Abstract—In this paper, we propose the neural Born iterative method (NeuralBIM) for solving 2D inverse scattering problems (ISPs) by drawing on the scheme of physics-informed supervised residual learning (PhiSRL) to emulate the computing process of the traditional Born iterative method (TBIM). NeuralBIM employs independent convolutional neural networks (CNNs) to learn the alternate update rules of two different candidate solutions regarding the residuals. Two different schemes are presented in this paper, including the supervised and unsupervised learning schemes. With the data set generated by the method of moments (MoM), supervised NeuralBIM are trained with the knowledge of total fields and contrasts. Unsupervised NeuralBIM is guided by the physics-embedded objective function founding on the governing equations of ISPs, which results in no requirement of total fields and contrasts for training. Numerical and experimental results further validate the efficacy of NeuralBIM.

Keywords—Inverse scattering problem, Born iterative method, deep learning, supervised learning, unsupervised learning

I. INTRODUCTION

Inverse scattering are an ubiquitous tool to determine the nature of unknown scatterers with knowledge of scattered electromagnetic (EM) fields [1], which has been applied widely across nondestructive testing [2], biomedical imaging [3], microwave imaging [4], geophysical exploration [5], [6], etc. The intrinsic nonlinearity and ill-posedness pose long-standing challenges to ISPs [1]. Many efforts are devoted to performing reliable inversion by addressing these two challenges, yielding noniterative and iterative inversion methods. Noniterative inversion methods linearize ISPs under specific conditions, such as Born or Rytov approximation method [7], [8], back-propagation (BP) method [9], etc. Iterative inversion methods transform ISPs into optimization problems of which optimal solutions are identified via an iterative process, for example, Born or distorted Born iterative method [10], [11], contrast source inversion method [12], [13], subspace-based optimization [14], level set method [15], etc. Prior information can be employed as a regularization to constrain and stabilize the iterative method for better inversion performance [16]. Commonly applied regularizations include Tikhonov [17], total variation (TV) [18] and multiplicative regularizations [12]. Besides, the compressive sensing paradigm can introduce an important type of regularization by decomposing the ISP solution based on a set of properly chosen basis functions [19]–[23]. Noniterative inversion methods hold good computational efficiency but limited applicability, while iterative ones can perform reliable inversion but suffer from immense computational loads.

Recently, machine learning (ML), especially deep learning (DL), has been applied in the EM field, such as forward modeling [24]–[26], array antenna design [27]–[29] and ISPs [30], [31], etc., which demonstrates unprecedented numerical precisions and computational efficiency [30]. The supervised descent method (SDM) is trained offline to learn and store the descent directions to guide the online inversions [32]. The powerful learning capacity of deep neural networks (DNNs), like CNNs [33], [34], generative adversarial networks [35], etc., is leveraged to retrieve the properties of scatterers from the scattered fields directly. Moreover, the combination of DL and traditional ISP methods can achieve improved computational efficiency and inversion quality [30]. DNNs are...
trained to enhance the initial guess of inversions generated by noniterative inversion methods [36]. Conversely, iterative inversion methods can take as input the reconstructions predicted by DNNs [39], [40]. Despite the remarkable success, most of the works mentioned above regard DNNs as "black-box" approximators, and the challenge of designing DNNs effective for ISPs has not been fully addressed.

The ordinary or partial differential equations (ODEs/PDEs) theory enables better insights into DNN properties, which guides the design of effective DNN architectures with better robustness and interpretability. In [41], ResNet, PolyNet, FractalNet, and RevNet are linked to different numerical methods of ODEs or PDEs, respectively. ResNet can also be interpreted from the perspective of dynamical systems [42], [43] or combined with the fixed-point iterative method [44]. The spectral method and random matrix theory are combined [45] or combined with the fixed-point iterative method [44]. PDE-Net is proposed based on the similarities between finite difference operators and convolution operations [47]. New structures of ResNets for image processing are motivated by parabolic, and hyperbolic PDEs [48].

In this paper, we propose the neural Born iterative method by drawing on PhiSRL to emulate the computational process of the traditional Born iterative method. Combining ResNet [49] and the fixed-point iterative method, PhiSRL iteratively modifies a candidate solution until convergence by applying CNNs to predict the modification. NeuralBIM builds the learned parameterized functions for the alternate update process of TBIM by drawing on the idea of PhiSRL. Therefore, NeuralBIM bears the same computation as TBIM to reconstruct the total fields and scatterers simultaneously. The supervised and unsupervised learning schemes of NeuralBIM are demonstrated and validated in this paper. Supervised NeuralBIM is first presented to be trained by applying MoM to generate total fields and contrasts as labels. Unsupervised NeuralBIM is proposed by deriving the physics-embedded objective function from the governing equations of ISPs, which further gets rid of total fields and contrasts for training. Numerical and experimental results verify the efficacy of NeuralBIM.

