East Asian Journal on Applied Mathematics
doi: 10.4208/eajam.2025-049.300725

Vol. **xx**, No. x, pp. 1-33
xxx 2026

Reconstructing Electromagnetic Parameters Based on Convolutional Neural Network for the Groove Scattering with Impedance Boundary Condition

Meiling Zhao[1,2], Yuchan Yang[1], Baoyi Jin[1] and Liqun Wang[3,4,*]

[1]*Department of Mathematics and Physics, North China Electric Power University, Baoding 071003, China.*
[2]*Hebei Key Laboratory of Physics and Energy Technology, North China Electric Power University, Baoding 071003, China.*
[3]*Department of Mathematics, College of Science, China University of Petroleum, Beijing 102249, China.*
[4]*Beijing Key Laboratory of Optical Detection Technology for Oil and Gas, China University of Petroleum, Beijing 102249, China.*

Received 6 March 2025; Accepted (in revised version) 30 July 2025.

Abstract. The reconstruction of electromagnetic parameters of the groove scattering is widely applied in the military and engineering fields. However, the inherent nonlinearity and ill-posed nature of the problem bring a big challenge to reconstruction. To over the difficulty, we develop a real-valued convolutional neural network (RV-CNN) and a complex-valued convolutional neural network (CV-CNN) to reconstruct the electromagnetic parameters of the open groove with impedance boundary. First the scattered field data of the open groove is obtained using the Petrov-Galerkin finite element interface method. Then, the RV-CNN separately extracts the magnitude and phase of the scattered field data, and introduces them as independent input channels into the network architecture, with the ELU activation function selected to align with the characteristics of the phase data. In contrast, the CV-CNN directly takes the complex-valued scattered field data as input channels and performs optimization derivation based on a complex-valued loss function. This approach not only accelerates the convergence speed of the model but also enhances the overall reconstruction capability. Numerical experimental results demonstrate that both RV-CNN and CV-CNN can achieve accurate reconstruction results with a smaller volume of training data and reflect good generalization performance for the reconstruction of electromagnetic parameters of the open groove filled with homogeneous and inhomogeneous media under impedance boundary.

AMS subject classifications: 65N21, 78A45

Key words: Convolutional neural network, groove with impedance boundary, electromagnetic parameter reconstruction, finite element methods.

*Corresponding author. *Email addresses:* wliqunhmily@cup.edu.cn (L. Wang), meilingzhao@ncepu.edu.cn (M. Zhao), 220232217017@ncepu.edu.cn (Y. Yang), 220212217032@ncepu.edu.cn (B. Jin)

1. Introduction

The problem of reconstruction of electromagnetic parameters of open grooves refers to the process in which one or more columns of electromagnetic waves irradiate open grooves and produce scattering, and the physical properties of the open grooves are investigated by means of the measuring the scattered fields of the grooves. Open grooves widely exist in military targets such as aircraft air intakes, tail vents and holes in metal surfaces, which are usually strong scattering sources. Regarding the electromagnetic scattering problems of cavities with various configurations, some research achievements have been made, including open rectangular grooves with large wave numbers [33], multiple grooves [12], and arbitrarily shaped grooves filled with inhomogeneous and anisotropic media [34]. To reduce the radar cross-section (RCS) of aircraft, it is necessary to coat its surface with wave-absorbing materials. When the absorbing material is very thin, it can exhibit characteristics of impedance boundary conditions, allowing the electromagnetic waves to effectively match the surface of the material, thereby reducing reflection and enhancing absorption. Consequently, the reconstruction of the electromagnetic parameters of open slots has attracted increasing attention from researchers. However, the obvious diffraction and multiple scattering effects of electromagnetic waves cause strong nonlinear characteristics and ill-posed property.

To overcome these difficulties, traditional methods such as the linear sampling method, integral equation method, Newton iteration method, factorization method, and so on, have been studied by many scholars. Tan and Zhu [22] proposed two numerical optimization algorithms based on distributed shape gradients to address the shape reconstruction problem with nonlinear elliptic interface constraints. The mathematical inversion algorithm enables the reconstruction of both the cavity geometry and boundary impedance [8] and the regularized Newton iteration method has been applied to the boundary reconstruction task [13]. For the inverse scattering problems of obstacles and the internal inverse scattering problems of cavities, the improved sampling methods have been proven to be effective in handling them [14]. Additionally, the linear sampling method demonstrates applicability to the inverse scattering problem of three-dimensional inhomogeneous medium cavities [29]. Meng *et al.* [16] transformed the inverse problem into a nonlinear undefined system of equations to probe the geometry of the groove by measuring the internal curves of the penetrable groove using the regularized Newton iterative method. In addition to the aforementioned methods, the factorization method can also be employed for the reconstruction of cracks outside penetrable cavities with mixed boundary conditions [7], and its effectiveness in reconstructing internal interfaces has been proved in [20]. Traditional numerical methods such as the linear sampling method and factorization method can obtain numerical solutions with low time and space complexity through fast solution techniques for specific integral equations, featuring high computational efficiency. However, their effectiveness highly depends on full-view observation data, which often poses significant challenges in practical engineering applications.

Neural networks with regression features have obtained good results in the research of denoising, anti-convolution, interpolation and other problems, which make up for the short-

comings of traditional methods and reduce the computational complexity, and are gradually applied to solve the electromagnetic inverse scattering. These networks can be broadly categorized as ordinary deep neural networks, convolutional neural networks (CNNs) and their derivative models, and personalized customized networks.

Wang *et al.* [24] applied a deep neural network (DNN) to solve the inverse scattering problem for cylindrical media and was able to deal with highly oscillatory scattering patterns to a certain extent. Li *et al.* [11] dealt with a highly nonlinear electromagnetic inverse scattering problem in different frequency bands involving large-scale and high-contrast objects by putting the complex data in scattered field into a cascade model of DNNs and a multilayered complex-valued residual convolutional neural network. Tian *et al.* [23] applied CNNs to reconstruct the relative dielectric constants and positional information of lossy media. Gao and Zhang [5] combined CNNs with sampling imaging schemes for reconstruction under finite apertural networks or phase-less far-field data. Puzyrev [19] used a deep fully convolutional neural network to highly accurately estimate subsurface resistivity models. Further, CNN-derived models have also been applied in the field of electromagnetic backscattering. For example, the U-Net network constructed by adding data compression units on top of CNNs can reconstruct the dielectric constant image by retaining the features of both high-dimensional data and low-dimensional data [26]. End-to-end scalable cascaded convolutional neural networks (SC-CNNs) can obtain high-resolution images directly from the scattered field [27]. Novel hybrid dilated convolutional neural network (HDCNN) can quickly reconstruct electromagnetic parameters [25]. For the shape of the scatterer, the fully connected neural network (FCNN) proposed by Gao [4] can be used for reconstruction. Zhang *et al.* [30] utilizes complex convolutional neural networks to accomplish the classification of synthetic aperture radar (SAR) images. The AlexNet model based on CNN can perform autonomous learning operations on the features of SAR images [6]. Furthermore, the first to incorporate physical factors into a deep learning framework was Yin and Yan [28], who proposed the deep decomposition method (DDM). As the first physics-aware machine learning approach for solving the inverse obstacle scattering problem with limited aperture, DDM achieves promising reconstruction performance even when both the incident and observation apertures are extremely restricted. Although the contrast source network proposed by SwitchNet [21] can solve the electromagnetic inverse scattering problem, its training process still remains data-driven at the core. However, Ning *et al.* [17, 18] organically combine the direct sampling method (DSM) with deep learning techniques (DSM-DL), where the U-Net neural network is utilized to process multi-channel inputs, enabling high-quality reconstruction of scatterers from a data-driven perspective.

However, deep neural networks have not emerged significant advantages in the electromagnetic inverse scattering of open grooves. In this paper, two deep neural network architectures based on convolutional neural networks are proposed for reconstructing electromagnetic parameters of open grooves with impedance boundary. RV-CNN reconstructs the real-valued dielectric constant by processing the complex-valued data of the scattering field into real-valued data and selecting the activation function individually according to the data characteristics, while CV-CNN directly inputs the complex-valued data into the neural network and extends the activation function, the loss function, and the back-propagation

algorithm from the real-valued case to the complex-valued case for reconstructing the complex-valued parameters. The scattered field data set of the impedance boundary groove is obtained using the Petrov-Galerkin finite element interface method. The effect of the proposed networks is validated by means of the experimental results, which show that both RV-CNN and CV-CNN network architectures are capable of highly accurate reconstruction of the relative permittivity of the media in open grooves with impedance boundary using less training data. The proposed networks are demonstrated to have good generalization performance for reconstructions of the electromagnetic parameters of the groove scattering filled with homogeneous and inhomogeneous media with impedance boundary.

The paper is organized as follows. Section 2 describes the geometry model and scattering problem of open grooves with impedance boundary. In Section 3, we give the synthesis method for CNN input data and develop the real-valued and complex-valued convolutional neural network to reconstruct the electromagnetic parameters. Numerical experiments verify the reconstruction and generalization ability of the networks, experimental results and model evaluation metrics in Section 4. Section 5 summarizes the main conclusions of this paper.

2. Scattering from Open Grooves with Impedance Boundary

In this section, we derive the governing equation for scattering by overfilled cavities with inhomogeneous anisotropic media in this section. Before investigating the reconstruction of electromagnetic parameters in open grooves, it is necessary to obtain the scattered field data for the groove. In this paper, we employ the Petrov-Galerkin finite element interface method to solve the groove scattering problem with impedance boundary conditions and present a specific method for synthesizing the scattered field data.

2.1. Groove with impedance boundary model

We consider a time-harmonic electromagnetic plane wave incident on an open groove embedded in the impedance ground plane as shown in Fig. 1.

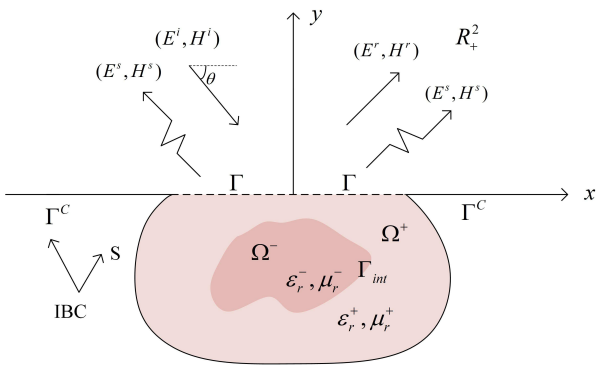


Figure 1: Groove geometry.

The groove is filled with inhomogeneous isotropic or anisotropic media. A bound region is denoted as Ω, which can be divided into two sub-regions Ω^+ and Ω^-, Γ_{int} represents the interface of the internal media. S, Γ and Γ^C are defined as the wall of groove, the aperture of groove and the ground plane outside the aperture, respectively. We assume $\theta \in [0,\pi]$, with free space $\mathbb{R}^2_+ = \{(x,y) \in \mathbb{R}^2 |\ y > 0\}$, which is the region above the ground plane.

Considering a time-harmonic field with angular frequency ω and time dependence of $e^{-\mathrm{i}\omega t}$. By combining Maxwell's equations with the constitutive equation

$$\nabla \times \mathbf{E} = -\mu_0 \boldsymbol{\mu}_r \frac{\partial \mathbf{H}}{\partial t}, \quad \nabla \times \mathbf{H} = \varepsilon_0 \boldsymbol{\varepsilon}_r \frac{\partial \mathbf{E}}{\partial t},$$

where the relative permittivity and relative permeability are respectively

$$\boldsymbol{\varepsilon}_r = \begin{cases} \boldsymbol{\varepsilon}_r^+ & \text{in } \Omega^+, \\ \boldsymbol{\varepsilon}_r^- & \text{in } \Omega^-, \end{cases} \quad \boldsymbol{\mu}_r = \begin{cases} \boldsymbol{\mu}_r^+ & \text{in } \Omega^+, \\ \boldsymbol{\mu}_r^- & \text{in } \Omega^-. \end{cases}$$

The dielectric constant of free space $\varepsilon_0 = 1/(36\pi) \times 10^{-9} F/m$ and magnetic permeability $\mu_0 = 4\pi \times 10^{-7} H/m$. The electric field $\mathbf{E} = (E_x, E_y, E_z)$ and the magnetic field $\mathbf{H} = (H_x, H_y, H_z)$ are represented in component form. If the groove is filled with inhomogeneous anisotropic media, the relative permittivity $\boldsymbol{\varepsilon}_r$ and relative permeability $\boldsymbol{\mu}_r$ are expressed in tensor form as follows:

$$\boldsymbol{\varepsilon}_r^{\pm} = \begin{pmatrix} \varepsilon_{11}^{\pm} & \varepsilon_{12}^{\pm} & 0 \\ \varepsilon_{21}^{\pm} & \varepsilon_{22}^{\pm} & 0 \\ 0 & 0 & \varepsilon_{33}^{\pm} \end{pmatrix}, \quad \boldsymbol{\mu}_r^{\pm} = \begin{pmatrix} \mu_{11}^{\pm} & \mu_{12}^{\pm} & 0 \\ \mu_{21}^{\pm} & \mu_{22}^{\pm} & 0 \\ 0 & 0 & \mu_{33}^{\pm} \end{pmatrix}.$$

In two-dimensional cases, $\mathbf{E} = \hat{z} E_z$ or $\mathbf{H} = \hat{z} H_z$, the impedance boundary condition can be expressed as follows:

$$E_z = \eta Z_0 (n_x H_y - n_y H_x), \quad H_z = -\frac{1}{\eta Z_0}(n_x E_y - n_y E_x). \tag{2.1}$$

Typically, the first case is referred to as transverse magnetic (TM) polarization, and the second is referred to as transverse electric (TE) polarization. In the TM case, $\mathbf{H}$ is perpendicular to the z-axis, and $\mathbf{E}$ is parallel to z-axis, denoted as $\mathbf{E} = \hat{z} E_z$, $\mathbf{H} = \hat{x} H_x + \hat{y} H_y$.

