

$$\mathcal{L} = \frac{1}{B} \sum_{b=1}^B \left( \left| \frac{\Re(\bar{\mathbf{J}}_b^{\text{Pred}}) - \Re(\bar{\mathbf{J}}_b^{\text{MoM}})}{N_b} \right| + \left| \frac{\Im(\bar{\mathbf{J}}_b^{\text{Pred}}) - \Im(\bar{\mathbf{J}}_b^{\text{MoM}})}{N_b} \right| \right), \quad (4)$$

where  $B$  is the batch size,  $N_b$  is the number of non-zero contrast sub-domains for the  $b$ th scatterer,  $\bar{\mathbf{J}}_b^{\text{Pred}}$  is equivalent electric currents for the  $b$ th scatterer obtained from the DeepONet while  $\bar{\mathbf{J}}_b^{\text{MoM}}$  is equivalent electric currents calculated by the MoM, and  $\Re(\cdot)$  and  $\Im(\cdot)$  denote taking the real and imaginary part respectively.

#### IV. NUMERICAL RESULTS

In this section, we use several numerical examples to verify our approach. The DeepONet in this paper is built by Pytorch [19] and written in Python. We use the MoM to generate labels of training, validation and testing sets, i.e., equivalent electric currents. Specifically, the generalized minimal residual (GMRES) algorithm [20] is employed, with the aid of the fast Fourier transform (FFT) acceleration. The performance of the DeepONet is investigated on a server with two Intel Xeon CPUs and NVIDIA RTX 3090 GPU.

The data employed in our work is obtained by the MoM. In particular, 2D dielectric scatterers are generated according to handwritten numbers from the MNIST [21], which range from 0 to 9. Each scatterer has the computational domain  $D$  with a dimension of  $2 \times 2 \text{ m}^2$ , centering at  $(0, 0)$ . The real and imaginary parts of the relative permittivity of the training samples are randomly selected from 1.10-2.00 and 0.00-1.00, respectively. The number of scatterers in the training, validation and testing set are respectively, 20,000, 500, and 1000. We limit our analysis to the scenario of a single incident angle of  $90^\circ$ . The incident field operates at a frequency of 300 MHz. The scattered fields are sampled at

360 points that are uniformly located on a circle with a radius of 5 m. Figure 3 depicts some inputs of the branch network as well as their corresponding labels.

In the training stage, the adaptive moment estimation method (Adam) [22] is employed to minimize the loss function shown in Eq. (4). The learning rates of all sub-networks are set to 0.0002 at the beginning and are halved sequentially at the 100th, 140th, and 170th iterations. Each training batch contains 40 samples. The total number of iterations is 201. In the testing stage, to quantify the difference between the DeepONet predictions and true values generated by the MoM, we define the relative error ( $\text{RE}_q$ ) for the  $q$ th obstacle as follows:

$$\text{RE}_q = \frac{\|\bar{\mathbf{J}}_q^{\text{Pred}} - \bar{\mathbf{J}}_q^{\text{MoM}}\|_2}{\|\bar{\mathbf{J}}_q^{\text{MoM}}\|_2}. \quad (5)$$

For the entire testing set, we define the mean relative error (MRE)

$$\text{MRE} = \frac{1}{Q} \sum_{q=1}^Q \text{RE}_q, \quad (6)$$

where  $Q$  is the number of samples in the testing set.

#### A. Feasibility validation

The testing set is set up in the same manner as the training set. Trajectories of losses are shown in Fig. 4 (a) and the histogram of REs for the testing set is shown in Fig. 4 (b). The MRE is found to be 0.043 while the variance of REs is estimated to be  $4.9\text{e-}04$ . The predicted results are illustrated in Figs. 5 (a) and 5 (b), comprising a test instance from the MNIST test set and another from a distinct graphic type, respectively, to exhibit the feasibility of the DeepONet. It can be found that the predicted values match well with the true ones.