This paper is organized as follows. Section II formulates inverse scattering problems. Section III introduces traditional Born iterative method. Both supervised and unsupervised neural Born iterative methods are introduced in Section IV. Section V demonstrates the numerical and experimental results of neural Born iterative method. Finally, Section VI draws conclusions and outlooks.

II. INVERSE SCATTERING PROBLEMS

In a 2D domain of interest (DoI) $D$, the transverse magnetic (TM) polarized line sources illuminate the unknown scatterers. The incident field $E^i$, total field $E^t$ and scattered field $E^s$ are related by

$$E^t(r) = E^i(r) + k_b^2 \int_D G_D(r, r') \chi(r') E^t(r') dr', r \in D$$

(1)

where $k_b$ is the wavenumber, $G_D$ and $G_S$ denote Green’s functions in free space, $\chi(r)$ is the scatterer contrast and $S$ is the observation domain. Eq. (1) and Eq. (2) can be discretized into a linear system of matrix equations by employing MoM:

$$\begin{align*}
(I - G_D \chi) E^t &= E^i \\
E^s &= G_S \chi E^t
\end{align*}$$

(3)

(4)

ISP aims to reconstruct $\chi$ based on the governing equations (Eq. (3) and Eq. (4)).

III. TRADITIONAL BORN ITERATIVE METHOD

Traditional Born iterative method is one of the effective algorithms for solving ISPs [7]. TBIM reconstructs unknown scatterers with the knowledge of incident and scattered fields via an alternate update process. TBIM first establishes the following minimization problem:

$$\min_{\chi} ||E^t - G_S \chi E^i||^2.$$  

(5)

The optimal solution $\chi_{next}$ of Eq. (5) serves as the update of $\chi$.

Then, the total field $E^t$ can be updated with $\chi_{next}$:

$$E^t_{next} = (I - G_D \chi_{next})^{-1} E^i$$

(6)

In TBIM, Eq. (3) and Eq. (6) are applied alternatively in the iterative process until $E^t$ and $\chi$ satisfy the stop criterion. In the first update of $\chi$, the incident field $E^i$ is used to approximate the total field $E^t$ in Eq. (5). Here, Eq. (5) is viewed and solved as a least-square problem as stated in [10], [11]. Eq. (6) is solved directly by MoM where the computation of $G_D$ is accelerated by fast Fourier transform. The algorithm flow of TBIM is summarized in Algorithm 1.

Algorithm 1 Traditional Born Iterative Method

**Initialization:** $E^t_0 = E^i$, $k = 0$, stopflag = 1

**while** stopflag do

**step 1:** $\chi_{k+1} = \arg \min_{\chi} ||E^t - G_D \chi E^t_k||^2$

**step 2:** $E^t_{k+1} = (I - G_D \chi_{k+1})^{-1} E^i$

**step 3:** $k = k + 1$

**if** stop criteria is satisfied **then**

stopflag = 0

**end if**

**end while**

**output:** $\chi_k$, $E^t_k$

IV. NEURAL BORN ITERATIVE METHOD

Neural Born iterative method is motivated by the physics-informed supervised residual learning. PhiSRL is built by interpreting ResNet as the fixed-point iterative method [44]. PhiSRL aims to solve the matrix equation $Ax = b$ and its $k + 1$-th update equation can be formulated as:

$$x_{k+1} = x_k + F_{k+1}(R_k, \Theta_{k+1}),$$

(7)
where $F_{k+1}$ and $\Theta_{k+1}$ denote the CNN and corresponding parameter set, $R_k = b - Ax_k$ denotes the residual of $k$-th iteration. It can be observed from Eq. (7) that PhiSRL applies TBIM and NeuralBIM are designed by applying CNNs to alternatively update the total field and contrasts. The employed CNN in Figure 3. Tanh nonlinearity can provide both positive and negative values, which allows an adaptive modification of the candidate solution. The input and output channels are also the same architecture but different parameter sets. The employed CNN comprises five stacked operations including $3 \times 3$ convolution, batch normalization and Tanh nonlinearity, as shown in Figure 3. The algorithm flow of NeuralBIM is summarized in Algorithm 2.