We consider the time-harmonic field $e^{\mathrm{i}\omega t}$. According to Maxwell's equations, we have

$$\begin{aligned} \frac{\partial E_z}{\partial y} &= -\mathrm{i}\omega\mu_0\big(\mu_{11}^{\pm} H_x + \mu_{12}^{\pm} H_y\big), \\ \frac{\partial E_z}{\partial x} &= \mathrm{i}\omega\mu_0\big(\mu_{21}^{\pm} H_x + \mu_{22}^{\pm} H_y\big), \\ \frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} &= \mathrm{i}\omega\varepsilon_0 \varepsilon_{33}^{\pm} E_z. \end{aligned} \tag{2.2}$$

Combining Eqs. (2.1) and (2.2), and letting u denote the third component of the vector $\mathbf{E}$, i.e. $u = E_z$, the impedance problem can be expressed in the following form [3]:

$$\begin{aligned} &\nabla\cdot(\boldsymbol{\gamma}_{TM}\nabla u)+\boldsymbol{k}_{TM}^2 u=0 \quad \text{in } \Omega\cup R_+^2,\\ &\mathbf{n}\cdot(\boldsymbol{\gamma}_{TM}^+\nabla u)-\sigma u=0 \qquad \text{in } S\cup\Gamma^c, \end{aligned}$$

where

$$\boldsymbol{\gamma}_{TM}^{\pm}=\begin{pmatrix}\mu_{11}^{\pm} & \mu_{21}^{\pm}\\ \mu_{12}^{\pm} & \mu_{22}^{\pm}\end{pmatrix}\Big/\begin{vmatrix}\mu_{11}^{\pm} & \mu_{21}^{\pm}\\ \mu_{12}^{\pm} & \mu_{22}^{\pm}\end{vmatrix},\quad \boldsymbol{\gamma}_{TM}=\begin{cases}\boldsymbol{\gamma}_{TM}^+ & \text{in } \Omega^+,\\ \boldsymbol{\gamma}_{TM}^- & \text{in } \Omega^-,\end{cases}\quad \boldsymbol{k}_{TM}=\begin{cases}\boldsymbol{k}_{TM}^+ & \text{in } \Omega^+,\\ \boldsymbol{k}_{TM}^- & \text{in } \Omega^-,\end{cases}$$

and $\boldsymbol{k}_{TM}^{\pm}=\omega^2\varepsilon_0\mu_0\varepsilon_{33}^{\pm}$. Additionally, $\sigma=\mathrm{i}k_0/\eta$ is the impedance coefficient, where $k_0=\omega\sqrt{\varepsilon_0\mu_0}$ is the wave number in free space.

We define the scattering field $u_{sca}=u-u_{inc}-u_{ref}$, u_{inc} and u_{ref} are the incident and reflected electric fields, respectively. Let

$$u_{inc}=e^{\mathrm{i}k_0(x\cos\vartheta-y\sin\vartheta)},\quad u_{ref}=Re^{\mathrm{i}k_0(x\cos\vartheta+y\sin\vartheta)},$$

where the reflection coefficient $R=-(\sigma-\mathrm{i}k_0\sin\vartheta)/(\sigma+\mathrm{i}k_0\sin\vartheta)$. Hence, u_{sca} satisfies the following system:

$$\begin{aligned} &\Delta u_{sca}+k_0^2u_{sca}=0 \qquad \text{in } R_+^2,\\ &\mathbf{n}\cdot\nabla u_{sca}-\sigma u_{sca}=0 \quad \text{on } \Gamma^c,\\ &u_{sca}=u-g \qquad\qquad \text{on } \Gamma, \end{aligned}$$

where $g=u_{inc}+u_{ref}$. Additionally, the scattering filed satisfies the Sommerfeld radiation condition

$$\lim_{r\to\infty}\sqrt{r}\left(\frac{\partial u_{sca}}{\partial r}-\mathrm{i}k_0u_{sca}\right)=0.$$

By using the impedance Green's function, the impedance-to-Dirichlet condition on the aperture surface can be obtained

$$u(\mathbf{x})=g(\mathbf{x})-\int_\Gamma G_\sigma\left(\mathbf{x}',\mathbf{x}\right)\left[\mathbf{n}\cdot\left(\gamma_{TM}^+\nabla u(\mathbf{x}')\right)+\sigma u(\mathbf{x}')\right]d\mathbf{x}',\quad \mathbf{x}\in\Gamma,$$

thus transforming the unbounded domain problem into a problem confined within the groove [31,32]. The problem reduce to the following equations:

$$\begin{aligned} &\nabla\cdot\left(\boldsymbol{\gamma}_{TM}^+\nabla u\right)+{\boldsymbol{k}_{TM}^+}^2u=0 && \text{in } \Omega^+,\\ &\nabla\cdot\left(\boldsymbol{\gamma}_{TM}^-\nabla u\right)+{\boldsymbol{k}_{TM}^-}^2u=0 && \text{on } \Omega^-,\\ &\mathbf{n}\cdot\left(\boldsymbol{\gamma}_{TM}^+\nabla u\right)-\sigma u=0 && \text{on } S,\\ &u(\mathbf{x})=g(\mathbf{x})-\int_\Gamma G_\sigma\left(\mathbf{x}',\mathbf{x}\right)\left[\mathbf{n}\cdot\left(\gamma_{TM}^+\nabla u(\mathbf{x}')\right)+\sigma u(\mathbf{x}')\right]d\mathbf{x}' && \text{on } \Gamma,\\ &u|_{\Gamma_{int}^+}=u|_{\Gamma_{int}^-} && \text{on } \Gamma_{int},\\ &\mathbf{n}\cdot\left(\boldsymbol{\gamma}_{TM}^+\nabla u\right)|_{\Gamma_{int}^+}=\mathbf{n}\cdot\left(\boldsymbol{\gamma}_{TM}^-\nabla u\right)|_{\Gamma_{int}^-} && \text{on } \Gamma_{int}, \end{aligned}$$

where

$$G_\sigma\left(\mathbf{x},\mathbf{x}'\right)=\frac{1}{4\pi}\int_{-\infty}^{\infty}e^{-\sqrt{\xi^2-k_0^2}|y'-y|}\frac{e^{\mathrm{i}(x'-x)\xi}}{\sqrt{\xi^2-k_0^2}}d\xi$$
$$-\frac{1}{4\pi}\int_{-\infty}^{\infty}\frac{\sigma+\sqrt{\xi^2-k_0^2}}{\sigma-\sqrt{\xi^2-k_0^2}}e^{-\sqrt{\xi^2-k_0^2}(y'+y)}\frac{e^{\mathrm{i}(x'-x)\xi}}{\sqrt{\xi^2-k_0^2}}d\xi.$$

To solve the variational formulation of the above equations, two different Sobolev spaces are defined as follows:

$$\mathscr{H}^1(\Omega)=\left\{u\in L^2(\Omega)\;\middle|\;\iint_\Omega\left[u^2+\left(\frac{\partial u}{\partial x}\right)^2+\left(\frac{\partial u}{\partial y}\right)^2\right]dxdy<\infty\right\},$$
$$\mathscr{H}_0^1(\Omega)=\left\{v\;\middle|\;v\in\mathscr{H}_0^1(\Omega),\;v|_{S\cup\Gamma}=0\right\},$$

where v is the test function. The impedance problem takes the form: For any $v\in\mathscr{H}_0^1(\Omega)$, find u such that

$$\iint_{\Omega^+}\gamma_{TM}^+\nabla u\cdot\nabla v dxdy+\iint_{\Omega^-}\gamma_{TM}^-\nabla u\cdot\nabla v dxdy-\int_\Gamma\mathbf{n}\cdot\gamma_{TM}^+\nabla uvdx-\int_S\sigma uvdl$$
$$+\int_{\Gamma_{int}}\left(\mathbf{n}\cdot\gamma_{TM}^+\nabla u-\mathbf{n}\cdot\gamma_{TM}^-\nabla u\right)vdl-\iint_{\Omega^+}k_{TM}^+uvdxdy-\iint_{\Omega^-}k_{TM}^-uvdxdy=0.$$

The method for solving the polarization of TE is similar to that of TM.

2.2. Petrov-Galerkin finite element interface method without body fitting

The groove is enclosed within a conceptual rectangular box with vertices at $(x^{\min},y^{\min})$, $(x^{\max},y^{\min})$, $(x^{\min},y^{\max})$ and $(x^{\max},y^{\max})$. This rectangle is partitioned into an $M\times N$ Cartesian grid, where each rectangular unit has a length of $\Delta x=(x^{\max}-x^{\min})/M$ and a width of $\Delta y=(y^{\max}-y^{\min})/N$. Each rectangular unit is further divided into two triangular elements by connecting its top-left and bottom-right vertices. These triangles serve as the fundamental computational units. When solving grooves of arbitrary shapes, employing a consistent and uniform mesh division can significantly reduce computational complexity and improve accuracy.

To determine the positional relationship between the nodes and the groove boundary, the level set function $\phi_b(x,y)$ is introduced, and relevant information for each triangular element is recorded. We focus on the condition $\phi_b(x_i,y_j)<0$, which indicates that the point lies inside the groove. When the boundary intersects a triangular elements, three cases can be identified, as shown in Fig. 2.

Class 1. The entire triangular unit is contained within the groove.

Class 2. The boundary Γ intersects one vertex and one side of the triangular unit.

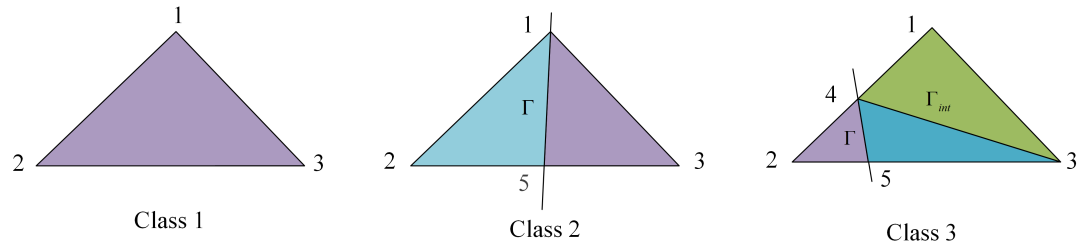


Figure 2: Three intersecting situations.

Class 3. The boundary Γ intersects two sides of the triangular unit. If the associated rectangular element is inside the groove, it will be divided into two triangular units.

Finally, the integration is performed element by element. By assembling the element stiffness matrices and element load vector, the global stiffness matrix and global load vector are constructed and using Matlab program. The scattered field data for the open groove, filled with either homogeneous or inhomogeneous media under an impedance boundary condition, is then obtained.

3. CNN Algorithms for Electromagnetic Parameter Reconstruction

Depending on the different cases of the filled media in open grooves with impedance boundaries, we propose two distinct CNN architectures to reconstruct the relative permittivity of media within a groove under impedance boundary conditions: a real-valued convolutional neural network and a complex-valued convolutional neural network. The following provides an overview of the data processing procedure and a detailed description of the network models employed.

3.1. Data preparation

Let AN, SN, and DS denote the number of incident angles of plane incident waves, the number of observation points in the groove aperture, and the sample size, respectively. In this paper, multiple field data are obtained by varying AN different incident angles. From these, SN observation points on the groove aperture surface are selected and combined to form a 3D complex tensor with dimensions $DS \times AN \times SN$. The output tensor has dimensions $DS \times C$, where C represents the number of parameters related to the relative permittivity.

For the RV-CNN, the synthetic 3D tensor is combined to form a 4D tensor of size $DS \times AN \times SN \times 2$ as shown in Fig. 3. The last dimension of the 4D tensor represents the number of channels, and the two channels are the magnitude and phase of the complex data. Accordingly, the input data should be represented as a tensor with dimensions $AN \times DS \times 2$.

For the CV-CNN, the synthesized 3D tensor is combined to form a 4D tensor with dimensions $DS \times AN \times SN \times 1$, as shown in Fig. 4. The value of 1 indicates that the complex data is fed directly into the CV-CNN without any processing, and the number of input channels is 1. Therefore, the input data is represented as a tensor with dimensions $AN \times DS \times 1$.

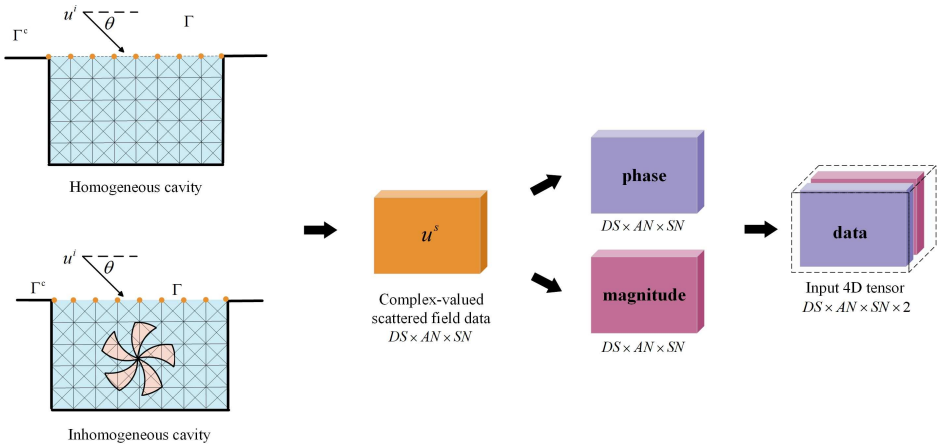


Figure 3: The data synthesis process of RV-CNN.

