(Received 30 December 2024, Accepted 16 March 2025, Scheduled 27 March 2025)

Unsupervised Deep Learning-Based Source Synthesis Method for Fast Power Pattern Shaping

Lu Zhuang and Jun Ouyang*

School of Electronic Science and Engineering
University of Electronic Science and Technology of China (UESTC), Chengdu 611731, China

ABSTRACT: This paper introduces a deep neural network (DNN) training framework to tackle the general power pattern synthesis problem. Compared to the iterative solving method, the DNN-based approach offers a shorter response time, which is significant in adaptive scenarios. In contrast to the widely adopted supervised learning framework, the encoder-decoder network structure utilized in this paper does not necessitate the pre-synthesized results as the training label. The issue of difficult convergence in training caused by the non-uniqueness of the solution is well solved in our method.

1. INTRODUCTION

For general power pattern synthesis problems, such as array factor synthesis in array antennas and scattering pattern synthesis in metasurface design, the process can be divided into two steps. The first step involves determining a set of equivalent electromagnetic current sources to meet the specified power pattern requirements. The second step entails controlling the excitation and adjusting the radiation or scattering structure to ensure that the equivalent radiation source of the structure aligns with the source distribution obtained in the first step [1].

The synthesis of equivalent sources for a given radiation power pattern is typically approached using iterative methods. Stochastic optimization algorithms like Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) can gradually converge to the global optimal solution [2–4]. Alternatively, the iterative projection technique (IPT) based on inverse sources method can be utilized to find an equivalent source solution, ensuring convergence of the radiation power pattern within the prescribed upper and lower bound functions [1, 5, 6]. To reduce the computational complexity, compressive sensing (CS) [7], and other methods based on sub-arrays and clustered arrays have also been proposed [8, 9]. Although these methods offer high accuracy, their iterative nature makes the synthesis process non-real-time. To fulfill the real-time requirements of array and reconfigurable metasurface control in adaptive scenarios, a possible solution is to employ data-driven deep neural networks. Once the training of the DNN is completed, the equivalent source distribution can be directly computed based on different input power pattern specifications.

In the field of electromagnetics, the utilization of deep learning methods based on DNN has been extensively investigated [10–15]. However, the further advancement of deep learning is impeded by two significant challenges. Firstly, the acquisition of datasets in the field of electromagnetics is a costly endeavor compared to domains like image and text. This limitation restricts the creation of large-scale datasets, thereby hindering the effectiveness of commonly used supervised learning frameworks. Secondly, when employing neural networks to solve electromagnetic inverse problems, the non-uniqueness of the inverse solution presents a major difficulty [16]. For instance, a given radiation pattern can correspond to several entirely different source distributions. This non-uniqueness of solutions inhibits the convergence of supervised learning methods [17–20]. One approach to addressing the problem of multiple solutions in training is to use generative models [21], such as Generative Adversarial Networks (GANs). However, generative models have drawbacks such as training instability and difficulties in hyperparameter optimization.

In this paper, we present an unsupervised DNN model with an encoder-decoder structure [22] for synthesizing equivalent source distributions. The proposed model utilizes a set of parameter vectors describing the desired power pattern as input to the DNN and generates the corresponding equivalent source distribution as output. The main innovations of this paper can be summarized in the following two aspects. Firstly, unlike the commonly used supervised learning approach, an unsupervised deep learning training framework is constructed. This eliminates the need for specific source distribution labels in the training dataset. Consequently, the high cost associated with obtaining source distributions for different radiation power pattern specifications through other algorithms is overcome. Secondly, inspired by [19], a cascade network structure, consisting of an encoder and decoder, is adopted to overcome the training non-convergence issue caused by multiple solutions in electromagnetic inverse problems. A deep neural network, functioning as the encoder, converts the input power pattern specification into output source distribution. The decoder maps the encoder's output source distribution to its corresponding radia-

* Corresponding author: Jun Ouyang (yjou@uestc.edu.cn).

doi:10.2528/PIERC24123003

Published by THE ELECTROMAGNETIC ACADEMY

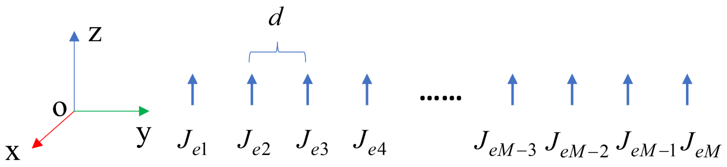

FIGURE 1. Illustration of dipole line array.

tion power pattern. The training of DNN utilizes a loss function which can quantify the difference between the decoder's output pattern and the desired far-field power pattern. This loss function design effectively resolves the issue of multi-value output in electromagnetic inverse problem.