Algorithm 2 Neural Born Iterative Method

**initialization:** $E_0^{tot} = E^{inc}$, $x_0 = 0$, $k = 0$, MaxIter

while $k < \text{MaxIter}$ do

step 1: $R_k^{E'} = E^s - G_S x_k E_k^t$

step 2: $x_{k+1} = x_k + F_k^{x'}(R_k^{E'}, x_k, \Theta_{k+1})$

step 3: $R_k^{E'} = E^t - (I - G_D x_{k+1}) E_{k+1}^t$

step 4: $x_{k+1} = x_{k+1} + F_k^{x'}(R_k^{E'}, x_{k+1})$

step 5: $k = k + 1$

end while

**output:** $x_{\text{MaxIter}}$, $E_{\text{MaxIter}}$

In this paper, NeuralBIM is assumed to have 7 iterative modules as depicted in Figure 2. Figure 3 illustrates the detailed structure of the iterative block. The $\chi$-CNN and $E^t$-CNN to modify contrasts and total fields share the same structure but different parameter sets. The employed CNN comprises five stacked operations including $3 \times 3$ convolution, batch normalization and Tanh nonlinearity, as shown in Figure 3. Tanh nonlinearity can provide both positive and negative values, which allows an adaptive modification of the candidate solution. The input and output channels are also denoted in the Figure 3. The $c_1$ and $c_2$ are 4 and 2 for updating lossy contrasts, and 2 and 2 for updating the total fields.

The training schemes of NeuralBIM include the supervised and unsupervised learning schemes:

**Supervised learning scheme:** The supervised learning scheme trains the NeuralBIM with the total fields and contrasts as labels. The objective function is defined as:

$$\text{loss}_{\chi} = \text{mse}_{\chi},$$

where $\text{loss}_{\chi}$ and $\text{loss}_{E'}$ denote the losses of total fields and contrasts respectively. The $\text{loss}_{E'}$ is defined as the mean squared error (MSE):

$$\text{loss}_{E'} = \frac{1}{N_{E'}} ||E^{t'} - E^{t'}||^2,$$
where $E'$ and $E''$ denote the ground-truth and predicted total field, $N_{E'}$ denotes the element number in the vector of $E''$, $\| \cdot \|_F$ represents Frobenius norm. The $\text{loss}_\chi$ adds the TV regularization item to the MSE of contrasts:

$$\text{loss}_\chi = \frac{1}{N_\chi}||\chi^a - \chi'||_1^2 + \alpha \frac{1}{N_\chi}||\nabla \chi'||_1,$$

(12)

where $\chi^a$ and $\chi'$ are the ground-truth and inverted contrast, $N_\chi$ denotes the element number in the vector of $\chi'$, $\| \cdot \|_1$ is fixed as 0.0001. In the supervised training scheme, NeuralBIM is trained to generate accurate predictions of total fields and contrasts. It takes a certain amount of time to generate training data samples by applying MoM to solve Eq. (3) and Eq. (4). This scheme is also different from the practical applications where total fields and contrasts are unknown.

**Unsupervised learning scheme:** In the unsupervised learning scheme, total fields and contrasts are unknown when training NeuralBIM. The training of NeuralBIM is constrained and guided by the governing equations of ISPs as formulated in Eq. (3) and Eq. (4). The objective function is defined as:

$$\text{obj}_{\text{unsu}} = \text{res}_{E_1} + \text{res}_{E_2}.$$  

(13)

The $\text{res}_{E_1}$ is formulated based on Eq. (3):

$$\text{res}_{E_1} = \frac{1}{N_{E_1}}\|\textbf{E}^a - \textbf{G}_S\chi'E'' - \textbf{E}'\|^2_F.$$  

(14)

The $\text{res}_{E_2}$ is defined based on Eq. (4):

$$\text{res}_{E_2} = \frac{1}{N_{E_2}}\|\textbf{E}^a - \textbf{G}_S\chi'E''\|^2_F + \alpha \frac{1}{N_\chi}||\nabla \chi'||_1,$$

(15)

where $N_{E_1}$ and $N_{E_2}$ denote the element number in the vector of $\textbf{E}^a$ and $\textbf{E}^a$, $\| \cdot \|_1$ is the TV regularization and $\alpha$ is fixed as 0.0001. It is noted that the incident and scattered fields in Eq. (14) and Eq. (15) are assumed to be known in ISPs and the Green functions are determined by the measurement setup of ISPs. The unsupervised training scheme aims to train NeuralBIM to generate solutions simultaneously satisfying Eq. (3) and Eq. (4) by leveraging the existing physical law and information. The TV regularization can stabilize the training process by enforcing the boundaries of reconstructions.