#### B. Generalization testing

In the context of the DeepONet framework, the size of the input function space and the size of the output

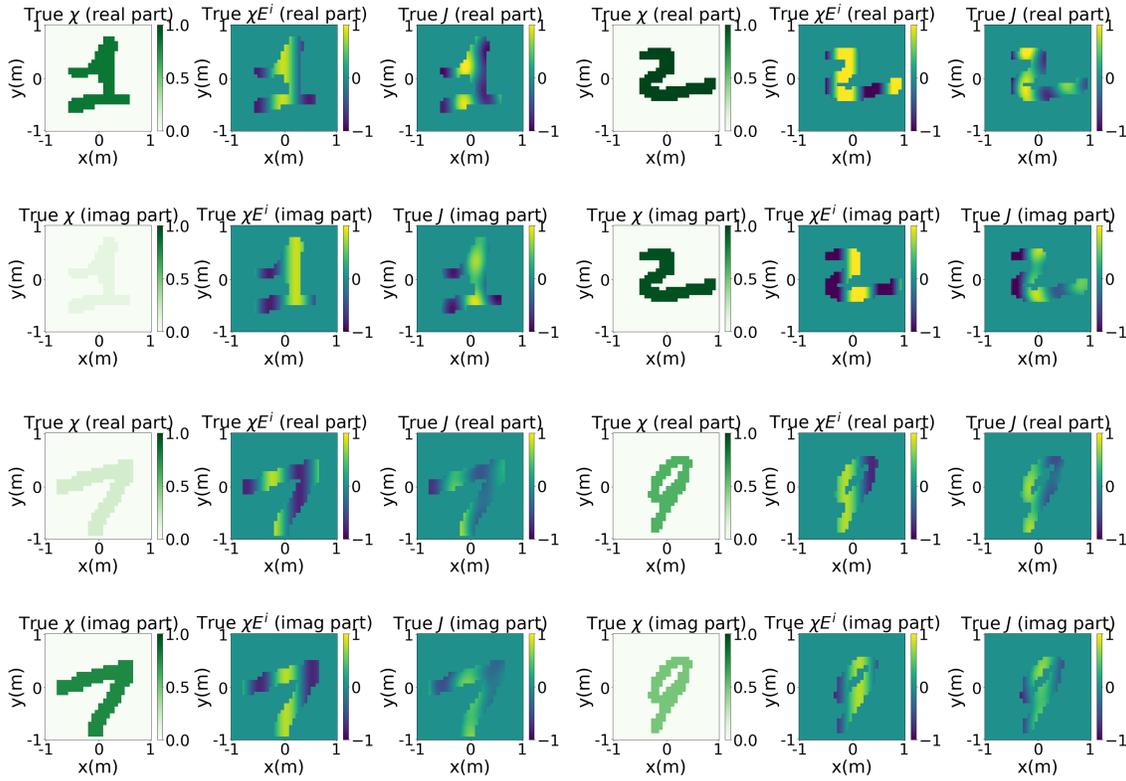


Fig. 3. Examples of the branch network input  $\bar{\chi} \oplus \bar{\chi} \circ \bar{\mathbf{E}}^i$  and corresponding equivalent electric current labels.