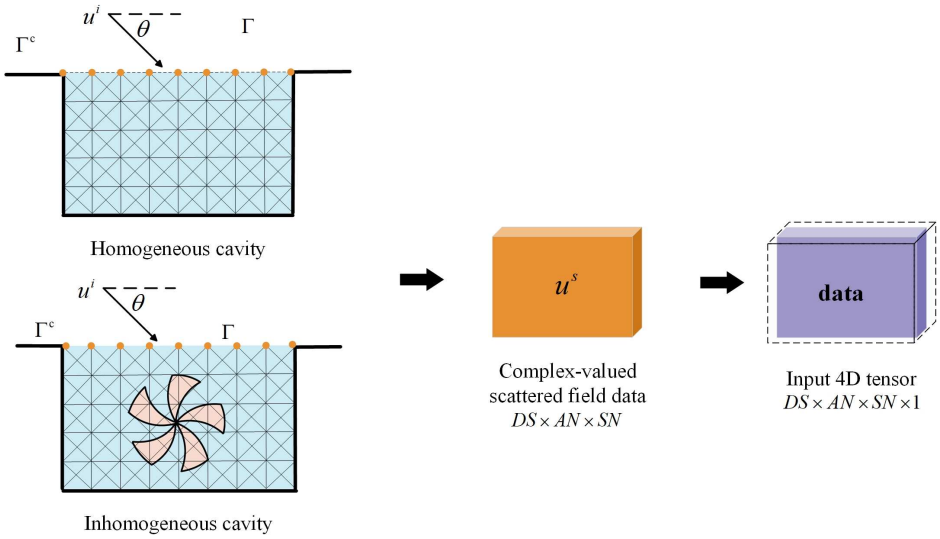


Figure 4: The data synthesis process of CV-CNN.

3.2. Real-valued convolutional neural network model

The convolutional neural network optimizes the weight parameters and biases through supervised learning so that the network output will match the given targets. The feature extraction process of the network is carried out alternately applying convolutional and pooling layers, along with nonlinear activation functions. The above process reflects the typical features of CNNs, such as local connectivity, weight sharing, pooling, and layer association levels [10]. The architectural details of the RV-CNN and the backpropagation process are outlined below.

3.2.1. The RV-CNN framework

First of all, the convolutional layer is the most crucial network layer structure in the convolutional neural network. The convolutional layer operates by a set of filters with shared weights that move smoothly over the feature map, performing calculations to extract and fuse information from the input feature map layers. The real-valued convolution operation can be expressed as

$$z_{k,l+1} = f_a\left(\sum_{i=1}^{n_l} w_{i,k,l+1} * z_{ikl} + b_{k,l+1}\right),$$

where $f_a(\cdot)$ denotes the real-valued activation function. n_l represents the number of inputs to the neurons in the $(l+1)$-th layer, and $w_{ik,l+1}$ denotes the i-th weight coefficient of the k-th neuron in layer $l+1$. z_{ikl} means the i-th output of the k-th neuron in the l-th layer. $*$ is the real-valued convolution operation. $b_{k,l+1}$ represents the k-th bias in the $(l+1)$-th layer, and $l \in \{1,\ldots,L-1\}$. $k = 1,\ldots,n_l$ is the number of neural network layers. In this study, the ELU activation function is selected, as shown below

$$f_a(x) = \begin{cases} x, & x > 0, \\ \eta(e^x - 1), & x \le 0, \end{cases}$$

where η denotes the learning rate, and it is taken as 1 by default in keras.

Subsequently, the pooling layer performs secondary feature extraction on the data output by the convolutional layer. The pooling layer independently computes summary statistics for local patches on each feature map in the convolutional layer. Therefore, the pooling layer reduces the feature dimension and decrease the number of training parameters in the neural network, while maintaining the same number of feature maps as in the convolutional layer. And the local translation invariance of the pooling layer is a very useful property when considering the features themselves rather than their locations. The max pooling and average pooling function are commonly used for real-valued pooling operation. We adopt the average pooling function since it comprehensively retains the information of all elements in local patches

$$z_{l+1} = f_p(z_l).$$

After feeding the data into the network input layer, we initialize the weight parameters of the network's for subsequent efficient training of the data by means of batch normalization, which can reduce the data distribution variation between different layers, and accelerate the convergence speed of the real-valued deep network. The process of batch normalization for the real-valued case is outlined below.

Suppose there are m real-valued data points in one batch, denoted as $\{z^1,\ldots,z^m\}$. The normalization process can then be expressed as follows:

$$\tilde{z}_i = \left(z - \frac{1}{m}\sum_{i=1}^{m} z_i\right)\left(\sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(z_i - \frac{1}{m}\sum_{i=1}^{m} z_i\right)^2 + \varepsilon}\right)^{-1},$$

where ε is a non-zero constant introduced to prevent the denominator from being zero. After the above normalization, the data distribution is transformed into a standard normal distribution with a mean of 0 and variance of 1. This treatment solves the problem of excessive differences in data distribution to a certain extent. However, it also disturbs the distribution characteristics learned at the upper layer. Therefore, the normalized data need to be corrected. The learning parameters γ and β are introduced to adjust the data and restore its original distribution characteristics prior to normalization

$$y_i = \gamma \tilde{z}_i + \beta,$$

where parameters γ and β can be learned by the back-propagation algorithm.

Thereafter, the fully connected layer expands all the elements in the obtained feature map into column vectors to fuse the features output by the convolutional layer

$$z_{k,l+1} = f_a\left(\sum_{i=1}^{n_l} w_{i,k,l+1} z_{ikl} + b_{k,l+1}\right),$$

where n_l denotes the number of neurons in the l-th layer.

Finally, after feature extraction through the convolution layers, pooling layers, and fully connected layers, the output layer produces a real-valued vector of size $1 \times c$, which can reconstruct the real-valued relative permittivity of a groove with the impedance boundary. Here, c represents the number of dielectrics in the groove.

Fig. 5 illustrates the proposed RV-CNN network architecture. Apart from the input and output layers, the hidden layer consists of BN layer, convolution layer, pooling layer, flatten layer and fully connected layer. The input feature map is first passed through the BN layer for normalization, followed by convolution operation with 64 filters of size 2×2 to obtain the first convolution result. It is then passed through the average convolution layer of size 2×2. The number of cells in the output layer depends on the sample size. It is one when the output corresponds to a uniform media reconstruction, and two when the output corresponds to a non-uniform media reconstruction. Since a step size of 1 shows better performance with smaller sized filters, the stride of the filters in all convolution and pooling layers is set to this value [1].

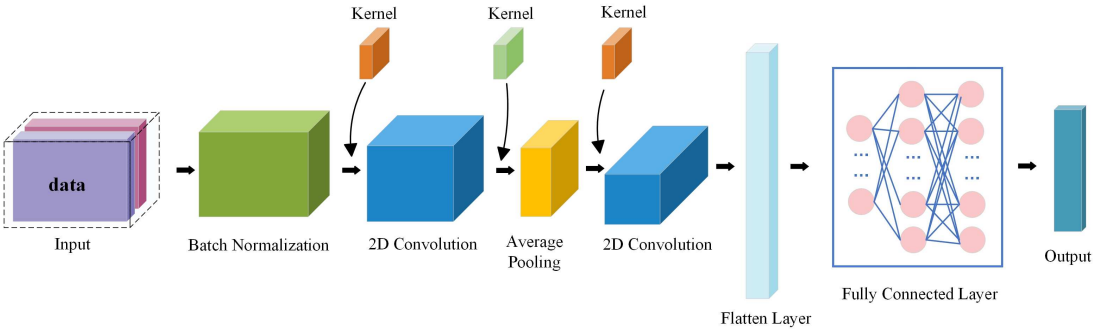


Figure 5: The RV-CNN framework for the groove scattering with impedance boundary condition.

3.2.2. Backpropagation process of RV-CNN

Error backpropagation is a crucial step for RV-CNN in performing parameter updates, which directly influences the model's reconstruction accuracy. Supervised training of RV-CNN is conducted to optimize the weights and bias parameters to ensure that the reconstructed values closely align with the true values. After feature extraction, the loss function quantifies the discrepancy between the reconstructed and true values in the output.

Assume that the training samples are represented as a paired set $\{(\boldsymbol{u}^1, \boldsymbol{t}^1), \ldots, (\boldsymbol{u}^m, \boldsymbol{t}^m)\}$, the loss function can be expressed as follows:

$$f_{loss}(\theta) = \frac{1}{m}\sum_{i=1}^{m}\left(t_i - \hat{t}_i\right)^2,$$

where θ represents the training parameter, and m denotes the sample size. The training input is given by $u^i \in C^{n_l}$, while $\hat{t}_i$ represents the corresponding training output.

In this study, the Adam algorithm is essentially an RMSprop with momentum terms. It dynamically adjusts the learning rate for each parameter while maintaining computational efficiency and low memory consumption, making it particularly suitable for solving problems with a large number of parameters [9].

The network parameters, which are initialized randomly, are updated in a direction that minimizes the loss function. Assuming that the number of iteration steps is t, we have

$$g_t = \nabla_{\theta_{t-1}} f_{loss}(\theta_{t-1}) = \theta_{t-1} - \eta\frac{\partial f_{loss}(\theta_{t-1})}{\partial \theta_{t-1}},$$

where g_t represents the gradient of parameter θ_{t-1}, and θ denotes the learning rate.

Among them, s_t and r_t represent the first- and second-moment estimates of the gradient, respectively, which can be regarded as estimates of the expectation $E|\boldsymbol{g}_t|$, $E|\boldsymbol{g}_t^2|$

$$\boldsymbol{s}_t = \beta_1 \boldsymbol{s}_{t-1} + (1-\beta_1)\boldsymbol{g}_t, \quad \boldsymbol{r}_t = \beta_2 \boldsymbol{s}_{t-1} + (1-\beta_2)\boldsymbol{g}_t^2.$$

Here β_1 and β_2 represent the exponential decay rates of the first and second moment estimates of the gradient, respectively, and take values in the range $\beta_1, \beta_2 \in [0, 1)$. $\boldsymbol{g}_t^2$ represents the element-wise product of the two vectors. To ensure smoother parameter updates, it is also necessary to compute the first- and second-moment estimates of the gradient after bias correction, as follows:

$$\hat{\boldsymbol{s}}_t = \frac{\boldsymbol{s}_t}{1-\beta_1^t}, \quad \hat{\boldsymbol{r}}_t = \frac{\boldsymbol{r}_t}{1-\beta_2^t},$$

where $\hat{\boldsymbol{s}}_t$ and $\hat{\boldsymbol{r}}_t$ can be regarded as unbiased estimates of the expectations.

Therefore, the updated rules of Adam algorithm in the real-valued case are as follows:

$$\Delta\theta_t = \frac{\eta}{\sqrt{\hat{\boldsymbol{r}}_t + \varepsilon}}\hat{\boldsymbol{s}}_t.$$

In this case, the learning rate is given by $\eta/\sqrt{\hat{\boldsymbol{r}}_t + \varepsilon}$, which adapts to changes in the parameters and makes smaller updates to the parameters associated with frequently occurring features.

The RV-CNN model is benchmarked in Python, using the Keras API. The loss function of the network is the mean squared error in the real-valued case, and the backpropagation algorithm employed is the Adam algorithm, which provides better optimization compared to other backpropagation algorithms [9]. The activation functions of the networks are all ELU activation functions, which account for the negative values in the input tensor (especially the phase part). The ELU activation functions is more effective compared to other activation functions [2].

3.3. Complex-value convolutional neural network model

In the case of complex-valued electromagnetic scattering fields, we extend the convolutional network to the complex domain by utilizing complex-valued neural networks. For CV-CNN, all elements of the network, including filter parameters, activation functions, and pooling operations, should be complex-valued. The architectural details of the CV-CNN are presented below.

3.3.1. The CV-CNN framework

To begin with, each hidden unit in the convolutional layer is connected to local patches of the feature map from the previous layer through a set of complex weight matrices, referred to as filters. The cells in the complex local patches are convolved with the complex filters and then activated using a nonlinear activation function. This process implies that the cells within the same feature map share a set of filters, with each filter extracting specific features from the input matrix corresponding to different channels of the output feature map. One of the benefits of using local connectivity and weight sharing is the significant reduction in the number of parameters to be learned, which not only decreases memory requirements but also helps avoid overfitting.

For any complex input data $Z = x + \mathrm{i}y$, we suppose there exists a complex convolution kernel $W = \alpha + \mathrm{i}\beta$, such that the complex input tensor undergoes a complex convolution process

$$W * Z = (A * x - B * y) + \mathrm{i}(Ay + Bx), \tag{3.1}$$

where $*$ denotes the convolution operation, x, y, A, and B are real-valued vectors, and $\mathrm{i} = \sqrt{-1}$ is the imaginary unit. The above equation can be rewritten in matrix form as follows:

$$\begin{pmatrix} \Re(W * Z) \\ \Im(W * Z) \end{pmatrix} = \begin{pmatrix} A & -B \\ B & A \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix},$$

where $\Re$ and $\Im$ represents the real and imaginary parts of the complex-valued data, respectively.

The operation process of the input data and complex convolution kernel is shown in Fig. 6. The purple part represents the real part of the input data or convolution kernel, while the rose-red data represents the imaginary part of the input data or convolution kernel. After performing the convolution operation as described in Eq. (3.1), the output feature map is still divided into real and imaginary parts.