The computational complexity of NeuralBIM is determined by two parts: the matrix multiplication in calculating residuals and the basic computation in the CNN. In a single iteration, NeuralBIM has two subsequent branches to update $\chi$ and $\textbf{E}'$ respectively. The numbers of transmitters, receivers and pixels are denoted as $T$, $R$ and $P$ for simplification. In the $\chi$-branch, the complexity of calculating $\mathcal{R}_k^\chi$ is $O(TRP)$ due to the dense matrix multiplication. The complexity of the $\chi$-CNN is dominated by convolution operations. For a single convolutional layer, the complexity is $O(PK^wK^hC^i_lC^o)$, where $K^w$ and $K^h$ are the width and height of the convolutional kernel, $C^i_l$ and $C^o$ are the input and output channels. Assuming the $\chi$-CNN has $L$ layers, its complexity can be written as: $O(\sum_{l=1}^L PK^w_lK^h_lC^i_lC^o_l)$. Note that all feature maps output by the convolutional layers contain $P$ pixels. Similarly, in the $\textbf{E}'$-branch, the complexity of the calculation of $\mathcal{R}_k^\textbf{E}'$ and the $\textbf{E}'$-CNN are $O(TP^2)$ and $O(\sum_{l=1}^L PK^w_lK^h_lC^i_lC^o_l)$. For a single convolutional layer, the complexity can be written as: $O(\sum_{l=1}^L PK^w_lK^h_lC^i_lC^o_l)$. Note that all feature maps output by the convolutional layers contain $P$ pixels. Similarly, in the $\textbf{E}'$-branch, the complexity of the calculation of $\mathcal{R}_k^\textbf{E}'$ and the $\textbf{E}'$-CNN are $O(TP^2)$ and $O(\sum_{l=1}^L PK^w_lK^h_lC^i_lC^o_l)$ respectively. The complexity of calculating $\mathcal{R}_k^\chi$ can be further reduced by fast Fourier transform in the case of large-scale problems. For NeuralBIM with $N$ iterations, the total computational complexity is $O(NTRP + N \sum_{l=1}^L PK^w_lK^h_lC^i_lC^o_l + NTP^2 + N \sum_{l=1}^L PK^w_lK^h_lC^i_lC^o_l)$.}

**V. NUMERICAL RESULTS**

In this section, we first validate both supervised and unsupervised NeuralBIM with synthetic data inversion. The efficacy of unsupervised NeuralBIM is further verified with experimental data inversion. Supervised and unsupervised NeuralBIM are implemented in Pytorch and computed on three Nvidia V100 GPUs. The Adam optimizer is adopted to train NeuralBIM. The learning rate is initialized as 0.002 and decayed by $\sim 0.8$ every 20 epochs.

**TABLE I**

| Cylinder | Radius | Contrast (Real) | Contrast (Imag) |
|----------|--------|----------------|----------------|
| A        | 0.015-0.035m | [0, 1]     | [-1, 0]     |
| B        | 0.015-0.035m | [0, 1]     | [-2, -1]     |

**A. Synthetic Data Inversion**

As depicted in Figure 11, $D$ has a size of $0.15m \times 0.15m$ and it is uniformly discretized into $32 \times 32$ grids. 32 transmitters and receivers are evenly distributed on a circle outside $D$ of which radius is $1.67m$. The incident frequency is $3GHz$. Two cylinders are randomly located inside $D$ as summarized in
Table I. Figure 4 shows several examples of scatterer shapes. The data set is generated by MoM. Supervised and unsupervised NeuralBIM are trained and tested on the same data set. The data set comprises 20000 data samples of which 80% are for training and 20% for testing. The initial guess of the total field adopts the incident field. BP method is applied to generate the initial guess of the contrast.

1) Supervised Neural Born Iterative Method: Figure 5 plots the convergence curves of $\text{obj}_{su}$, $\text{loss}_{E}$, and $\text{loss}_{\chi}$. All of them drop steadily as the training progresses. It can be observed that $\text{loss}_{\chi}$ dominates $\text{loss}_{E}$ with the small value of $\text{loss}_{E}$. The testing $\text{loss}_{\chi}$ is a little higher than the training one which leads to overfitting. The overfitting can be alleviated by introducing better regularizations or stopping training earlier.

Figure 6 depicts the comparisons between ground truth and reconstructions that are randomly chosen in the testing data set. The inverted contrasts are in a good agreement with the ground truth as reflected in the absolute error distributions. The differences mostly lie on the boundaries of scatterer shapes.

The updated reconstructions of contrasts are illustrated in Figure 7. The update of contrast begins with the initial guess generated by BP method and it improves gradually with the increase of iterations.