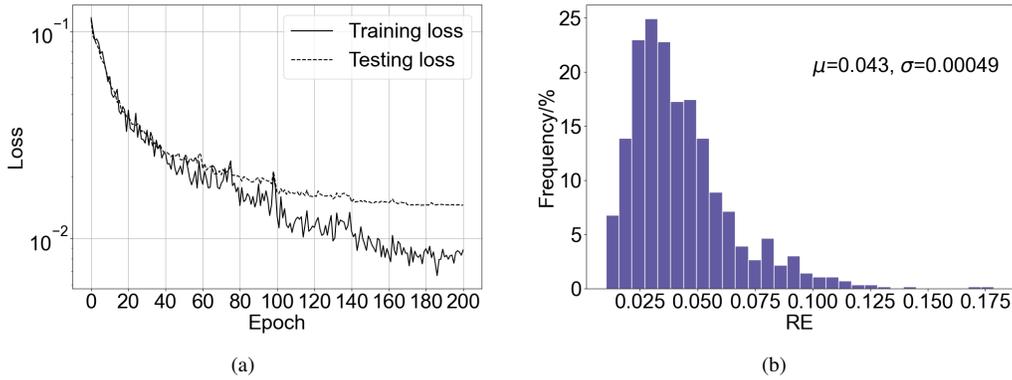


Fig. 4. (a) Trajectories of losses as a function of the number of epochs. (b) The RE histogram of the testing set with 1000 samples.

function space after discretization are found to be independent. While fixing the input size of the branch network, we can flexibly adjust the output of the DeepONet by varying the coordinate input of the trunk network during the testing stage. In this part, we explore the potential of predicting equivalent electric currents with grids of  $32 \times 32$  and  $96 \times 96$  using the DeepONet trained

on the training set of grid size  $64 \times 64$ , with the position and dimension of the domain  $D$  being held constant. For each of the two aforementioned cases, we generate 1,000 scatterers for the testing set. The configuration of the scatterers are set similarly to the previous example. Figure 6 (a) and 6 (b) give histograms of REs for the two cases. It is indicated that the DeepONet successfully