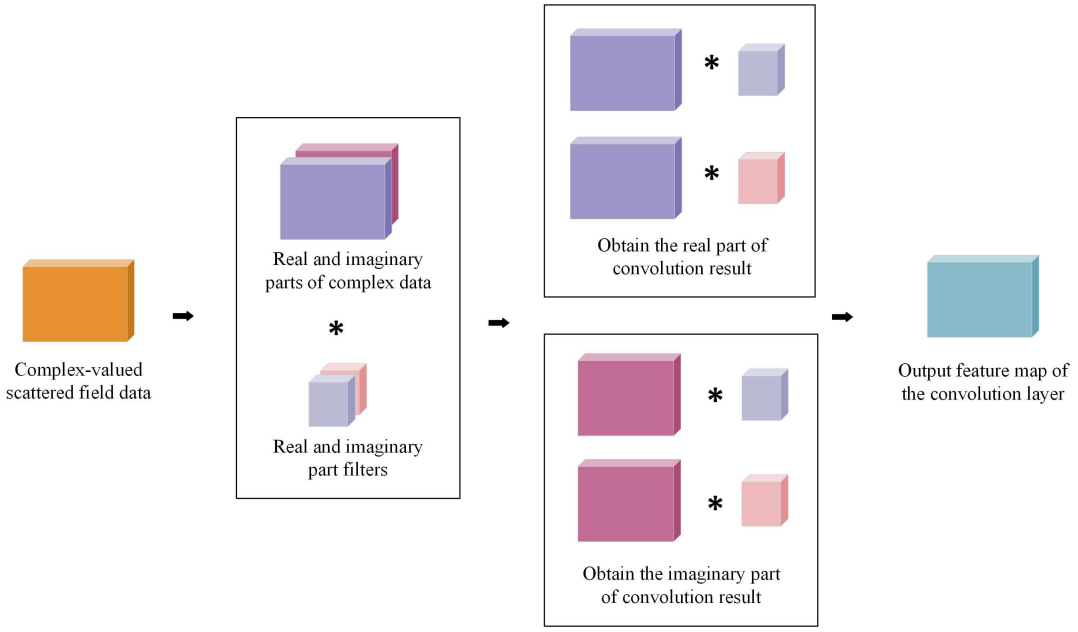


Figure 6: Complex-valued convolution operations.

The complex output feature map in the convolution layer is obtained by convolving all the input feature maps from the previous layer with the filters of this layer, adding bias, and then applying a nonlinear activation function. The specific activation operation is computed as follows:

$$\begin{aligned} Z_{k,l+1} &= f_a\big(\Re(O_{k,l+1})\big) + \mathrm{i} f_a\big(\Im(O_{k,l+1})\big) \\ &= \begin{cases} O_{k,l+1}, & O_{k,l+1} > 0, \\ 0.01 O_{k,l+1}, & O_{k,l+1} \le 0, \end{cases} \end{aligned} \tag{3.2}$$

where

$$\begin{aligned} O_{k,l+1} &= \sum_{i=1}^{n_l} W_{ik,l+1} * Z_{ikl} + b_{k,l+1} \\ &= \sum_{i=1}^{n_l} \big(\Re(W_{ik,l+1}) * \Re(Z_{ikl}) - \Im(W_{ik,l+1}) * \Im(Z_{ikl})\big) \\ &\quad + \mathrm{i} \sum_{i=1}^{n_l} \big(\Re(W_{ik,l+1}) * \Im(Z_{ikl}) + \Im(W_{ik,l+1}) * \Re(Z_{ikl})\big) + b_{k,l+1}. \end{aligned}$$

Here $f_a(\cdot)$ in Eq. (3.2) denotes the activation function, and the C-leaky-relu function is selected in this study. This function activates the real and imaginary parts of the complex numbers separately. When both the real and imaginary parts of the C-leaky-relu are strictly

positive or strictly negative simultaneously, the function satisfies the Cauchy-Riemann equation. Note that $O_{k,l+1}$ is defined as the input weighted sum for the k-th neuron in layer $l+1$, and n_l represents the number of inputs; $W_{ik,l+1}$ denotes the i-th weight coefficient of the k-th neuron in layer $l+1$; Z_{ikl} denotes the i-th output of the k-th neuron in layer l; $b_{k,l+1}$ represents the k-th bias in layer $l+1$, where $l \in \{1,\ldots,L-1\}$. $k = 1,\ldots,n_l$ corresponds to the number of layers in the neural network. We claim that our L-layer neural network has a topology of the form $n_1 - n_2 - \cdots - n_L$. Therefore, we consider convolutional neural networks that contain at least one hidden layer, implying that $L \geq 3$.

Then, we extend the real-valued pooling operation to complex-valued pooling. The separated complex pooling performs pooling operations separately for the real and imaginary parts of the complex-valued data, which simplifies the computation to some extent. For the uniformity and efficiency of the model, the calculation process, using the separated pooling operation, can be expressed as follows:

$$Z^{l+1} = f_p\big(\Re(Z^l)\big) + \mathrm{i} f_p\big(\Im(Z^l)\big),$$

where $f_p(\cdot)$ denotes the split-average pooling function.

Third, the neurons in the topmost layer of the fully connected layer are connected to all neurons in the previous layer, which can be viewed as a special case of convolution. Therefore, it is also referred to as a dense layer. The operation process of the complex-valued fully connected layer is similar to that of the convolutional layer, where the output feature maps are obtained by calculating the interaction between the real and imaginary part weight parameters and the real and imaginary part input feature maps

$$\begin{aligned} Z_{k,l+1} &= f_a\big(\Re(O_{k,l+1})\big) + \mathrm{i} f_a\big(\Im(O_{k,l+1})\big), \\ O_{k,l+1} &= \sum_{i=1}^{n_l} W_{i,k,l+1} Z_{ikl} + b_{k,l+1}, \end{aligned}$$

where n_l means the number of neurons in the l-th fully connected layer.

Finally, after feature extraction in the convolutional, pooling, and fully connected layers, the output of the final layer is a complex-valued vector of size $1 \times c$, which is used to reconstruct the relative permittivity within a groove with impedance boundary. Here, c represents the number of dielectric materials in the homogeneous or inhomogeneous medium inside the groove. The parameters of the CV-CNN are learned in an end-to-end supervised learning manner by minimizing the loss function of the training a priori, given the output samples and the output relative permittivity taken.

The proposed CV-CNN network architecture is shown in Fig. 7. In addition to the input and output layers, the hidden layer consists of convolution layer, pooling layer, flatten layer and fully connected layer. The input feature map is first convolved with 64 complex-valued filters of size 2×2 to obtain the first convolution result, which is then passed through the average pooling layer of size 2×2. Next, the filter of the same size is applied in the two-dimensional convolution layer. The number of units in the output layer depends on the sample size. It is 1 for a homogeneous medium reconstruction and 2 for an inhomogeneous medium reconstruction.

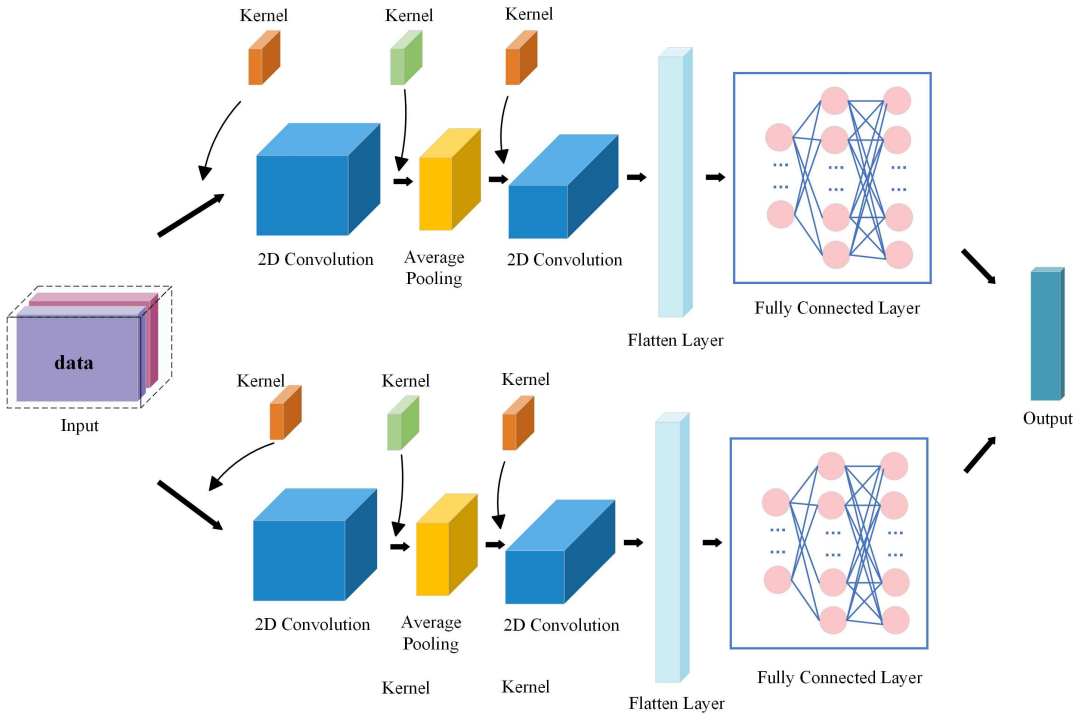


Figure 7: The CV-CNN framework for the groove scattering with impedance boundary condition.

3.3.2. Backpropagation process of CV-CNN

Supervised training of CV-CNN is performed to optimize the weight and bias parameters so that the reconstruction results of the network match the targets of the training data. Despite several stages of feature extraction, an error remains between the output and the targets, which is quantified by the loss function.

Suppose the training sample is represented by the pairwise set $\{(\boldsymbol{u}^1, \boldsymbol{t}^1), \ldots, (\boldsymbol{u}^m, \boldsymbol{t}^m)\}$, where m denotes the sample size. Here, $\boldsymbol{u}^j \in C^{n_1}$ represents the training sample, and $\hat{\boldsymbol{t}}^j = (t_1^j, \ldots, t_{n_L}^j) \in C^{n_L}$ represents the corresponding training targets. We feed $\boldsymbol{u}^j$ as the input layer to the untrained CV-CNN, and the signal is passed through the network. The corresponding output vector $\hat{\boldsymbol{t}}^j = (t_1^j, \ldots, t_{n_L}^j)$ is then obtained. Hence, we define the error function for the training samples as follows:

$$f_{loss}(\boldsymbol{w}) = \frac{1}{m}\sum_{j=1}^{m} \|\hat{\boldsymbol{t}}^j - \boldsymbol{t}^j\|^2 = \frac{1}{m}\sum_{j=1}^{m}\sum_{k=1}^{n_L} |\hat{\boldsymbol{t}}^j - \boldsymbol{t}^j|^2, \tag{3.3}$$

where $f_{loss}(\cdot)$ denotes the loss function, and $\boldsymbol{w}$ is the weight vector of the neural network. $|t| = t \cdot \bar{t}$ represents the norm of the complex number t, and $\|\boldsymbol{t}\| = \sqrt{(\boldsymbol{t}, \bar{\boldsymbol{t}})}$ denotes the norm of the vector $\boldsymbol{t}$. The loss function in Eq. (3.3) is a scalar real-valued function of the weights. It is minimized using the Adam algorithm to train the network parameters during the backpropagation process.

Both the backpropagation algorithm for the complex-valued case and the chain rule for the loss function are derived by expanding the real-valued case. In the backpropagation process of the complex-valued convolutional neural network, the network separately computes the real and imaginary parts of the last layer's neurons and calculates the derivatives of all network parameters from the final forward pass using the chain rule for complex-valued numbers. The weights are updated using Adam algorithm. The updating process for $\boldsymbol{g}_t$ becomes

$$\boldsymbol{g_t} = \nabla_{\theta_{t-1}} f_{loss}(\theta_{t-1}) = \theta_{t-1} - \eta \frac{\partial f_{loss}(\theta_{t-1})}{\partial \theta_{t-1}},$$

where

$$\begin{aligned}\frac{\partial f_{loss}(\theta_{t-1})}{\partial \theta_{t-1}} &= \frac{\partial f_{loss}(\theta_{t-1})}{\partial \Re(\theta_{t-1})} + \frac{\partial f_{loss}(\theta_{t-1})}{\partial \Im(\theta_{t-1})} \\ &= \left(\frac{\partial f_{loss}}{\partial \Re(O_{t-1})} \frac{\partial \Re(O_{t-1})}{\partial \Re(\theta_{t-1})} + \frac{\partial f_{loss}}{\partial \Im(O_{t-1})} \frac{\partial \Im(O_{t-1})}{\partial \Re(\theta_{t-1})} \right) \\ &\quad + \mathrm{i}\left(\frac{\partial f_{loss}}{\partial \Re(O_{t-1})} \frac{\partial \Re(O_{t-1})}{\partial \Im(\theta_{t-1})} + \frac{\partial f_{loss}}{\partial \Im(O_{t-1})} \frac{\partial \Im(O_{t-1})}{\partial \Im(\theta_{t-1})} \right).\end{aligned}$$

Thus, both $\boldsymbol{s_t}$ and $\boldsymbol{r_t}$ also become complex-valued

$$\begin{aligned}\boldsymbol{s_t} &= \mu \boldsymbol{s_{t-1}} + (1-\mu)\boldsymbol{g_t}, \\ \boldsymbol{r_t} &= \nu \boldsymbol{r_{t-1}} + (1-\nu)\big(\Re^2(g_t) + \mathrm{i}\Im^2(g_t)\big).\end{aligned}$$

In the above equation, $\boldsymbol{g_t}$ should be squared. This procedure corresponds to the conjugate multiplication of complex values, where the real and imaginary parts are squared separately. Similarly, we introduce $\hat{\boldsymbol{s}}_t$ and $\hat{\boldsymbol{r}}_t$ as corrections to the second-order momentum of $\boldsymbol{s_t}$ and $\boldsymbol{r_t}$

$$\hat{\boldsymbol{s}}_t = \frac{\boldsymbol{s_t}}{1-\mu^t}, \quad \hat{\boldsymbol{r}}_t = \frac{\boldsymbol{r_t}}{1-\nu^t}.$$

Overall, the parameter update process is as follows:

$$\Delta\theta_t = -\left(\frac{\eta \Re(\hat{\boldsymbol{s}}_t)}{\sqrt{\Re(\hat{\boldsymbol{r}}_t)} + \varepsilon} + \mathrm{i}\frac{\eta \Im(\hat{\boldsymbol{s}}_t)}{\sqrt{\Im(\hat{\boldsymbol{r}}_t)} + \varepsilon} \right).$$

The learning rates for the real and imaginary parts in the above equation are equivalently expressed as $\eta/\sqrt{\Re(\hat{\boldsymbol{r}}_t)} + \varepsilon$ and $\eta/\sqrt{\Im(\hat{\boldsymbol{r}}_t)} + \varepsilon$, respectively.