The mean absolute error (MAE) histograms of all data samples are charted in Figure 8(a). The specific means and standard deviations (stds) are summarized in Table II. The MAE histogram of total fields has small means and stds, and this indicates the good reconstruction precisions. The MAE
TABLE II
MAE MEANS AND STANDARD DEVIATIONS OF SUPERVISED NEURALBIM

| Item                        | MAE-R* (mean/std) | MAE-I+ (mean/std) |
|-----------------------------|-------------------|-------------------|
| Contrast (train)¹          | 0.0118/0.0048     | 0.0133/0.0032     |
| Contrast (test)²           | 0.0135/0.0066     | 0.0155/0.0049     |
| Total field (train)³       | 0.0054/0.0013     | 0.0053/0.0012     |
| Total field (test)⁴        | 0.0055/0.0013     | 0.0054/0.0013     |

¹ contrasts in training data set  
² contrasts in testing data set  
³ total fields in training data set  
⁴ total fields in testing data set  
* MAE of real part  
+ MAE of imaginary part

Fig. 9. The final losses of supervised NeuralBIM trained with data sets of different sizes. Training and testing loss (all, chi, total E) denote obj_{su}, loss_{\chi}, and loss_{E,t}.

histogram of testing contrasts has higher mean and std than the one of training contrasts, which is consistent with the convergence curve in the Figure 5. Figure 8(b) charts a histogram of the reconstruction MAE of each cylinder at different contrast ranges for all data samples. The reconstruction MAE is only evaluated for the cylinders and the background is not considered. The real and imaginary part ranges of the contrast is evenly divided into five intervals respectively. For real parts, the five intervals are [0, 0.2], [0.2, 0.4], [0.4, 0.6], [0.6, 0.8], [0.8, 1.0], and they are [-2, -1.6], [-1.6, -1.2], [-1.2, -0.8], [-0.8, -0.4], [-0.4, 0] for imaginary parts. Supervised NeuralBIM shows a stable performance regarding a wide range of contrast values.

The impact of data set sizes on the performance of supervised NeuralBIM is first investigated. The data sets are assumed to have 5000, 10000, 15000, 20000 samples respectively, of which 80% and 20% are for training and testing. For all data sets, all training processes are assumed to stop at the 100th epoch. Figure 9 shows the final losses of supervised NeuralBIM trained with data sets of different sizes. It can be observed that the larger the training data set, the better the reconstruction performance. When the size of data set increases from 15000 to 20000, the testing loss_{\chi} is slightly reduced although the training loss_{\chi} has a noticeable drop. Therefore, considering the training time, the data set of 20000 samples is adopted to train NeuralBIM in this paper.

Fig. 10. Losses of contrasts and total fields reconstructed by supervised NeuralBIM under different noise levels. Chi and total E denote contrast and total field.

Fig. 11. Reconstructed scatterers of supervised NeuralBIM under different noise levels. From left to right are ground truth, reconstructions at the noise level of 0%, 10%, 20%, 30%, 40%, 50%.

Fig. 12. Verification of generalization ability of supervised NeuralBIM on different scatterer shapes.

Fig. 13. The CNN structure and the reconstructed scatterers in different contrasts. The CNN structure in the re-trained supervised NeuralBIM (left sub-panel) and the reconstructed scatterers in different contrasts (right sub-panel). From left to right, the contrasts are: 1.1 – j2.1, 1.2 – j2.2, 1.3 – j2.3, 1.4 – j2.4, 1.5 – j2.5.
The anti-noise capability is then validated. We randomly select 20 testing data samples and add different levels of Gaussian white noise. Figure 10 plots $\text{loss}_{\text{su}}$ and $\text{loss}_{\text{xi}}$ of each noise level. Both $\text{loss}_{\text{su}}$ and $\text{loss}_{\text{xi}}$ fluctuate within a very small range, which demonstrates the strong anti-noise capability of supervised NeuralBIM. The inverted contrasts are detailed in Figure 11. The reconstruction maintains a good and stable quality even at the noise level of 50%.

The generalization ability is also verified on different scatterer geometries unseen at the training time. Six different kinds of scatterer geometries are considered as shown in Figure 12. The comparisons between the ground truth and reconstructions are demonstrated in Figure 12. Supervised NeuralBIM can perform reliable inversion although the scatterer geometries are not included in the training data set. The discrepancy primarily lies in the tiny structures that are hard to distinguish.