The following sections will introduce the structure of the proposed network model in detail and demonstrate its effectiveness through a 1-dimensional source synthesis case.

2. PROBLEM STATEMENT

Consider the scenario in Fig. 1, where M electric dipole sources are arranged linearly along the y-axis with a spacing of d between adjacent dipoles. For simplicity, we assume that the dipole sources have only a z polarization component. Consequently, the far-field electric fields have only θ polarization component and can be expressed as (1),

$$[E]_N = [G]_{N\times M}[J]_M \tag{1}$$

where the M-dimensional vector $[J]_M$ represents M dipoles; $[E]_N$ is an N-dimensional vector denoting far-field electric fields radiated by $[J]_M$ with N sampling points; and $[G]_{N\times M}$ is the mapping matrix from $[J]_M$ to $[E]_N$, whose specific expression can be deduced from [23]. For the 1-dimensional problem, suppose that we only care about the 1-dimensional power pattern at $\theta = 90$ deg, and the far-field space sampling range is $\phi = [-90, 90]$.

In general, the desired radiation power pattern is characterized by a set of far-field pattern parameters, such as the main beam direction, half power beamwidth (HPBW), sidelobe level (SLL), nulls, and others [24]. The objective in this paper is to identify a combination of sources $[J]_M$, whose radiation power pattern satisfies the given far-field pattern specification. This is often formulated as a mask-constrained problem, where upper and lower bound mask functions are defined based on the far-field pattern specification [1, 25, 26]. The original problem is then transformed into finding a set of sources that minimizes the discrepancy between the radiation power pattern and the interval defined by the upper and lower bound mask functions. However, the mask-constrained problem introduces non-uniqueness in the solution, making it challenging for supervised learning methods to converge. In addition, through the truncated singular value decomposition (TSVD) of $[G]_{N\times M}$ [27], Equation (1) can be rewritten as,

$$[E]_N = [U_R]_{N\times\tau}[\Sigma_R]_{\tau\times\tau}[\gamma_R]_{\tau\times 1} \tag{2}$$

$$[J]_M = [V]_{M\times M}\begin{bmatrix}[\gamma_R]_{\tau\times 1}\\ [\gamma_{NR}]_{(M-\tau)\times 1}\end{bmatrix} \tag{3}$$

where unitary matrix $[V]_{M\times M}$ is the M-dimensional right singular value vector of matrix $[G]_{N\times M}$; $[U_R]_{N\times\tau}$ is the truncated left singular vector matrix with truncation order τ; and diagonal matrix $[\Sigma_R]_{\tau\times\tau}$ is the truncated singular value matrix of $[G]_{N\times M}$. In addition, $[\gamma_R]$ and $[\gamma_{NR}]$ constitute the coordinate of $[J]_M$ in $[V]_{M\times M}$ coordinate system [27].

According to Equation (2), the radiation field $[E]_N$ is exclusively determined by $[\gamma_R]$. This implies that while the value of $[\gamma_{NR}]$ may impact the value of the source $[J]_M$, it does not alter the radiation field $[E]_N$. The current associated with $[\gamma_R]$ is referred to as the radiative current J_R, while the current associated with $[\gamma_{NR}]$ is known as non-radiative current J_{NR} [28]. The total current $[J]_M$ can be expressed as (4) [28],

$$[J]_M = J_R + J_{NR}. \tag{4}$$

The presence of the non-radiative current J_{NR} contributes to the non-uniqueness of the solution in inverse source synthesis. In order to alleviate the issue of multiple solutions, our DNN focuses solely on the output goal of $[\gamma_R]$ (i.e., $[\gamma_{NR}] = 0$).

3. PROPOSED NETWORK STRUCTURE

The training framework of our proposed DNN is illustrated in Fig. 2(b). It comprises two main modules enclosed in dashed frames: the DNN encoder to be trained, represented by the blue dotted frame, and the decoder, denoted by the yellow dotted frame. The role of the DNN encoder is to map the input far-field power pattern specification from the dataset to the radiative source coefficient $[\gamma_R]$. Conversely, the decoder takes the radiative source coefficient $[\gamma_R]$ as input and computes the corresponding radiation power pattern as output. During training, the loss function is computed by comparing the output power pattern generated by the decoder with the upper and lower bound mask functions provided in the dataset. More detailed information regarding each block and the specific training process will be discussed in subsequent sections.