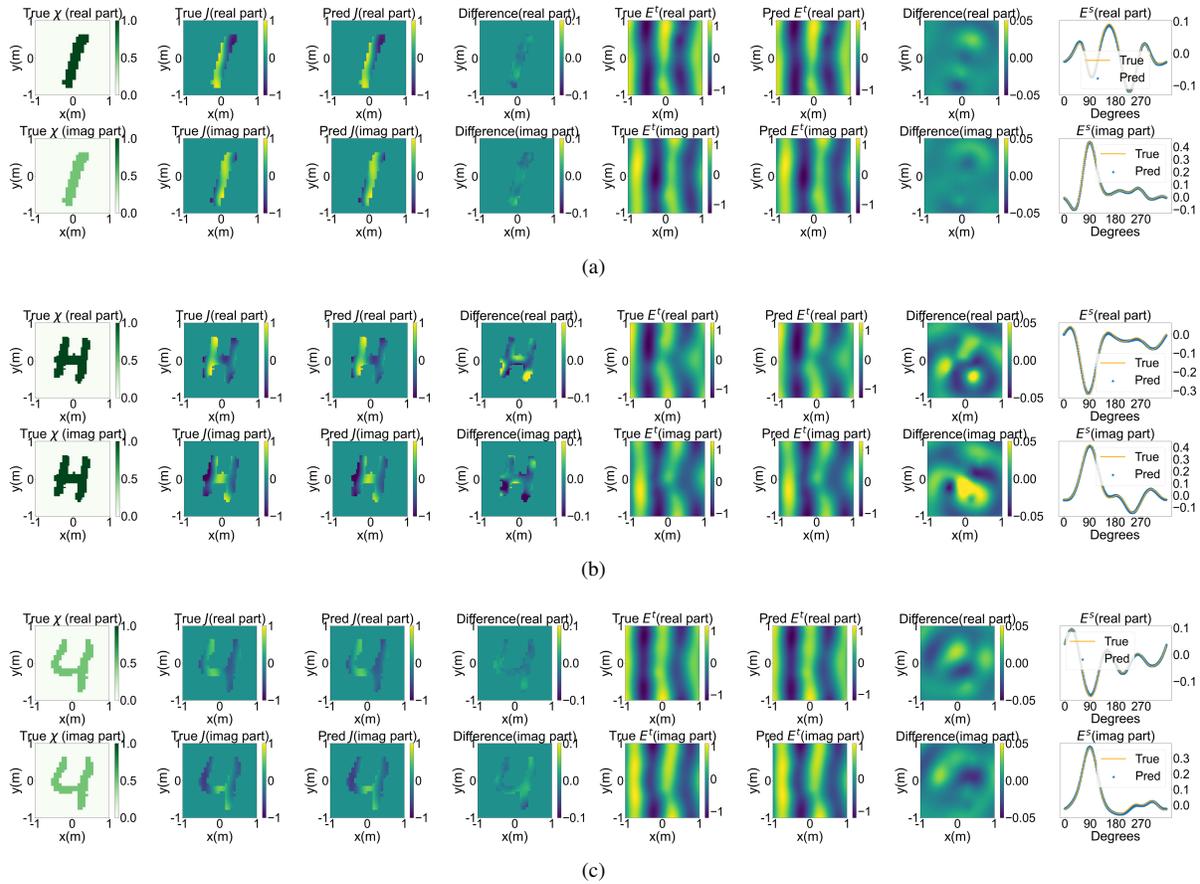


Fig. 5. The predicted results. The 1st column is the real and imaginary parts of contrasts. The 2nd column gives the real and imaginary parts of true equivalent electric currents computed by the MoM. The 3rd column is the real and imaginary parts of predicted equivalent electric currents. The 4th and 7th columns show the differences between true and predicted results. The 5th and 6th columns are the total fields calculated with the true and predicted equivalent electric currents, respectively. The last column are the true and predicted scattered fields in  $\phi \in [0^\circ, 359^\circ]$ . (a) A sample from the MNIST testing set with a grid size of  $64 \times 64$ , (b) a sample in a different graphic type with a grid size of  $64 \times 64$ , (c) a sample from the MNIST testing set with a grid size of  $1024 \times 1024$ .

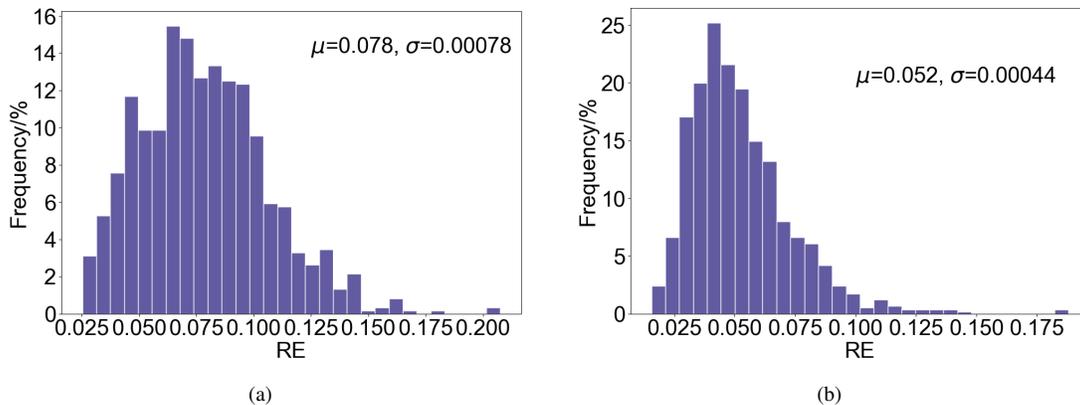


Fig. 6. The RE histograms of the testing sets with different grids of the equivalent electric currents. Each testing set contains 1000 samples. (a)  $32 \times 32$  case. (b)  $96 \times 96$  case.

predicted multi-grid results, confirming its nice generalization capability as a grid-free EM numerical solver.

### C. Computational time

Although the DeepONet is a picture-to-pixel model that yields the equivalent electric current on a single sub-domain per forward calculation, we can supply all the coordinate data concurrently and utilize the graphics processing unit (GPU) to expedite computations, thus significantly reducing the computation time. Additionally, the prediction of equivalent electric currents on sub-domains with non-zero contrasts leads to further reduction of total prediction time. Notably, as the domain  $D$  is divided into finer partitions, the traditional numerical solver suffers from the “curse of dimensionality,” resulting in an exponential increase in computation time. Conversely, the DeepONet’s computation time remains relatively unchanged. The reason is that the time increase in the trunk network and dot product during the forward calculation due to the increase in coordinate inputs is negligible, while the cost of the branch network remains constant.