The network's loss function is the mean squared error in the complex domain, and the backpropagation algorithm employed is the Adam optimizer, also in the complex domain. The activation functions utilized throughout the network are C-leaky ReLU functions. This function assigns a non-zero slope to the complex data, making it more suitable for complex scattered field data by sacrificing hard-zero sparsity to obtain potentially more robust gradients during optimization [15].

In summary, compared to RV-CNN, the operation of CV-CNN is more complex. With the same input feature map and convolutional kernel size, the number of complex-valued convolutional kernels is twice that of the real-valued convolutional kernels, which may result in longer training times during network.

4. Numerical Experiments and Results

This section presents the detailed implementation of RV-CNN and CV-CNN for the reconstruction of electromagnetic parameters in groove scattering with impedance boundaries. First, the basic structure of the input data is outlined. Then, the network architecture and parameter configurations of both algorithms are described in detail. Finally, the dataset is fed into the model for reconstruction, and the reconstruction performance is evaluated.

In the following examples, we set the wave number $k_0 = 2\pi$, which corresponds to $\omega = 600\pi$ rad/s and incident frequency $f = 300$MHz.

4.1. Reconstruction of the relative permittivity

4.1.1. Application to RV-CNN

We select the rectangular groove filled with homogeneous media under impedance boundary conditions, or the inhomogeneous groove with turbine-shaped internal interfaces, as shown in Fig. 8. The relative permittivity of the filled media in the groove is uniformly distributed in the real-valued range $[0, 10]$. The incident angle of the plane wave increases uniformly from 0 to 90 degrees. The complex-valued scattered field, measured at the observation point on the aperture surface, is processed into magnitude and phase and fed into the two channels of the input tensor, respectively. The detailed data for the training and testing sets, along with the model parameters, are provided in Tables 1 and 2.

Table 1: Data set dimensionality of RV-CNN.

Overview of real-valued data		Input ($DS \times AN \times SN \times 1$)	Output ($DS \times C$)
Homogeneous media	Training set	$382 \times 33 \times 33 \times 2$	382×1
	Testing set	$18 \times 33 \times 33 \times 2$	18×1
Inhomogeneous media	Training set	$382 \times 33 \times 33 \times 2$	382×2
	Testing set	$18 \times 33 \times 33 \times 2$	18×2

Table 2: Training parameters of the RV-CNN model.

Optimizer parameters	Model parameters
Batch size	50
Epoch	200/500
Learning rate	0.01
Hide layer	8
Activation function	ELU

Homogeneous-media groove. Fig. 9 shows the change of loss function and metrics with the number of iterations in the training process and testing process of RV-CNN on the real-valued case uniform media data set. As can be seen from the figure, the loss functions

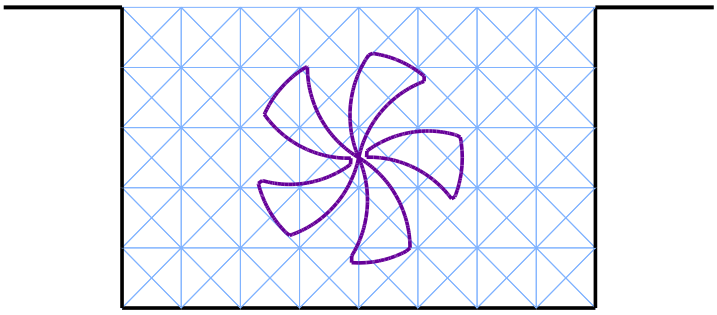


Figure 8: Inhomogeneous groove.

for both the training and testing sets converge around 150, while their respective metrics converge around 175. There is no obvious rebound trend in the loss function curve and metrics curve, and the model does not appear to be overfitted.

The reconstruction results of RV-CNN on the real-valued dataset of parameters of homogeneous media are shown in Fig. 10, where the horizontal axis indicate the true values of relative permittivity taken, and the vertical axis indicate the reconstructed value of relative permittivity. Fig. 10 visually demonstrates the reconstruction performance of RV-CNN for a homogeneous medium in the real-valued case.

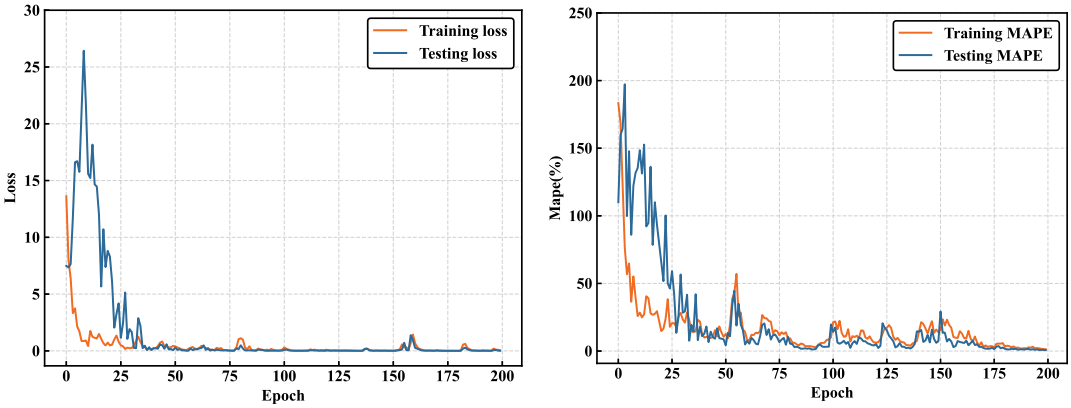
Inhomogeneous-media groove. As shown in Fig. 11, it can be seen that the RV-CNN reduces the loss value to nearly 0 after 100 iterations and remains stable during the subsequent iterations.

The reconstruction results are shown in Fig. 12. Note that the reconstructed values of the two relative permittivities are approximately close to the true values. Increasing the number of reconstructed relative permittivity does not significantly affect the reconstruction performance of the model.

4.1.2. Application to CV-CNN

Filling the groove with impedance boundary conditions with the homogeneous or inhomogeneous media, respectively. The second medium inside the non-uniform groove is set as a turbine shape. The relative permittivity of the filled media in the groove is uniformly distributed in the complex domain [0, 10+10i], and the incident angle of the plane incident wave is uniformly increasing from 0 to 90 degrees. The complex-valued scattering field measured at the observation point of the aperture surface forms the input tensor directly without processing. The detailed data of the training and testing sets, along with the model parameters, are provided in Tables 3 and 4.

Homogeneous-media groove. As shown in Fig. 13, after 100 iterations, the loss values for both the training and testing sets remain close to 0, and the metrics stay at a low level, with no significant upward trend.



(a) Loss curve (b) Metrics curve

Figure 9: Iterative process of RV-CNN on homogeneous data set.

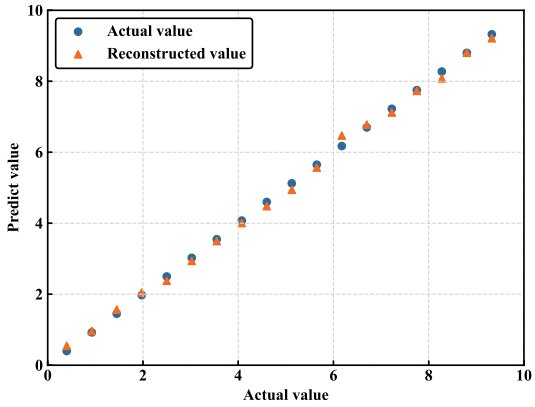
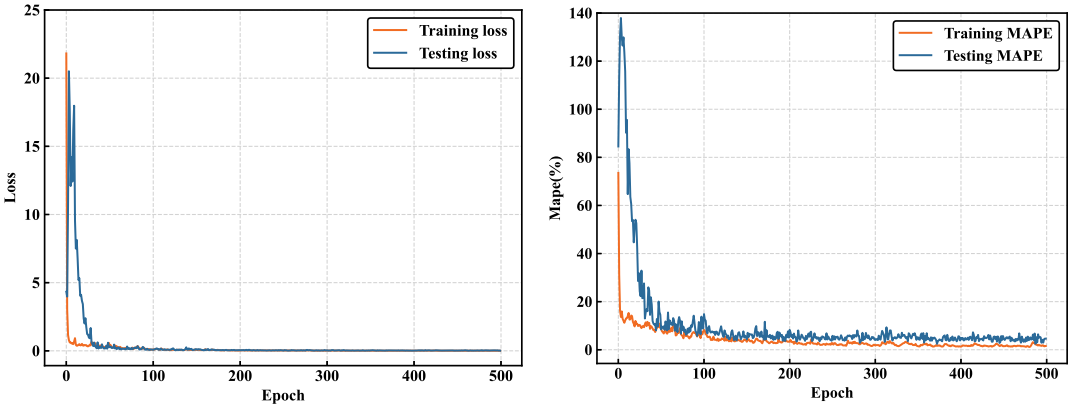
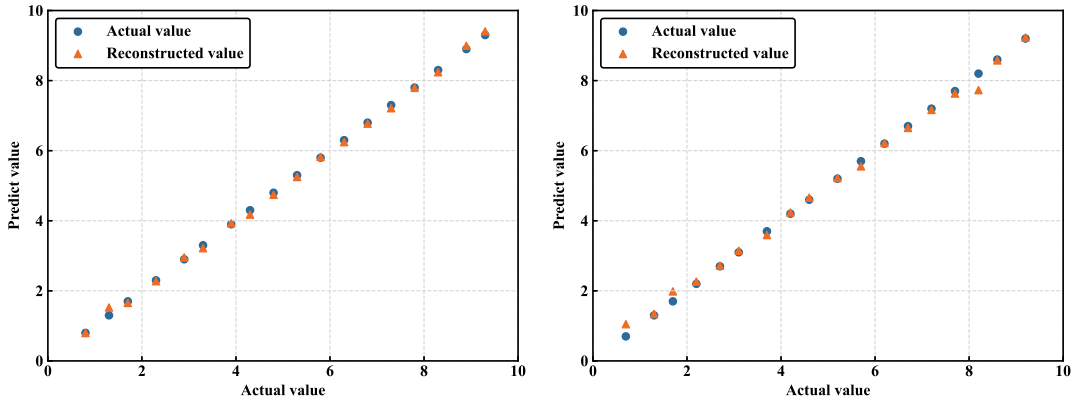


Figure 10: Reconstruction results of RV-CNN on homogeneous media dataset.



(a) Loss curve (b) Metrics curve

Figure 11: Iterative process of RV-CNN on inhomogeneous dataset.



(a) The reconstructive value of ε_r^+ (b) The reconstructive value of ε_r^-

Figure 12: Reconstruction results of RV-CNN on inhomogeneous media dataset.

Table 3: Data set dimensionality of CV-CNN.

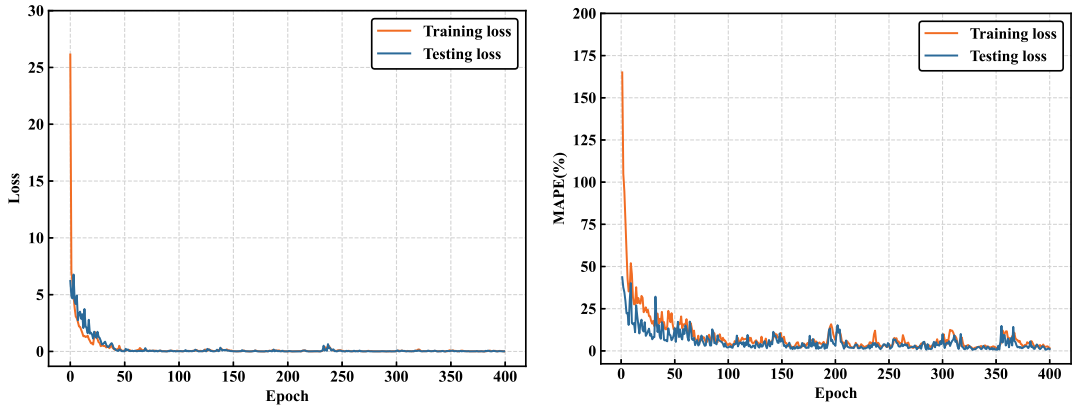
Overview of complex-valued data		Input ($DS \times AN \times SN \times 1$)	Output ($DS \times C$)
Homogeneous media	Training set	$382 \times 33 \times 33 \times 1$	382×1
	Testing set	$18 \times 33 \times 33 \times 1$	18×1
Inhomogeneous media	Training set	$382 \times 33 \times 33 \times 1$	382×2
	Testing set	$18 \times 33 \times 33 \times 1$	18×2

Table 4: Training parameters of the CV-CNN model.

Optimizer parameters	Model parameters
Batch size	50
Epoch	400/500
Learning rate	0.01
Hide layer	7
Activation function	cv-leaky-relu

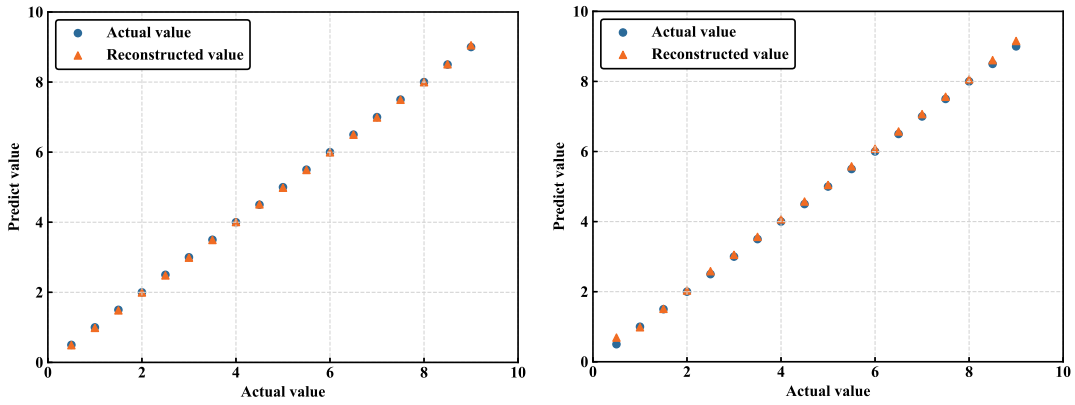
To more intuitively reflect the advantages and disadvantages of the reconstruction effect, the real and imaginary parts of the complex data are split and visualized separately. The reconstruction results for the complex uniform media are shown in Fig. 14. It can be seen that the CV-CNN reconstruction effect is excellent for the impedance-bounded groove scattering with homogeneous media.