Furthermore, supervised NeuralBIM is employed to invert the scatterers with higher contrasts. Supervised NeuralBIM with CNN structures as shown in Figure 3 is insensitive to the higher contrasts, which could be caused by that Tanh functions are saturating nonlinearities. The modified CNN model with ELU nonlinearities [50] is adopted as shown in Figure 13. The modified supervised NeuralBIM is re-trained under the same training settings. It demonstrates an improved performance on inverting scatterers on higher contrasts, as shown in Figure 13. The generalized performance is an improvement of supervised NeuralBIM. The inverted contrasts are generated by BP method. The candidate contrasts are continuously modified with the increase of iterations.

Supervised NeuralBIM is also tested whether it can be generalized to solve large-scale problems when trained for small-scale problems. Supervised NeuralBIM is trained on $32 \times 32$ grids. Then, it is applied to perform inversion of $64 \times 64$ grids. According to Eq. (16), in the $k$-th iteration, $\chi$-CNN is dependent on the concatenation of $\mathcal{R}_k^{E'}$ and $\chi_k$. Note that $\mathcal{R}_k^{E'}$ and $\chi_k$ have the same size and are concatenated in the channel dimension. The size of $\mathcal{R}_k^{E'}$ is determined by the number of transmitters and receivers. When the size of $\chi_k$ is $64 \times 64$, the number of transmitters and receivers need be set as 64. Such change of the measurement configuration results in a reduced reconstruction performance of Supervised NeuralBIM. Figure 14 demonstrates two different reconstructed scatterers of $64 \times 64$ grids. The inverted results can reflect the locations and shapes of scatterers but the corresponding contrast values are not accurate enough.

2) Unsupervised Neural Born Iterative Method: The convergence curves of unsupervised NeuralBIM are summarized in Figure 15. It is shown that $\text{obj}_{\text{unu}}$, $\text{res}_{\text{E}}$, and $\text{res}_{\chi}$ fall steadily as the training progresses. They fluctuate several times due to the unsupervised learning scheme. Since $\text{res}_{\text{E}}$ and $\text{res}_{\chi}$ cannot reflect the performance intuitively, the MSEs between the ground truth and the reconstructions are employed as the evaluation functions:

$$
\text{eval}_{E} = \frac{1}{N_{E'}} ||E'^* - E'||_F^2
$$

$$
\text{eval}_{\chi} = \frac{1}{N_{\chi'}} ||\chi^* - \chi'||_F^2.
$$

The convergence curves of $\text{eval}_{E}$ and $\text{eval}_{\chi}$ are also plotted in Figure 15. The curves of $\text{eval}_{E}$ and $\text{eval}_{\chi}$ have the same trends as the ones of $\text{obj}_{\text{unu}}$, which further validates the efficacy of the objective function based on the governing equations (Eq. (3) and Eq. (4)).

Figure 16 demonstrates two reconstructed contrasts that are randomly chosen in the testing data set. The inverted contrasts are in good agreement with the ground truth, and the absolute error distributions are in a low error level. The TV regularization enables the homogenous background and the boundaries of scatterers in the reconstructions.

Figure 17 demonstrates the updated reconstructions of contrasts during the alternate update process. The initial guesses of contrasts are generated by BP method. The candidate contrasts are continuously modified with the increase of iterations.

Figure 18(a) charts the MAE histograms of inverted contrasts and total fields in the training and testing data sets. The specific means and stds of MAE histograms are summarized in Table III. The real parts of inverted scatterers have better precisions than the imaginary parts. The MAE histograms of training and testing data have close means and stds, which is consistent with the convergence curve shown in Figure 15. Figure 18(b) plots the reconstruction MAE of each cylinder at different contrast ranges for all data samples. The contrast range is divided in the same way as Figure 18(b). It can be observed that the MAEs grow with the increase of absolute values of contrasts. In order to improve the performance on...
Fig. 16. Comparison between ground truth and scatterers inverted by unsupervised NeuralBIM. From left to right are: ground truth, reconstruction and their absolute error distribution.

Fig. 17. Updated reconstructions of contrasts in the iterative computing of unsupervised NeuralBIM. From left to right are: initial guess, reconstructions at the first, fifth and seventh iteration. In each sub-panel, the first and second row show the real and imaginary parts.

Fig. 18. MAE histograms of contrasts and total fields inverted by unsupervised NeuralBIM. (a) is a histogram of the reconstruction MAE of each data sample in training and testing data sets. (b) is a histogram of the reconstruction MAE of each cylinder at different contrast ranges for all data samples. Chi, Etot, Real, Imag denote the contrast, total field, real part and imaginary part in the legend.