3.1. Dataset Format

Similar to the approach described in [26], we adopt a 4-channel vector representation for the input power pattern specification to facilitate feature extraction by the DNN. This representation is illustrated in Fig. 3. In the case of synthesizing a 1-dimensional far-field pattern in a half-space, we begin by discretizing the far-field region into a spatial angle distribution. The sampling interval used is 1 degree, and this angle distribution is denoted by Phi. The four characteristics that we focus on for the far-field pattern are Beam Direction, HPBW, SLL, and the pointing of nulls. Each characteristic parameter

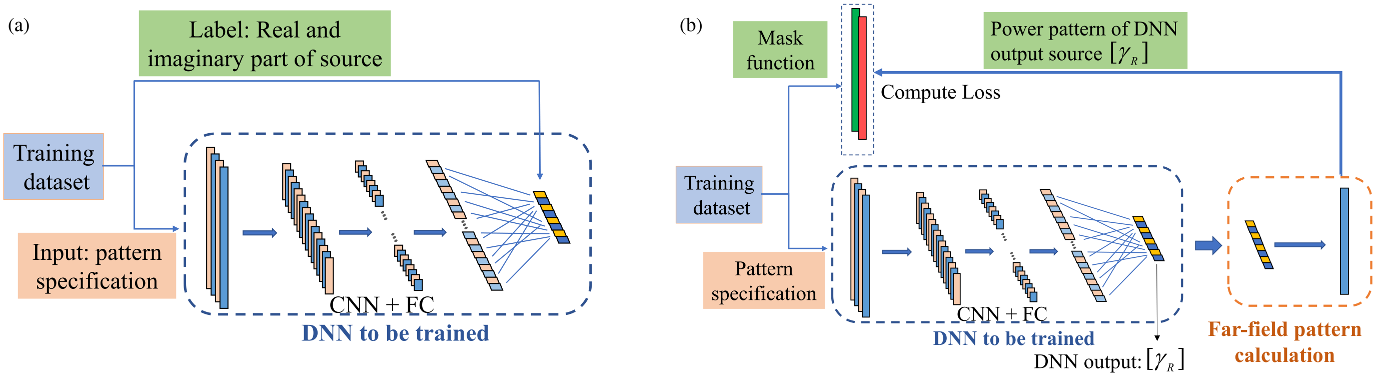

FIGURE 2. Illustration of the DNN training framework. (a) Supervised learning training framework. (b) Proposed training framework.

Phi	-90	-89	-88	-87	-86	-85	-84	-83	-82	…	88	89	90
Beam Direction	0	0	0	0	0	0	1	0	0	…	0	0	0
HPBW	0	0	0	0	1	1	1	1	1	…	0	0	0
SLL	-15	-15	-15	-15	-15	-15	-15	-15	-15	…	-15	-15	-15
Nulls	0	0	0	0	0	0	0	0	0	…	1	0	0

FIGURE 3. Illustration of the far-field power pattern dataset format.

is represented by a vector of the same length as the spatial sampling vector Phi. For the parameters of Beam Direction and Nulls, a value of 1 is assigned to the corresponding pointing angle, while the remaining angular areas are filled with 0. In the case of the HPBW parameter, the region within the maximum pointing half-power beamwidth is assigned a value of 1, while the other areas are assigned 0. For SLL vector, all spatial sampling points are assigned the desired SLL values. As shown in the example in Fig. 3, the main beam of the pattern points to $Phi = -84$ degrees, with an HPBW of 4 deg from $Phi = -86$ deg to $Phi = -82$ deg. The null is located at $Phi = 88$ deg, and the SLL is -15 dB.

The upper and lower bound mask functions are generated according to the power pattern specification data. The mask functions corresponding to Fig. 3 are shown in Fig. 4, and the upper bound function is set to -30 dB in the nulls region (3 degrees wide, centered around the Nulls parameter). The upper bound function is fixed at 0 dB in the main beam region (4 degrees wider than the HPBW) and takes the value specified by the SLL parameter elsewhere. The lower bound function is set to 0 dB at the main beam pointing angle and -3 dB within the HPBW region. The remaining angle range of the lower bound function is fixed at -100 dB. By adhering to our mask functions generation rules, the resulting far-field pattern constrained by the mask functions can satisfy the desired pattern specification.