Table 1: Comparison of computation time between the DeepONet and MoM

Grid Size	DeepONet (s)	MoM (s)
$64 \times 64$	0.018	0.014
$256 \times 256$	0.018	0.38
$1024 \times 1024$	0.024	6.39

Table 1 compares the DeepONet and MoM on different grids in terms of the computational time to obtain equivalent electric currents. The test obstacle is configured in an identical manner as in previous experiments. Using an example of grid size  $1024 \times 1024$ , the DeepONet predicts equivalent electric currents in just 0.024 s, compared to 6.39 s for the MoM to complete the corresponding calculation. The predicted results of grid size  $1024 \times 1024$  are shown in Fig. 5 (c). In this case, both the total and scattered fields calculated with the true equivalent electric currents and the predicted ones match well.

## V. CONCLUSION

Based on PI-DL, this paper utilizes the DeepONet to solve 2D dynamic EM scattering problems from the perspective of approximating the unknown current generation operator. The underlying mathematical foundation distinguishes the DeepONet from other neural networks. The accuracy, efficiency, and generalization of the DeepONet are verified through numerical examples. Furthermore, the DeepONet exhibits the ability to perform multi-grid prediction and overcome the “curse of

dimensionality.” Since the DeepONet can be extended to three-dimensional cases, the proposed approach has the potential to solve 3D dynamic EM scattering problems.

## ACKNOWLEDGMENT

This work was partly supported by NSFC under grant 62171033.

## REFERENCES

- [1] W. C. Chew, M. S. Tong, and B. Hu, “Integral equation methods for electromagnetic and elastic waves,” *Synthesis Lectures on Computational Electromagnetics*, vol. 3, no. 1, pp. 1–241, 2008.
- [2] D.-M. Yu, X. Ye, X.-M. Pan, and X.-Q. Sheng, “Fourier bases-expansion for three-dimensional electromagnetic inverse scattering problems,” *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022.
- [3] X. Chen, *Computational Methods for Electromagnetic Inverse Scattering*, John Wiley & Sons, 2018.
- [4] Y.-N. Liu and X.-M. Pan, “Solution of volume-surface integral equation accelerated by MLFMA and skeletonization,” *IEEE Transactions on Antennas and Propagation*, vol. 70, no. 7, pp. 6078–6083, 2022.
- [5] 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 Transactions on Antennas and Propagation*, vol. 67, no. 3, pp. 1819–1825, 2018.
- [6] S. Qi, Y. Wang, Y. Li, X. Wu, Q. Ren, and Y. Ren, “Two-dimensional electromagnetic solver based on deep learning technique,” *IEEE Journal on Multiscale and Multiphysics Computational Techniques*, vol. 5, pp. 83–88, 2020.
- [7] Z. Ma, K. Xu, R. Song, C.-F. Wang, and X. Chen, “Learning-based fast electromagnetic scattering solver through generative adversarial network,” *IEEE Transactions on Antennas and Propagation*, vol. 69, no. 4, pp. 2194–2208, 2020.
- [8] R. Guo, T. Shan, X. Song, M. Li, F. Yang, S. Xu, and A. Abubakar, “Physics embedded deep neural network for solving volume integral equation: 2-D case,” *IEEE Transactions on Antennas and Propagation*, vol. 70, no. 8, pp. 6135–6147, 2021.
- [9] Y. Hu, Y. Jin, X. Wu, and J. Chen, “A theory-guided deep neural network for time domain electromagnetic simulation and inversion using a differentiable programming platform,” *IEEE Transactions on Antennas and Propagation*, vol. 70, no. 1, pp. 767–772, 2021.
- [10] B.-W. Xue, D. Wu, B.-Y. Song, R. Guo, X.-M. Pan, M.-K. Li, and X.-Q. Sheng, “U-net conjugate gradient solution of electromagnetic