Inhomogeneous-media groove. The training and testing iterative processes of CV-CNN on the non-uniform complex-valued dataset of the media parameters are shown in Fig. 15. From the figure, the loss function reaches its minimum around the 30th iteration, and the metric curve stabilizes at a relatively low level thereafter, with no significant upward trend.



(a) Loss curve (b) Metric curve

Figure 13: Iterative process of CV-CNN on homogeneous dataset.



(a) The reconstructive value of real (b) The reconstructive value of image

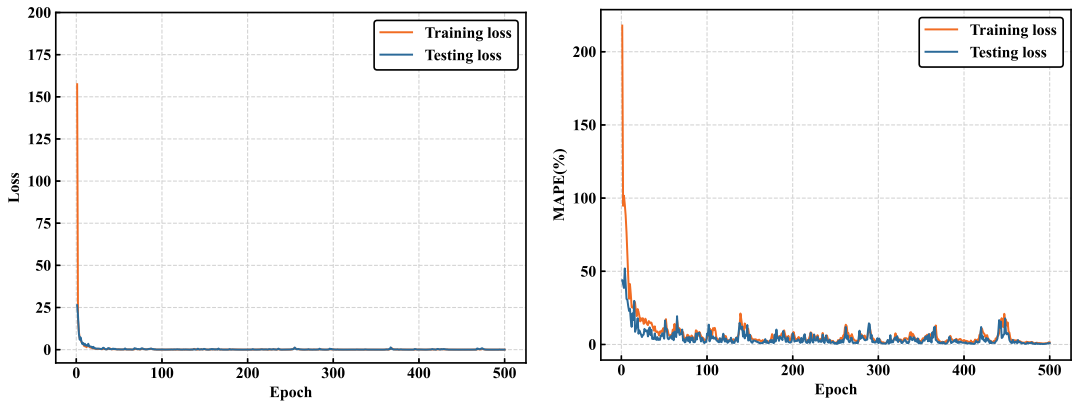
Figure 14: Reconstruction results of CV-CNN on homogeneous media dataset.

This suggests that the reconstruction model has converged to the global optimal solution.

The comparison between the true and reconstructed values is shown below. As seen in Fig. 16, for inhomogeneous media, CV-CNN can efficiently reconstruct the electromagnetic parameters of the groove scattering and also demonstrate good generalization ability.

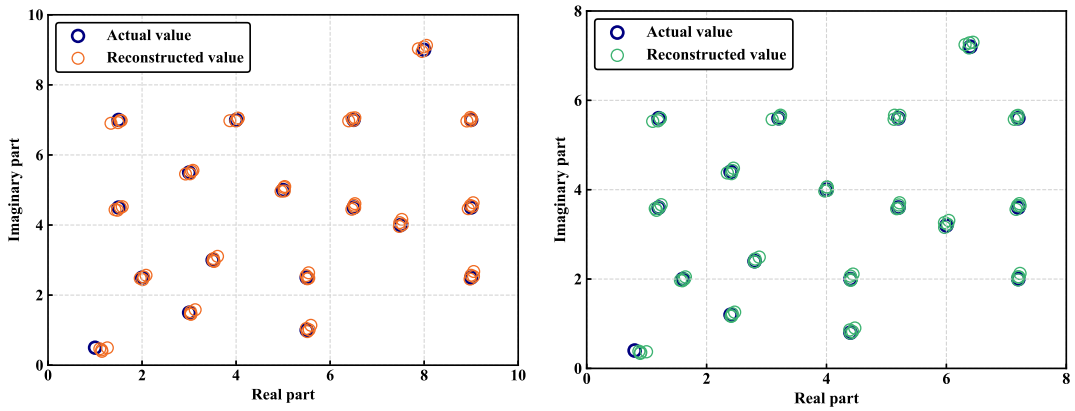
In addition to using the mean absolute percentage error (MAPE) as the evaluation criterion to analyze the model's performance, we also selected the mean squared error (MSE), root mean squared error (RMSE), and the correlation coefficient (R^2).

As shown in Tables 5 and 6, the MAPE of the reconstruction results for the four test sets using RV-CNN and CV-CNN inversion of the relative permittivity is significantly less than 4%. Both the MSE and RMSE are extremely small, and R^2 approaches 1, demonstrating excellent reconstruction and generalization abilities. From the two tables, we can be seen that CV-CNN shows a significant improvement in the MAPE index for both homogeneous and inhomogeneous media.



(a) Loss curve (b) Metric curve

Figure 15: Iterative process of CV-CNN on inhomogeneous data set.



(a) Real and imaginary parts of ε_r^+ (b) Real and imaginary parts of ε_r^-

Figure 16: Reconstruction results of CV-CNN on inhomogeneous media data set.

Finally, we compared our approach with BP neural networks and SVM in both real-number and complex-number domains for inhomogeneous media. Fig. 17 presents the reconstruction results of test samples by RV-CNN, CV-CNN, BP and SVM. The first row shows the comparison between RV-CNN, BP and SVM, where the x-axis represents ε_r^+, and the y-axis represents ε_r^-. In the second row, it shows the comparison among CV-CNN, BP and SVM. The x-axis represents the real part of the relative permittivity, while the y-axis represents the imaginary part. It can be clearly observed that the reconstructed values of RV-CNN and CV-CNN are more concentrated around the true values, while those of BP and SVM are discretely distributed around the true values. Therefore, the reconstruction effect of the proposed method is superior to those of BP and SVM, demonstrating excellent adaptability and flexibility.

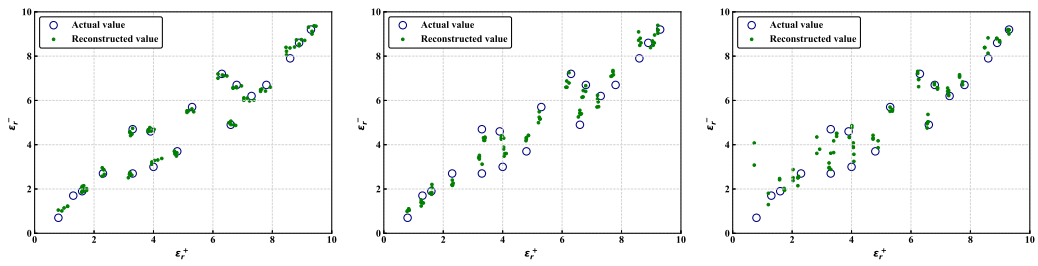
Fig. 18 compares the evaluation metrics of RV-CNN and CV-CNN with those of BP and SVM, respectively. The left y-axis corresponds to the mean absolute error (MAE) repre-

Table 5: Comparison of statistical indicators in different media.

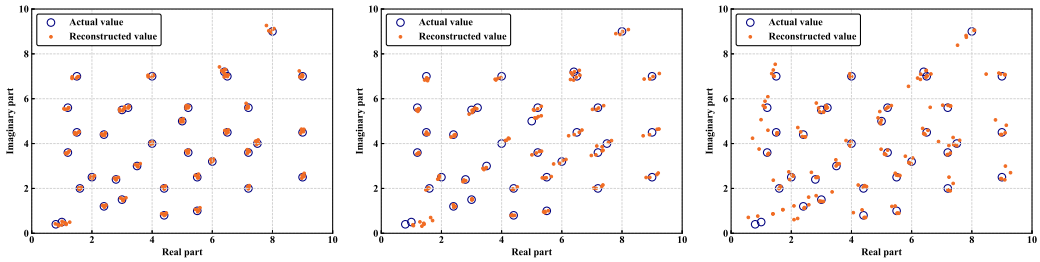
Media		MAPE (%)	MSE	RMSE	R^2
Homogeneous media		1.49756	0.00249	0.04988	0.99970
Inhomogeneous media	ε_r^+	0.83272	0.00359	0.05988	0.99879
	ε_r^-	1.05117	0.00493	0.07021	0.99945

Table 6: Evaluation metrics of CV-CNN reconstruction results.

Statistical indicators		MAPE (%)	MSE	RMSE	R^2
Homogeneous media	Real part of ε_r	0.00230	0.04791	0.00908	0.99969
	Imaginary part of ε_r	0.00245	0.04947	0.00723	0.99970
	ε_r	0.03023	0.17388	1.20114	0.99971
Inhomogeneous media	Real part of ε_r^+	0.00199	0.04459	0.29556	0.99975
	Imaginary part of ε_r^+	0.00115	0.03397	0.44208	0.99985
	ε_r^+	0.00157	0.01400	0.11840	0.99980
	Real part of ε_r^-	0.00126	0.03554	0.29043	0.99975
	Imaginary part of ε_r^-	0.00100	0.03163	0.58408	0.99980
	ε_r^-	0.00113	0.01190	0.10910	0.99978



(a) Reconstruction of real-valued permittivity using RV-CNN, BP, and SVM



(b) Reconstruction of complex-valued permittivity using CV-CNN, BP, and SVM

Figure 17: Comparison of reconstructed relative permittivity using RV-CNN, CV-CNN, BP and SVM.

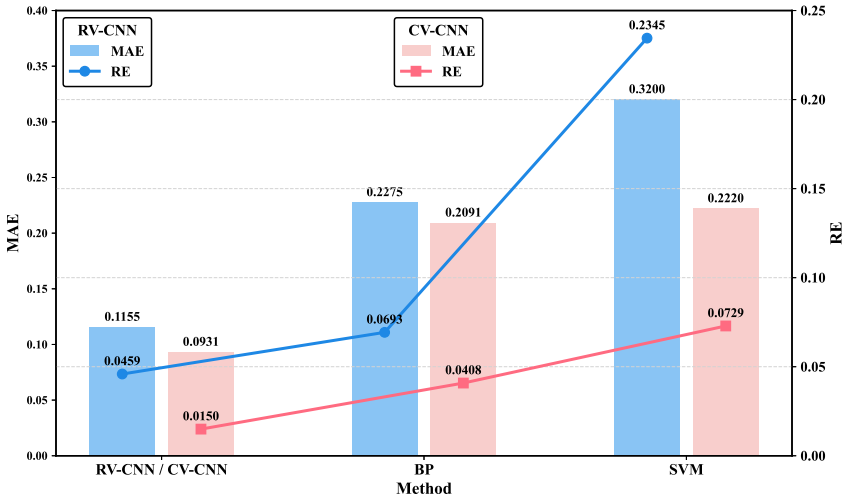


Figure 18: Comparison of error metrics RV-CNN and CV-CNN separately with BP and SVM.

sented by bar charts, while the right y-axis corresponds to the relative error (RE) presented by line charts. The blue color signifies the comparison among RV-CNN, BP and SVM, while the pink color denotes the comparison among CV-CNN, BP and SVM. It can be intuitively observed that the proposed method has smaller errors and higher inversion accuracy, demonstrating strong robustness.

4.2. Reconstruction of the relative permittivity distribution matrix

4.2.1. The impedance cavity with a semi-elliptical interior filled with inhomogeneous four-leaf clovers

We consider a semi-elliptical groove in the shape of a four-leaf clover, filled with inhomogeneous media, as shown in Fig. 19. The semi-major axis of the semi-elliptical is 0.5m, while the semi-minor axis measures 0.4m. The level set function of the internal medium consists of two parts,

$$\varphi_i^1(a,b) = \sqrt{(a-0.625)^2+2(b+0.2)^2}\cdot\sqrt{(a-0.375)^2+2(b+0.2)^2}-0.125^2,$$
$$\varphi_i^2(a,b) = \sqrt{2(a-0.5)^2+(b+0.325)^2}\cdot\sqrt{2(a-0.5)^2+(b+0.075)^2}-0.125^2.$$

In the RV-CNN model, the relative permittivity of the internal medium is set to the range $[4.3, 10]$, with a step size of $h_{int}^1 = 0.3$. Meanwhile, the permittivity of the remaining part of the groove varies from $[0.5, 4.3]$, with a step size of $h_{out}^1 = 0.2$. In contrast, in the CV-CNN model, the real part of the relative permittivity for the internal medium ranges from $[0.5, 10]$, with $h_{int}^2 = 0.5$. Concurrently, the imaginary part is equal to the real counterpart. The relative permittivity for other sections of the groove increases uniformly within the range $[0.4, 8]$, with $h_{out}^2 = 0.4$. This configuration results in a total of 400 labels, from which 382

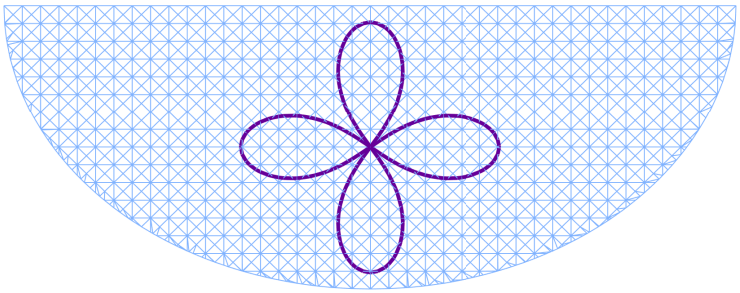


Figure 19: The impedance groove filled with clover inhomogeneous media.

samples are selected for training and 18 samples for testing. The incident angle increases uniformly from 0° to 90° in steps of 2°.