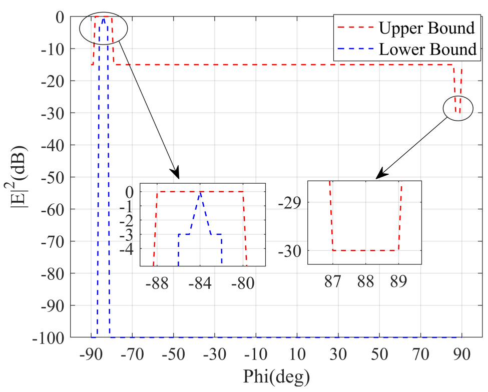

FIGURE 4. Illustration of the mask function dataset.

3.2. Deep Neural Network Model

The proposed DNN model structure is illustrated in Fig. 5. It consists of five 1-dimensional convolution blocks and one fully connected layer. Each convolution block comprises a 1-dimensional convolution layer (Conv1d), a LeakyReLU activation layer, and a 1-dimensional maximum pooling layer (Maxpool1d). The Conv1d layer has a kernel size parameter of 3, and both the stride and padding parameters are set to 1. The Maxpool1d layer has a kernel size and stride of 2. The 4-channel input far-field power pattern specification doubles the number of channels and halves the data length with each passing through a convolution block. After five convolution blocks, the number of channels reaches 128, and the data length is downsampled to 5. Subsequently, the data from all channels are flattened into a vector of length 640, which is then mapped to a vector of length 46 through a fully connected layer. The activation function used after the fully connected layer is the Tanh function, which maps the output data to the interval of -1 to 1. It is necessary to note that, in the given example, the truncation order τ is set to 23, while the output vector length of the DNN is 46. This is because the coefficient $[\gamma_R]$ is complex, with two dimensions: the real part and imaginary part. The first 23 elements of the output vector represent the real part of $[\gamma_R]$, while the rest represents its imaginary part.

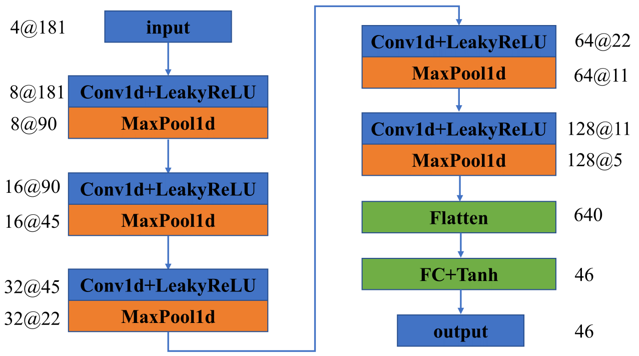

FIGURE 5. Illustration of the proposed deep neural network structure.

3.3. Far-Field Pattern Calculation

In contrast to the general supervised learning framework, the training of the DNN in this paper involves an additional pattern calculation module, known as the decoder. The predicted $[\gamma_R]$ from the DNN is processed by the decoder to obtain the corresponding radiation pattern. The training objective is to minimize the disparity between this radiation pattern and the desired pattern constrained by the given bound mask function. The far-field power pattern calculation process is as follows.

The complex number representation of $[\gamma_R]$ is synthesized by combining the real and imaginary parts output by the DNN. Using Equation (2), the radiation fields $[E]_N$ corresponding to the predicted source $[\gamma_R]$ can be computed. The power pattern is then defined as the square of the absolute value of the electric fields E. To facilitate comparison, the power pattern is normalized. This is achieved by scaling the power pattern so that the normalized power pattern $\widetilde{|E|}^2$ ranges from 0 to 1.

3.4. DNN Loss Computation

The loss function is defined to quantify the deviation between $\widetilde{|E|}^2$ and the boundary mask constraint function. The total loss consists of two terms, as expressed by Equation (5),

$$\mathcal{L}_{total} = \mathcal{L}_{upper} + \mathcal{L}_{lower} \tag{5}$$

where $\mathcal{L}_{total}$ represents the total loss; $\mathcal{L}_{upper}$ denotes the loss incurred when $\widetilde{|E|}^2$ exceeds the upper bound mask function; and $\mathcal{L}_{lower}$ represents the loss incurred when $\widetilde{|E|}^2$ falls below the lower bound mask function.