- scattering from dielectric objects,” in *2021 International Applied Computational Electromagnetics Society (ACES-China) Symposium*, pp. 1–2, IEEE, 2021.
- [11] Z. Hao, S. Liu, Y. Zhang, C. Ying, Y. Feng, H. Su, and J. Zhu, “Physics-informed machine learning: a survey on problems, methods and applications,” *arXiv preprint arXiv:2211.08064*, 2023.
- [12] P. Zhang, Y. Hu, Y. Jin, S. Deng, X. Wu, and J. Chen, “A Maxwell’s equations based deep learning method for time domain electromagnetic simulations,” *IEEE Journal on Multiscale and Multiphysics Computational Techniques*, vol. 6, pp. 35–40, 2021.
- [13] L. Lu, X. Meng, Z. Mao, and G. E. Karniadakis, “DeepXDE: A deep learning library for solving differential equations,” *SIAM Review*, vol. 63, no. 1, pp. 208–228, 2021.
- [14] S. Lanthaler, S. Mishra, and G. E. Karniadakis, “Error estimates for deepONets: A deep learning framework in infinite dimensions,” *Transactions of Mathematics and Its Applications*, vol. 6, no. 1, p. tnac001, 2022.
- [15] L. Lu, P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis, “Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators,” *Nature Machine Intelligence*, vol. 3, no. 3, pp. 218–229, 2021.
- [16] T. Chen and H. Chen, “Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems,” *IEEE Transactions on Neural Networks*, vol. 6, no. 4, pp. 911–917, 1995.
- [17] C. Trabelsi, O. Bilaniuk, Y. Zhang, D. Serdyuk, S. Subramanian, J. F. Santos, S. Mehri, N. Rostamzadeh, Y. Bengio, and C. J. Pal, “Deep complex networks,” *arXiv preprint arXiv:1705.09792*, 2017.
- [18] X.-M. Pan, B.-Y. Song, D. Wu, G. Wei, and X.-Q. Sheng, “On phase information for deep neural networks to solve full-wave nonlinear inverse scattering problems,” *IEEE Antennas and Wireless Propagation Letters*, vol. 20, no. 10, pp. 1903–1907, 2021.
- [19] N. Ketkar, J. Moolayil, N. Ketkar, and J. Moolayil, “Introduction to pytorch,” *Deep Learning with Python: Learn Best Practices of Deep Learning Models with PyTorch*, pp. 27–91, 2021.
- [20] Y. Saad and M. H. Schultz, “GMRES: A generalized minimal residual algorithm for solving non-symmetric linear systems,” *SIAM Journal on Scientific and Statistical Computing*, vol. 7, no. 3, pp. 856–869, 1986.
- [21] The MNIST database can be downloaded at <http://yann.lecun.com/exdb/mnist/>.
- [22] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.



**Ji-Yuan Wang** received the B.S. degree from Communication University of China, Beijing, China, in 2021. He is currently pursuing the M.S. degree with the School of Integrated Circuit and Electronics, Beijing Institute of Technology, Beijing. His current research interests include deep learning and computational electromagnetics.



**Xiao-Min Pan** received the B.S. and M.S. degrees from Wuhan University, Wuhan, China, in 2000 and 2003, respectively, and the Ph.D. degree from the Institute of Electronics, Chinese Academy of Sciences, Beijing, China, in 2006. He is currently a professor with the School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing. He has authored or coauthored over 60 papers in refereed journals. His current research interests include high-performance methods in computational electromagnetics, artificial intelligence in electromagnetics, and electromagnetic compatibility analysis of complex systems.

Prof. Pan was the third recipient of the First Prize of Beijing Science and Technology Awards in 2011, the recipient of the New Century Excellent Talents in University in 2012 and of Young Elite Talents in Beijing in 2012. He received the Ulrich L. Rohde Innovative Conference Paper Award in 2016 and several other international academic awards. He served as a technical program committee co-chair, special sessions co-chairs, and technical program committee member for several international conferences and symposiums.