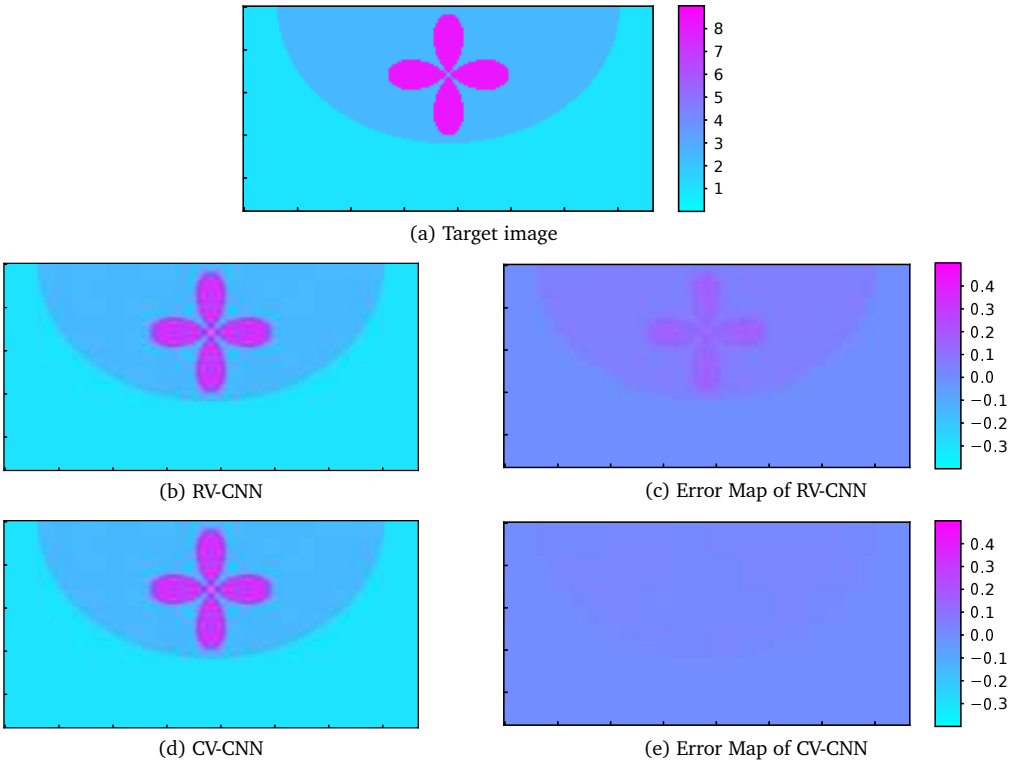
To ensure the accuracy of the inversion results, we employed a grid discretization method with a resolution of 192×192, resulting in an input matrix consisting of 96×192 pixel grids. The values in this matrix correspond to the relative permittivities to be inverted. Both magnitude and phase are processed separately before forming the input tensor. The final output matrix has dimensions of 96×192.

To verify that the generalization ability of CV-CNN is superior to that of RV-CNN, we invert the distribution matrix of the relative permittivity in the real number domain and visualize one of the samples. Fig. 20 illustrates reconstruction results and corresponding errors for both RV-CNN and CV-CNN models. The top row represents the target image, with the left column displaying the reconstruction results for both RV-CNN and CV-CNN, and the right column showing the corresponding error maps. Although the two reconstruction images in the left column appear similar, the error maps clearly indicate that using CV-CNN results in near-zero errors at the internal media interfaces, demonstrating its superior performance compared to RV-CNN.

Fig. 21 presents the inversion results to validate the effectiveness of the CV-CNN in handling inverse problems with complex relative permittivity. The first row shows the original image, the second row presents the inverted image, and the third row displays the error image. The first column represents the real part of the relative permittivity, while the second column represents the imaginary part. The inverted images are largely consistent with the original images, and the error maps clearly indicate that the inversion performance is satisfactory.

To verify the generalization capability of the CV-CNN model, this study devises an out-of-distribution (OOD) test. With all other parameters kept unchanged, the real part of the relative permittivity for the internal medium ranges from $[0.1, 0.48]$, with $h_{o_1} = 0.02$, the real part for the remaining regions within the groove ranges $[0.01, 0.39]$, with $h_{o_2} = 0.02$, and the imaginary part of the relative permittivity for the groove medium equals its corresponding real part.

Fig. 22 presents the results of out-of-distribution testing for CV-CNN. Three rows of images are presented: the first row illustrates the original image, the second row depicts the



(a) Target image

(b) RV-CNN

(c) Error Map of RV-CNN

(d) CV-CNN

(e) Error Map of CV-CNN

Figure 20: Reconstruction results of RV-CNN and CV-CNN.

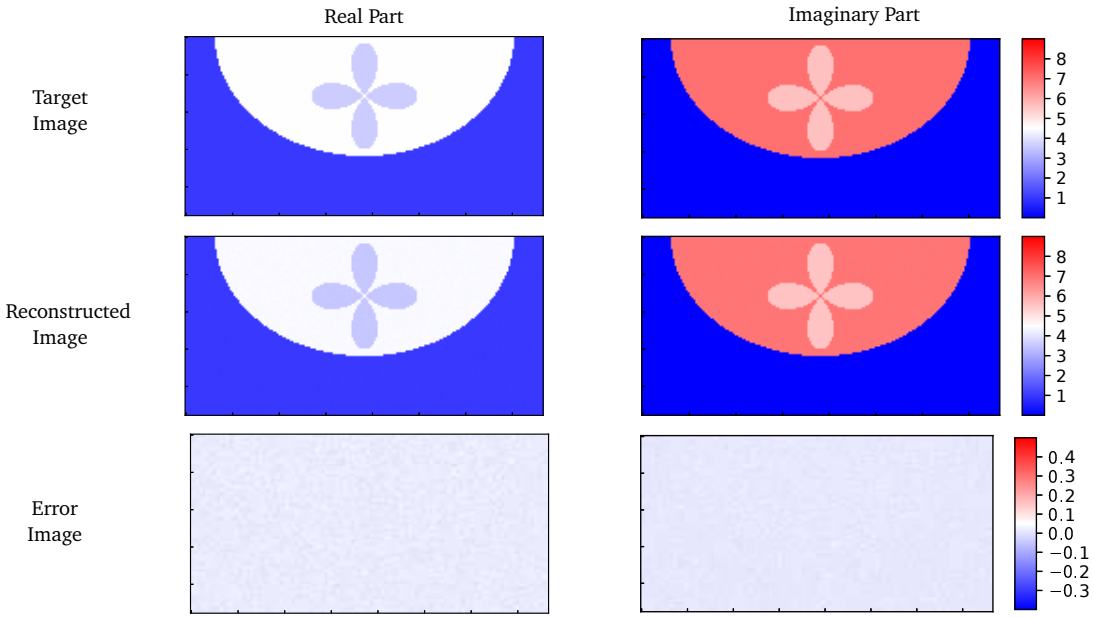


Figure 21: Reconstruction results of CV-CNN.

reconstructed image, and the third row presents the error map. As the range of the relative permittivity is narrowed, the color bar scale is correspondingly adjusted to a numerical interval matching the data distribution. The results indicate that CV-CNN still maintains low inversion errors in this out-of-distribution scenario, demonstrating its strong generalization ability.

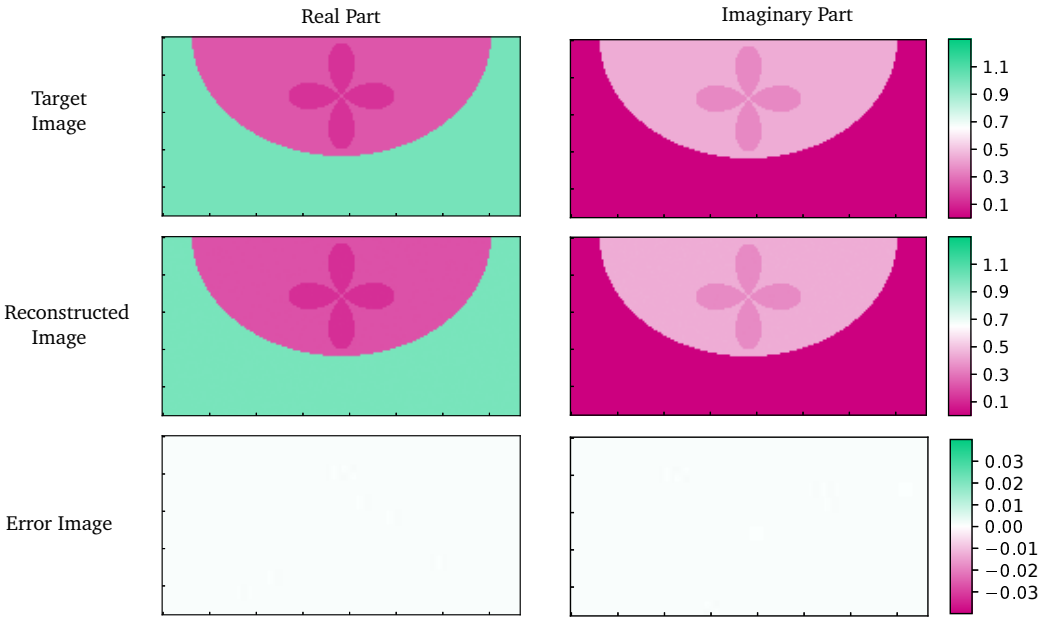


Figure 22: Reconstruction results of out-of-distribution testing for CV-CNN.

4.2.2. The impedance groove with a peanut interior filled with two inhomogeneous circular hexagonal

In this example, we address a more complex groove scattering problem characterized by a peanut-shaped groove, measuring 2m in length and 0.6m in width, as depicted in Fig. 23.

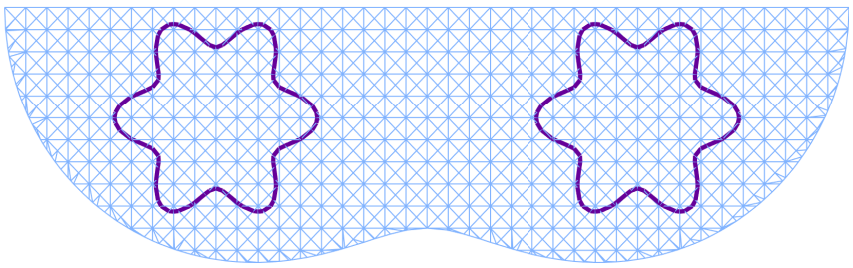


Figure 23: The internal partitioning of a peanut-shaped groove with rounded hexagonal contours.

The internal medium consists of two circular hexagonal structures, with the level set function expression provided below

$$\varphi_i^1(a,b) = \sqrt{(a-0.5)^2+(b+0.25)^2} - 0.04\cos\left(6\arctan\left(\frac{b+0.25}{a-0.5}\right)\right) - 0.2,$$
$$\varphi_i^2(a,b) = \sqrt{(a-1.5)^2+(b+0.25)^2} - 0.04\cos\left(6\arctan\left(\frac{b+0.25}{a-1.5}\right)\right) - 0.2$$

with the relative permittivity, incident angle, and other parameters held constant, the input matrix of the network remains at a size of 192×192, while the output matrix is reduced to 48×192.

To assess algorithm performance effectively, Gaussian white noise with known intensity was introduced into the images during the evaluation process. The evaluation metric used is the signal-to-noise ratio (SNR), which is expressed as follows:

$$SNR = 20 \times \lg\left(\frac{1}{\upsilon}\right),$$

where υ represents the ratio of the noise amplitude to the signal amplitude.

As υ increases, its interference with signal clarity becomes more pronounced. In this study, experiments were conducted using SNR values of 26.02 dB (υ at 5%), 20.00 dB (υ at 10%), 13.98 dB (υ at 20%), and 7.96 dB (υ at 40%).

The reconstruction results of CV-CNN with added noise are presented in Fig. 24. The left column shows the reconstructed images, while the right column displays the corresponding error maps. It can be observed that as the noise ratio υ increases, the errors grow larger, leading to a more blurred reconstructed image. Similarly, corresponding tests were conducted for RV-CNN. The loss values for the training sets of both networks are visualized.

In Fig. 25, the x-axis represents the ratio of noise amplitude to signal amplitude, while the y-axis shows the loss value on the training set. As υ increases from 5% to 40%, it is

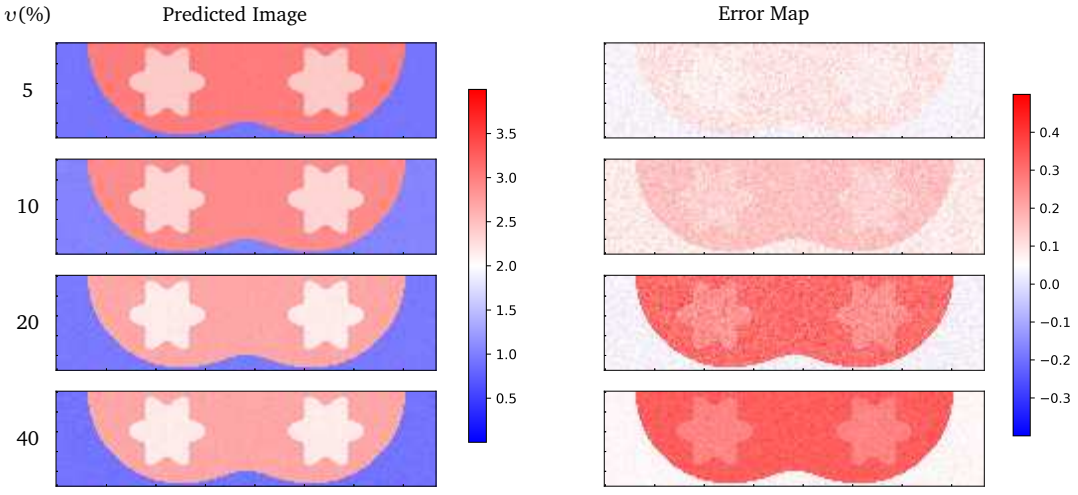


Figure 24: Reconstructed results and error maps of CV-CNN at different SNR levels.