To calculate the loss, the logarithmic form of the mask function in the dataset is converted to its absolute value form to match the range of the normalized $\widetilde{|E|}^2$. The following definitions are used:

$$|E|^{clamp}_{upper} = \begin{cases} \widetilde{|E|}^2, & \text{if} \quad \widetilde{|E|}^2 < \mathcal{M}_{upper} \\ \mathcal{M}_{upper}, & \text{if} \quad \widetilde{|E|}^2 >= \mathcal{M}_{upper} \end{cases} \tag{6}$$

$$|E|^{clamp}_{lower} = \begin{cases} \widetilde{|E|}^2, & \text{if} \quad \widetilde{|E|}^2 > \mathcal{M}_{lower} \\ \mathcal{M}_{lower}, & \text{if} \quad \widetilde{|E|}^2 <= \mathcal{M}_{lower} \end{cases} \tag{7}$$

where $\mathcal{M}_{upper}$ and $\mathcal{M}_{lower}$ are the upper and lower bound mask functions in dataset. Then, $\mathcal{L}_{upper}$ and $\mathcal{L}_{lower}$ are defined as (8) and (9). The mean square error (MSE) function is utilized as the loss function [29], which has been found to yield favorable results in our experiments.

$$\mathcal{L}_{upper} = \text{MSE}\left(\frac{\widetilde{|E|}^2 - |E|^{clamp}_{upper}}{\mathcal{M}_{upper}}, 0\right) \tag{8}$$

$$\mathcal{L}_{lower} = \text{MSE}\left(\frac{|E|^{clamp}_{lower} - \widetilde{|E|}^2}{\mathcal{M}_{lower}}, 0\right) \tag{9}$$

4. NETWORK MODEL ASSESSMENT

To prove the effectiveness of the DNN framework, a case with 32 dipoles (i.e., $M = 32$) to be synthesized is demonstrated in this paper. The spacing d equals $0.3\lambda_0$, where λ_0 is the free space wavelength at 4.5 GHz. The singular value of the matrix $[G]$ in formula (1) is shown in Fig. 6(a), and the TSVD truncation index τ is set to $\tau = 23$. The DNN model structure to be trained is shown in Fig. 5, and the PyTorch deep learning framework is used to train the model. In our dataset, there are a total of 51200 data, 80% of which are used as training datasets and 20% as testing datasets. All datasets are randomly generated, with Beam Direction and Nulls ranging from -50 to 50 deg, HPBW ranging from 6 to 10 deg, and SLL ranging from -20 to -10 dB. The upper and lower bound masks are generated based on the correspond pattern specifications using a mask generation function, and the total time consumption for dataset generation is only 0.87 seconds. The Adam optimizer is used to update DNN weight parameters.

The training loss and testing loss exhibit a decreasing trend as the number of training epochs increases, as depicted in Fig. 6(b). To facilitate visualization, the loss values in Fig. 6(b) are normalized and presented in logarithmic form. As the number of training epochs increases, the loss function gradually converges, and the testing loss becomes nearly identical to the training loss. This indicates that the DNN training process does not suffer from over-fitting issues. After 500 epochs of training, the testing loss finally converges to 0.00024.

Figure 7 shows the far-field pattern synthesized by the trained DNN. The input pattern specification of Fig. 7(a) has the Beam Direction = 0 deg, HPBW=8deg, SLL = -20 dB, Nulls = -15 deg. Meanwhile, the input specification of Fig. 7(b) is with the Beam Direction = 50 deg, HPBW = 10 deg, SLL=-18 dB, Nulls=-10 deg. It can be seen that both of the patterns in Fig. 7(a) and Fig. 7(b) can be well limited to the bound mask function and satisfy the desired far-field specification. The reasoning losses of Fig. 7(a) and Fig. 7(b) are 0.00010 and 0.00085, respectively. In addition, it should be noted that the output $[\gamma_R]$ of DNN is the source distribution in $[V]_{M\times M}$ coordinate system, and the source distribution in spatial Cartesian coordinate system needs to be transformed according to formula (3). The value of $[\gamma_{NR}]$ can be set arbitrarily and will not affect the result of far-field radiation pattern. If $[\gamma_{NR}]$ is set to zero vector, the resulting source

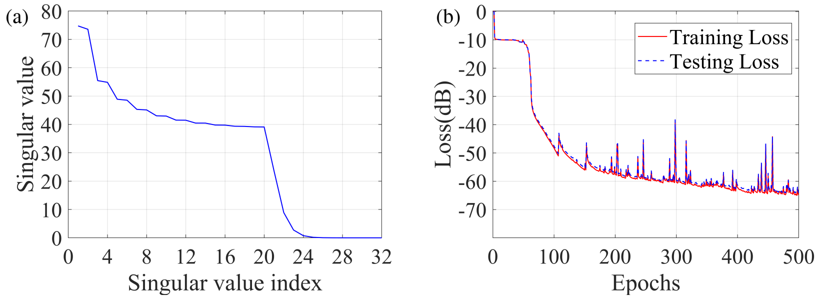

FIGURE 6. (a) Plot of the singular value. (b) Normalized training and testing loss of DNN versus the training epochs.