### TABLE III

| Item                   | MAE-R$^*$ (mean/std) | MAE-I$^+$ (mean/std) |
|------------------------|----------------------|----------------------|
| Contrast (train)$^1$   | 0.0472/0.0248        | 0.0745/0.0341        |
| Contrast (test)$^2$    | 0.0465/0.0239        | 0.0740/0.0337        |
| Total field (train)$^3$| 0.00047/0.0014       | 0.0018/0.0014        |
| Total field (test)$^4$ | 0.000047/0.0013      | 0.00018/0.0014       |

1 contrasts in training data set
2 contrasts in testing data set
3 total fields in training data set
4 total fields in testing data set
* MAE of real part
$^+$ MAE of imaginary part

the cases of high contrasts, we can increase the proportion of high contrast samples in the training data set.

The performance of unsupervised NeuralBIM is then evaluated under different noise levels of scattered fields. A total of 20 data samples are selected from the testing data set and the Gaussian white noise is added to the field data for validation. The curves of $\text{eval}_E$ and $\text{eval}_\chi$ are plotted in Figure 19. It can be observed that both $\text{eval}_E$ and $\text{eval}_\chi$ are

Fig. 19. Losses of contrasts and total fields reconstructed by unsupervised NeuralBIM under different noise levels. Chi and total E denote contrast and total field.
Fig. 20. Reconstructed scatterers of unsupervised NeuralBIM under different noise levels. From left to right are ground truth, reconstructions at the noise level of 0%, 10%, 20%, 30%, 40%, 50%.

Fig. 21. Verification of generalization ability of unsupervised NeuralBIM on different scatterer shapes.

stable under different noise levels. The anti-noise capability of unsupervised NeuralBIM is verified. The inverted scatterers at the noise level from 0% to 50% are compared in Figure 20. The inverted scatterers demonstrates a good and stable image quality.

The generalization ability on scatterer shapes is then verified. The shapes of scatterers are totally different from the training ones. The corresponding reconstructions are demonstrated in Figure 21. The inverted scatterers are in good agreement with the ground truth with clear boundaries.

B. Comparison With TBIM

The performances of TBIM, supervised and unsupervised NeuralBIM are compared in this section. NeuralBIM is assumed to stop at the 7-th iteration and TBIM converges at the 14-th iteration. The TBIM is computed on Intel(R) Core(TM) i5-9600K CPU @ 3.70GHz. Besides, the initial guess of TBIM is also generated by BP method. The testing example assumes that three cylinders are located in \( D \). The real and imaginary parts of their contrasts are in [0, 1] and \([-2,0]\) respectively. Note that the testing example is unseen during the training of both supervised and unsupervised NeuralBIM. Figure 22 illustrates the comparison between reconstructions of TBIM, supervised and unsupervised NeuralBIM. TBIM can locate the positions of three cylinders, but it cannot capture the accurate shapes of scatterers with some artifacts. Supervised NeuralBIM demonstrates the best performance. The inverted result is high-resolution and the difference from the ground truth primarily lies on the boundaries of cylinders. Unsupervised NeuralBIM can also locate the cylinders and provide better shapes of scatterers than TBIM. The detailed comparison of performance is summarized in Table IV. Although supervised and unsupervised NeuralBIM needs approximately 41 hours and 45 hours for offline training respectively, they demonstrates a significant reduction in online computing time compared to TBIM. The MAEs between ground truth and reconstructions of these methods further validate the aforementioned observations derived from Figure 22.

C. Experimental Data Inversion

In this section, we validate unsupervised NeuralBIM with experimental data inversion. The experimental data is published by Institut Fresnel [51]. The TM polarized measured data of \( \text{FoamDielInt} \) and \( \text{FoamDielExt} \) models are selected for inversion. The frequency of measured data is 3 GHz. The measurement is performed with 8 transmitters and 241 receivers in [51]. The measurement configuration is different

| Cylinder | \( \text{Radius} \) | \( \text{Contrast (Real)} \) | \( \text{Contrast (Imag)} \) |
|----------|-----------------|------------------|------------------|
| A        | 0.03-0.045m     | 0.2-0.8          | 0                |
| B        | 0.01-0.02m      | 1.5-2.3          | 0                |
from the synthetic one. The Green’s functions in Eq. (3) and Eq. (4) are changed as a result. Because the Green’s functions are incorporated, NeuralBIM needs to be re-trained when the measurement configuration changes.

The training configuration has 8 transmitters and downsamples 241 receivers to 128 receivers. The training data set of 16000 samples is generated by assuming two cylinders are randomly located inside the DoI. Table V summarizes the geometries and material properties of cylinders. The measured data takes the size of \(8 \times 32 \times 32\) and they are reshaped into \(8 \times 32 \times 32\) for training.