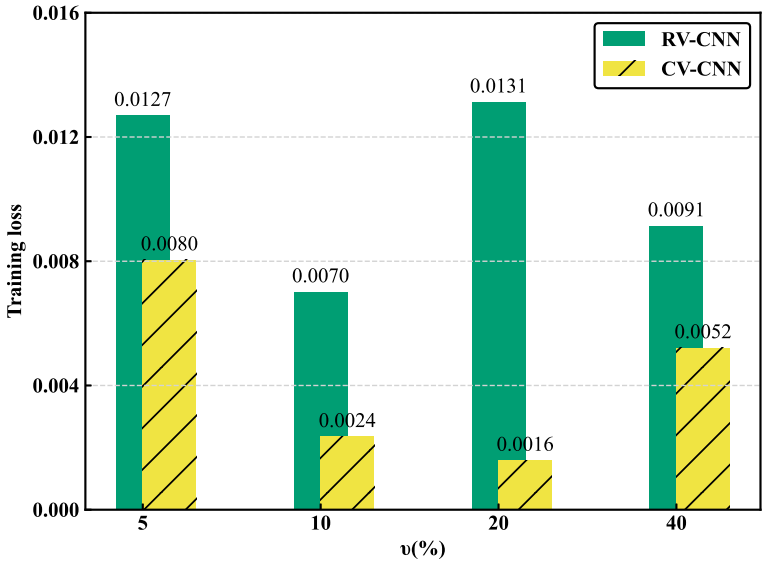


Figure 25: Comparison of loss between RV-CNN and CV-CNN at different SNR levels.

clear that, for any given ratio, the loss value of RV-CNN consistently exceeds that of CV-CNN. This suggests that CV-CNN achieves both lower and more stable loss values. This observation provides strong evidence of the superior performance of CV-CNN.

In addition to conducting anti-interference detection against white Gaussian noise with global continuity, we also test the robustness of this network under locally discrete interference environments. We add varying degrees of salt-and-pepper noise to the images and introduced discrete perturbations into the data to test the generalization ability of the model. The peak signal-to-noise ratio (PSNR) is selected as the evaluation metric, and its formula is

$$\mathrm{PSNR} = 10 \cdot \log_{10}\left(\frac{\mathrm{MAV}^2}{\mathrm{MSE}}\right),$$

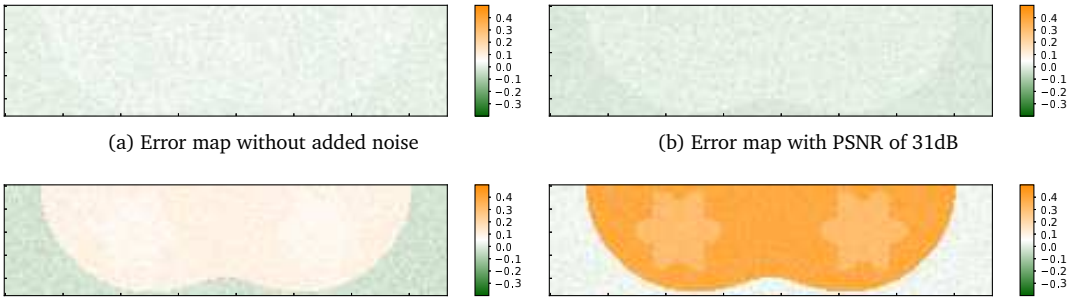
where MAV represents the maximum absolute value among the real and imaginary parts of the original complex image. The pixel value of the complex image is denoted as $z[m,n] = x[m,n]+\mathrm{i}y[m,n]$, where m and n are pixel coordinates with the value ranges satisfying $m \times n \in M \times N$, and $M \times N$ represents the image size, the expression for MAV is given by

$$\mathrm{MAV} = \max\Big(\max_{m,n}|x[m,n]|, \max_{m,n}|y[m,n]|\Big),$$

and the mean squared error is defined as

$$\mathrm{MSE} = \frac{1}{M \times N}\sum_{m=1}^{M}\sum_{n=1}^{N}\big|z[m,n]-z'[m,n]\big|^2,$$

where $z'[m,n]$ denotes the pixel value of the noisy image.



(a) Error map without added noise

(b) Error map with PSNR of 31dB

(c) Error map with PSNR of 24dB

(d) Error map with PSNR of 11 dB

Figure 26: Error maps under different noise intensities (noiseless, PSNR of 31dB, 24dB and 11dB respectively).

PSNR is inversely proportional to the noise intensity, meaning that the lower the PSNR, the higher the noise intensity and the greater the disturbance to the image. As the inverted image is in close approximation to the actual image, Fig. 26 presents the error maps under (a) noiseless conditions, (b) mild noise, (c) moderate noise and (d) severe salt-and-pepper noise. Notably, a lighter color signifies a better inversion performance. Experimental results demonstrate that CV-CNN exhibits favorable stability in restoring denoised images, indicating its strong robustness in feature extraction for noisy images.

5. Conclusion

In this paper, two deep learning algorithms, RV-CNN and CV-CNN, are proposed for reconstructing the real-valued and complex-valued relative permittivity of the media in the groove with impedance boundary. In the process of network training, both RV-CNN and CV-CNN use the relative permittivity as labels. Specifically, RV-CNN takes the magnitude and phase of the scattering field as features. In contrast, CV-CNN makes a breakthrough by directly using complex-domain scattering field data as features. It is worth emphasizing that CV-CNN, by virtue of its unique complex-domain processing capability, can directly reconstruct the contrast matrix of the complex permittivity. This matrix fully preserves the complex characteristics of electromagnetic parameters. After visualization processing, the matrix can be transformed into intuitive physical images, effectively presenting the distribution characteristics of electromagnetic parameters. The results indicate that both of the proposed deep learning algorithms can reconstruct electromagnetic parameters from scattering field data with a limited number of observation points. Moreover, CV-CNN outperforms RV-CNN in terms of reconstruction accuracy and robustness, demonstrating a stronger ability to handle complex media. Learning process of the black-box strategy may lead that reconstruction is dependent on the training data and network structure. However, the model have good reconstruction and generalization capabilities if computational power is ensured.

Acknowledgments

The research of M.Z. was supported in part by the National Natural Science Foundation of China (Grant No. 12271159) and by the Natural Science Foundation of Hebei Province (Grant No. A2024502009). The research of L.W. was supported by the National Natural Science Foundation of China (Grant Nos. 12171482, U23A20301) and by the State Key Laboratory of Petroleum Resources and Engineering, China University of Petroleum-Beijing (Grant No. PRE/DX-2504).

References

[1] J.A. Barrachina, C. Ren, C. Morisseau, G. Vieillard and J-P. Ovarlez, *Complex-valued vs. real-valued neural networks for classification perspectives: An example on non-circular data*, in: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2990–2994, (2021).

[2] D.A. Clevert, T. Unterthiner and S. Hochreiter, *Fast and accurate deep network learning by exponential linear units (ELUS)*, in: *4th International Conference on Learning Representations (ICLR)*, pp. 1–14 (2015).

[3] K. Du, B.Y. Li, W.W. Sun and H.H. Yang, *Electromagnetic scattering from a cavity embedded in an impedance ground plane*, Math. Methods Appl. Sci. **41**, 7748–7765 (2018).

[4] Y. Gao, H.Y. Liu, X.C. Wang and K. Zhang, *On an artificial neural network for inverse scattering problems*, J. Comput. Phys. **448**, Paper No. 110771 (2022).

[5] Y. Gao and K. Zhang, *Machine learning based data retrieval for inverse scattering problems with incomplete data*, J. Inverse Ill-Posed Probl. **29**, 249–266 (2021).

[6] M. Ghosh and B. Basu, *Detection of malignant breast tissue using SAR observation with microwave imaging and convolutional neural network*, Comput Electr Eng. **119**, Paper No. 109620 (2024).

[7] J. Guo, Q. Yang, M.J. Cai, G.Z. Yan, Z.K. Guo, *Reconstruction of a crack with the incident waves and measurements inside a penetrable cavity*, J. Inverse Ill-Posed Probl. **27**, 643–656 (2019).

[8] Y.Q. Hu, F. Cakoni and J.J. Liu, *The inverse scattering problem for a partially coated cavity with interior measurements*, Appl. Anal. **93**, 936–956 (2014).

[9] D.P. Kingma and J. Ba, *ADAM: A method for stochastic optimization*, in: *3th International Conference on Learning Representations (ICLR)*, pp. 1–15 (2015).

[10] Y. LeCun, Y. Bengio and G. Hinton, *Deep learning*, Nature. **521(7553)**, 436–444 (2015).

[11] L.L. Li, L.G. Wang, F.L. Teixeira, C. Liu, A. Nehorai and T.J. Cui, *Deepnis: Deep neural network for nonlinear electromagnetic inverse scattering*, IEEE Trans. Antennas Propag. **67(3)**, 1819–1825 (2018).

[12] P.J. Li and A.H. Wood, *A two-dimensional Helmhotlz equation solution for the multiple cavity scattering problem*, J. Comput. Phys. **240**, 100–120 (2013).

[13] L.H. Liu, J.Q. Cai and Y.Z. Xu, *Regularized newton iteration method for a penetrable cavity with internal measurements in inverse scattering problem*, Math. Methods Appl. Sci. **43(5)**, 2665–2678 (2020).

[14] X.D. Liu, S.X Meng and B. Zhang, *Modified sampling method with near field measurements*, SIAM J. Appl. Math. **82(1)**, 244–266 (2022).

[15] A.L. Maas, A.Y. Hannun and A.Y. Ng, *Rectifier nonlinearities improve neural network acoustic models*, 2013. `https://www.awnihannun.com/papers/relu_hybrid_icml2013_final.pdf`

[16] S.X. Meng, H. Haddar and F. Cakoni, *The factorization method for a cavity in an inhomogeneous medium*, Inverse Probl. **30(4)**, Paper No. 045008 (2014).
[17] J.F. Ning, F.Q. Han and J. Zou, *A direct sampling-based deep learning approach for inverse medium scattering problems*, Inverse Probl. **40(1)**, Paper No. 015005 (2023).
[18] J.F. Ning, F.Q. Han and J. Zou, *A direct sampling method and its integration with deep learning for inverse scattering problems with phaseless data*, SIAM J. Sci. Comput. **47(2)**, C343–C368 (2025).
[19] V. Puzyrev, *Deep learning electromagnetic inversion with convolutional neural networks*, Geophys. J. Int. **218(2)**, 817–832 (2019).
[20] F.L. Qu, J.Q. Yang and H.W. Zhang, *Shape reconstruction in inverse scattering by an inhomogeneous cavity with internal measurements*, SIAM J. Imaging Sci. **12(2)**, 788–808 (2019).
[21] Y. Sanghvi, Y. Kalepu and U.K. Khankhoje, *Embedding deep learning in inverse scattering problems*, IEEE Trans. Comput. Imaging. **6**, 46–56 (2019).
[22] Y.X. Tan and S.F. Zhu, *Numerical shape reconstruction for a semi-linear elliptic interface inverse problem*, East Asian J. Appl. Math. **14(1)**, 147–178 (2024).
[23] F. Tian, Y.K. Yang and L. Mao, *Electromagnetic inversion algorithm based on convolutional neural network*, in: *2020 Cross Strait Radio Science & Wireless Technology Conference (CSRSWTC)*, pp. 1–3, (2020).
[24] X.F. Wang, C. Yang, Z. Zhang and Y. Du, *Dct-based deep learning of polarimetric scattering from a dielectric cylinder*, IEEE Geosci. Remote Sens. Lett. **19**, 1–5 (2022).
[25] Y. Wang, Y.W. Zhao, L.F. Wu, Y.Y. Zhang and J. Hu, *Hybrid dilated convolutional neural network for solving electromagnetic inverse scattering problems*, Int. J. RF Microwave Comput. Aided Eng. **32(3)**, Paper No. e23023 (2022).
[26] L.H. Xie, Q. Zhao, C.G. Ma, B.B. Liao and J.J Huo, *Ü-net: Deep-learning schemes for ground penetrating radar data inversion*, J. Environ. Eng. Geophys. **25(2)**, 287–292 (2020).
[27] K.W. Xu, C. Zhang, X.Z. Ye and R.C. Song, *Fast full-wave electromagnetic inverse scattering based on scalable cascaded convolutional neural networks*, IEEE Trans. Geosci. Remote Sens. **60**, 1–11 (2021).
[28] Y. Yin and L. Yan, *Physics-aware deep learning framework for the limited aperture inverse obstacle scattering problem*, SIAM J. Sci. Comput. **47(2)**, C313–C342 (2025).
[29] F. Zeng and S.X. Meng, *The interior inverse electromagnetic scattering for an inhomogeneous cavity*, Inverse Probl. **37(2)**, Paper No. 025007 (2021).
[30] Z.M. Zhang, H.P. Wang, F. Xu and Y.-Q. Jin, *Complex-valued convolutional neural network and its application in polarimetric SAR image classification*, IEEE Trans. Geosci. Remote Sens. **55(12)**, 7177–7188 (2017).
[31] M.L. Zhao, X.Q. Fang, Z.B. Yuan and L.Q. Wang, *Electromagnetic scattering from arbitrarily shaped cavities coated by absorbing material filled with heterogeneous anisotropic media*, Eng. Anal. Bound. Elem. **165**, Paper No. 105761 (2024).
[32] M.L. Zhao, L.Y. Li, X.Q. Fang and L.Q. Wang, *Electromagnetic reconstruction based on HS-Mnet neural network for the heterogeneous cavity with impedance boundary*, J. Comput. Phys. **541**, Paper No. 114279 (2025).
[33] M.L. Zhao, Z.H. Qiao and T. Tao, *A fast high order method for electromagnetic scattering by large open cavities*, J. Comput. Math. **29(3)**, 287–304 (2011).
[34] M.L. Zhao, N. Zhu and L.Q. Wang, *The electromagnetic scattering from multiple arbitrarily shaped cavities with inhomogeneous anisotropic media*, J. Comput. Phys. **489**, Paper No. 112274 (2023).