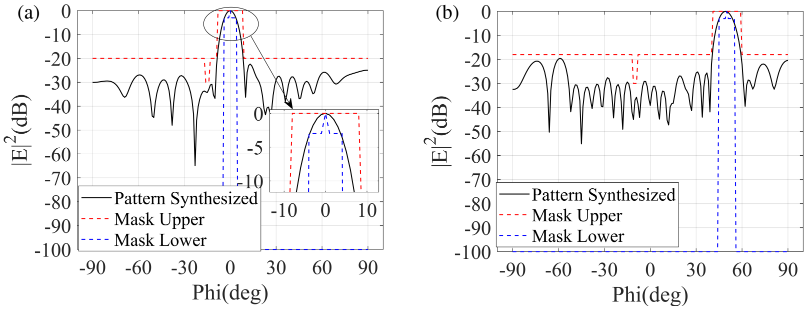

FIGURE 7. Synthesized far-field power pattern based on trained DNN. (a) Beam Direction: 0 deg, HPBW: 8 deg, SLL: -20 dB, Nulls: -15 deg. (b) Beam Direction: 50 deg, HPBW: 10 deg, SLL: -18 dB, Nulls: -10 deg.

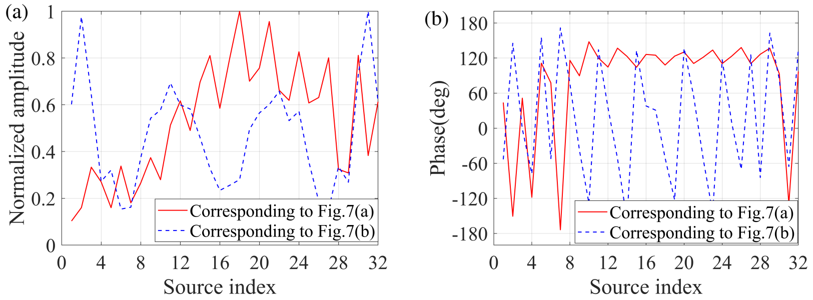

FIGURE 8. Synthesized minimum norm source. (a) Amplitude. (b) Phase.

distribution is called the minimum norm solution [28]. The minimum norm source corresponding to Fig. 7 is shown in Fig. 8.

To compare and demonstrate the reasoning capability of the trained DNN, we conducted an experimental comparison with the synthesis performance of the traditional PSO algorithm. We randomly selected 512 pattern specifications from the test dataset and applied the PSO algorithm iteratively to determine the radiation source distribution. To improve the global search ability of the PSO algorithm, we set a relatively large $populationsize$ parameter, specifically, $populationsize = 400$. The PSO algorithm iterations were halted when the PSO loss reached the average reasoning loss of the trained DNN, specifically 0.00024. Fig. 9 illustrates the number of PSO iterations corresponding to the 512 source synthesis cases, showing an average iteration count of 176. This indicates that the reasoning performance of the DNN network trained using the proposed framework is comparable to the effectiveness of the PSO algorithm with a population size of 400 after 176 iterations. Under the same computational hardware conditions, the time taken for 176 iterations of PSO is 7.5 seconds, while the reasoning time for the DNN is only 0.001 seconds.

To illustrate the inability of traditional supervised learning training framework to converge due to the non-uniqueness of solutions, we provide the training results under a supervised learning framework as a reference. As shown in Fig. 2(a), the real and imaginary parts of the sources, which satisfy the input pattern specification, are used as training labels, and this is obtained through pre-solving using the PSO algorithm. The goal of training is to make the output values of the DNN as consistent as possible with the label values, and the MSE loss function is used to evaluate the difference. The input pattern specifications of training and testing datasets are completely consistent with those used in the proposed unsupervised training framework. The convergence of training and testing loss in the training of supervised learning framework is shown in

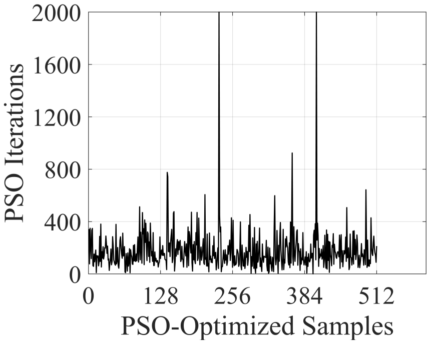

FIGURE 9. The number of iterations of the PSO algorithm under the same loss condition.