After the training is finished, unsupervised NeuralBIM is employed to invert the measurement data of FoamDielInt and FoamDielExt models. Figure 23 shows the reconstructions and the corresponding inverted scattered fields. The reconstructed geometries and material properties of scatterers agree well with the ground truth. Then the scattered fields are calculated based on the reconstructed results. The inverted scattered fields also show a good agreement with the measured ones, which further validate unsupervised NeuralBIM.

VI. CONCLUSIONS

In this paper, neural Born iterative method is proposed to solve 2D inverse scattering problems. Inspired by PhiSRL, NeuralBIM emulates the computational process of TBIM by applying CNNs to learn the parametric functions of alternate update rules. Both supervised and unsupervised learning schemes of NeuralBIM are presented in this paper, and they all demonstrate better performance than TBIM. Numerical and experimental results validate the efficacy of the proposed NeuralBIM. The anti-noise capability and generalization ability of NeuralBIM are also verified.

This paper investigates a potential way to incorporate deep learning techniques into traditional computational electromagnetic algorithms. Additionally, the feasibility is also verified that the physics-incorporated deep neural network can be trained via an unsupervised process to perform reliable computations by applying governing equations as its objective function.

REFERENCES

[1] X. Chen, *Computational methods for electromagnetic inverse scattering*. John Wiley & Sons, 2018.
[2] M. Salucci, N. Anselmi, G. Oliveri, P. Calmon, R. Morelli, C. Reboud, and A. Massa, “Real-time NDT-NDE through an innovative adaptive partial least squares SVR inversion approach,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 11, pp. 6818–6832, 2016.
[3] A. Abubakar, P. M. Van den Berg, and J. J. Mallorqui, “Imaging of biomedical data using a multiplicative regularized contrast source inversion method,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 50, no. 7, pp. 1761–1771, 2002.
[4] M. Pastorino and A. Randazzo, *Microwave Imaging Methods and Applications*. Artech House, 2018.
[5] A. Abubakar, T. Habashy, V. Druskin, L. Knizhnerman, and D. Alumbaugh, “2.5 D forward and inverse modeling for interpreting low-frequency electromagnetic measurements,” *Geophysics*, vol. 73, no. 4, pp. F165–F177, 2008.
[6] M. Salucci, L. Poli, N. Anselmi, and A. Massa, “Multiparticle frequency swarm optimization for enhanced multiresolution GPR microwave imaging,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 3, pp. 1305–1317, 2016.
[7] M. Slaney, A. C. Kak, and L. E. Larson, “Limitations of imaging with first-order diffraction tomography,” *IEEE transactions on microwave theory and techniques*, vol. 32, no. 8, pp. 860–874, 1984.
[8] A. Devaney, “Inverse-scattering theory within the Rytov approximation,” *Optics letters*, vol. 6, no. 8, pp. 374–376, 1981.
[9] K. Belkebir, P. C. Chaumet, and A. Sentenac, “Superresolution in total internal reflection tomography,” *JOSA A*, vol. 22, no. 9, pp. 1889–1897, 2005.
[10] Y. Wang and W. C. Chew, “An iterative solution of the two-dimensional electromagnetic inverse scattering problem,” *International Journal of Imaging Systems and Technology*, vol. 1, no. 1, pp. 100–108, 1989.
[11] W. C. Chew and Y.-M. Wang, “Reconstruction of two-dimensional permittivity distribution using the distorted Born iterative method,” *IEEE transactions on medical imaging*, vol. 9, no. 2, pp. 218–225, 1990.
[12] P. Van den Berg and A. Abubakar, “Contrast source inversion method: State of art,” *Progress in Electromagnetics Research*, vol. 34, pp. 189–218, 2001.
[13] A. Zakaria, C. Gilmore, and J. LoVerdi, “Finite-element contrast source inversion method for microwave imaging,” *Inverse Problems*, vol. 26, no. 11, p. 115010, 2010.
[14] X. Chen, “Subspace-based optimization method for solving inverse-scattering problems,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 48, no. 1, pp. 42–49, 2009.
[15] O. Dorn and D. Lesselier, “Level set methods for inverse scattering,” *Inverse Problems*, vol. 22, no. 4, p. R67, 2006.
[16] P. Mojabi and J. LoVerdi, “Overview and classification of some regularization techniques for the Gauss-Newton inversion method applied to inverse scattering problems,” *IEEE Transactions on Antennas and Propagation*, vol. 57, no. 9, pp. 2658–2665, 2009.