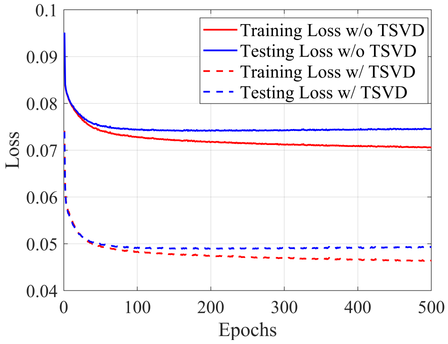

FIGURE 10. Supervised learning framework training and testing loss.

Fig. 10. After training for 100 epochs, the testing loss has essentially converged, but the decrease in comparison to the initial loss value is very limited. We also compared the training differences between using the minimum norm source labels obtained through TSVD and using source labels without TSVD. It is evident from the figure that using the minimum norm current sources as labels leads to a smaller convergence loss, as the absence of non-radiative sources in the labels alleviates the non-uniqueness of solutions.

To better compare the reasoning capabilities of the DNN trained under the two frameworks, we use the loss from Equations 8 and 9 to evaluate the trained supervised learning networks. The average testing loss for supervised learning with TSVD is 1839.7 and without TSVD is 1891.7. This shows a significant difference compared to the loss of 0.00024 achieved by the proposed DNN training framework. This is because even though the non-uniqueness issue of sources caused by the singular values of the radiation matrix $[G]$ can be alleviated by truncating the singular values, the power pattern constrained by the mask function are inherently non-unique. In fact, for the same power pattern, the corresponding phase pattern of the radiation field can also be non-unique. These non-uniqueness aspects make the convergence of supervised learning training very poor.

5. CONCLUSION

This paper presents a deep neural network-based source synthesis framework for far-field power pattern shaping. Unlike traditional supervised learning approaches, the proposed framework operates in an unsupervised manner, eliminating the need for providing source results corresponding to different power patterns during training. Moreover, the proposed training framework overcomes the convergence challenges caused by the non-uniqueness of solutions in inverse problems. Since the trained DNN has no iterative process of forward computation, it is very suitable for some real-time scenarios.

REFERENCES

[1] Oliveri, G., P. Rocca, M. Salucci, and A. Massa, "Holographic smart EM skins for advanced beam power shaping in next generation wireless environments," *IEEE Journal on Multiscale and Multiphysics Computational Techniques*, Vol. 6, 171–182, 2021.

[2] Mitchell, M., *An Introduction to Genetic Algorithms*, MIT Press, 1998.

[3] Kennedy, J. and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 — International Conference on Neural Networks*, Vol. 4, 1942–1948, 1995.

[4] Ismail, T. H. and Z. M. Hamici, "Array pattern synthesis using digital phase control by quantized particle swarm optimization," *IEEE Transactions on Antennas and Propagation*, Vol. 58, No. 6, 2142–2145, 2010.

[5] Bucci, O. M., G. Franceschetti, G. Mazzarella, and G. Panariello, "Intersection approach to array pattern synthesis," in *IEE Proceedings H (Microwaves, Antennas and Propagation)*, Vol. 137, No. 6, 349–357, 1990.

[6] Rocca, P., R. L. Haupt, and A. Massa, "Sidelobe reduction through element phase control in uniform subarrayed array antennas," *IEEE Antennas and Wireless Propagation Letters*, Vol. 8, 437–440, 2009.

[7] Oliveri, G., M. Carlin, and A. Massa, "Complex-weight sparse linear array synthesis by Bayesian compressive sampling," *IEEE Transactions on Antennas and Propagation*, Vol. 60, No. 5, 2309–2326, 2012.

[8] Abdulqader, A. J., J. R. Mohammed, and R. H. Thaher, "Antenna pattern optimization via clustered arrays," *Progress In Electromagnetics Research M*, Vol. 95, 177–187, 2020.

[9] Abdulkader, A. J., J. R. Mohammed, and R. H. Thaher, "Phase-only nulling with limited number of controllable side elements," *Progress In Electromagnetics Research C*, Vol. 99, 167–178, 2020.

[10] Shi, D., C. Lian, K. Cui, Y. Chen, and X. Liu, "An intelligent antenna synthesis method based on machine learning," *IEEE Transactions on Antennas and Propagation*, Vol. 70, No. 7, 4965–4976, 2022.

[11] Niu, C., M. Phaneuf, T. Qiu, and P. Mojabi, "A deep learning-based approach to design metasurfaces from desired far-field specifications," *IEEE Open Journal of Antennas and Propagation*, Vol. 4, 641–653, 2023.

[12] Koziel, S. and M. Abdullah, "Machine-learning-powered EM-based framework for efficient and reliable design of low scattering metasurfaces," *IEEE Transactions on Microwave Theory and Techniques*, Vol. 69, No. 4, 2028–2041, 2021.

[13] Zhou, Z., Z. Wei, J. Ren, Y. Yin, G. F. Pedersen, and M. Shen, "Two-order deep learning for generalized synthesis of radiation patterns for antenna arrays," *IEEE Transactions on Artificial Intelligence*, Vol. 4, No. 5, 1359–1368, 2022.

[14] Zhang, J., C. Qu, X. Zhang, and H. Li, "A deep learning method for the phase-only pattern synthesis," in *2023 International Applied Computational Electromagnetics Society Symposium (ACES — China)*, 1–2, Hangzhou, China, Aug. 2023.

[15] Yang, X., D. Yang, Y. Zhao, J. Pan, and Y. Chen, "Synthesis of linear sparse array using DNN-based machine-learning method," *IEEE Transactions on Antennas and Propagation*, Vol. 71, No. 8, 6513–6522, 2023.

[16] Ren, S., A. Mahendra, O. Khatib, Y. Deng, W. J. Padilla, and J. M. Malof, "Inverse deep learning methods and benchmarks for artificial electromagnetic material design," *Nanoscale*, Vol. 14, No. 10, 3958–3969, 2022.

[17] Hansen, P. C., *Rank-Deficient and Discrete ILL-posed Problems: Numerical Aspects of Linear Inversion*, SIAM, 1998.

[18] Kabir, H., Y. Wang, M. Yu, and Q.-J. Zhang, "Neural network inverse modeling and applications to microwave filter design," *IEEE Transactions on Microwave Theory and Techniques*, Vol. 56, No. 4, 867–879, 2008.

[19] Liu, D., Y. Tan, E. Khoram, and Z. Yu, "Training deep neural networks for the inverse design of nanophotonic structures," *ACS Photonics*, Vol. 5, No. 4, 1365–1369, 2018.

[20] Yuan, L., L. Wang, X.-S. Yang, H. Huang, and B.-Z. Wang, "An efficient artificial neural network model for inverse design of metasurfaces," *IEEE Antennas and Wireless Propagation Letters*, Vol. 20, No. 6, 1013–1017, 2021.

[21] Gu, Z., D. Li, Y. Wu, Y. Fan, C. Yu, H. Chen, and E.-P. Li, "A solution to the dilemma for FSS inverse design using generative models," *IEEE Transactions on Antennas and Propagation*, Vol. 71, No. 6, 5100–5109, 2023.

[22] Prince, S. J. D., *Understanding Deep Learning*, MIT Press, 2023.

[23] Osipov, A. V. and S. A. Tretyakov, *Modern Electromagnetic Scattering Theory with Applications*, John Wiley & Sons, 2017.

[24] Brown, T., C. Narendra, Y. Vahabzadeh, C. Caloz, and P. Mojabi, "On the use of electromagnetic inversion for metasurface design," *IEEE Transactions on Antennas and Propagation*, Vol. 68, No. 3, 1812–1824, 2019.

[25] Cui, C., W. T. Li, X. T. Ye, Y. Q. Hei, P. Rocca, and X. W. Shi, "Synthesis of mask-constrained pattern-reconfigurable nonuniformly spaced linear arrays using artificial neural networks," *IEEE Transactions on Antennas and Propagation*, Vol. 70, No. 6, 4355–4368, 2022.

[26] Niu, C., M. Kelly, and P. Mojabi, "An encoder-only transformer to generate power patterns from far-field performance criteria," in *2022 16th European Conference on Antennas and Propagation (EuCAP)*, 1–4, Madrid, Spain, Mar. 2022.

[27] Caorsi, S. and G. L. Gragnani, "Inverse-scattering method for dielectric objects based on the reconstruction of the nonmeasurable equivalent current density," *Radio Science*, Vol. 34, No. 1, 1–8, 1999.

[28] Salucci, M., A. Gelmini, G. Oliveri, N. Anselmi, and A. Massa, "Synthesis of shaped beam reflectarrays with constrained geometry by exploiting nonradiating surface currents," *IEEE Transactions on Antennas and Propagation*, Vol. 66, No. 11, 5805–5817, 2018.

[29] Team, P., "Pytorch official website," Available: https://pytorch.org/docs/, 2023.