

Article

High-Resolution ISAR Imaging and Autofocusing via 2D-ADMM-Net

Xiaoyong Li ¹ , Xueru Bai ^{2,*} and Feng Zhou ¹

¹ Key Laboratory of Electronic Information Countermeasure and Simulation Technology of Ministry of Education, Xidian University, Xi'an 710071, China; lixyxd@stu.xidian.edu.cn (X.L.); fzhou@mail.xidian.edu.cn (F.Z.)

² National Lab of Radar Signal Processing, Xidian University, Xi'an 710071, China

* Correspondence: xrbai@xidian.edu.cn

Abstract: A deep-learning architecture, dubbed as the 2D-ADMM-Net (2D-ADN), is proposed in this article. It provides effective high-resolution 2D inverse synthetic aperture radar (ISAR) imaging under scenarios of low SNRs and incomplete data, by combining model-based sparse reconstruction and data-driven deep learning. Firstly, mapping from ISAR images to their corresponding echoes in the wavenumber domain is derived. Then, a 2D alternating direction method of multipliers (ADMM) is unrolled and generalized to a deep network, where all adjustable parameters in the reconstruction layers, nonlinear transform layers, and multiplier update layers are learned by an end-to-end training through back-propagation. Since the optimal parameters of each layer are learned separately, 2D-ADN exhibits more representation flexibility and preferable reconstruction performance than model-driven methods. Simultaneously, it is able to better facilitate ISAR imaging with limited training samples than data-driven methods owing to its simple structure and small number of adjustable parameters. Additionally, benefiting from the good performance of 2D-ADN, a random phase error estimation method is proposed, through which well-focused imaging can be acquired. It is demonstrated by experiments that although trained by only a few simulated images, the 2D-ADN shows good adaptability to measured data and favorable imaging results with a clear background can be obtained in a short time.



Citation: Li, X.; Bai, X.; Zhou, F. High-Resolution ISAR Imaging and Autofocusing via 2D-ADMM-Net. *Remote Sens.* **2021**, *13*, 2326. <https://doi.org/10.3390/rs13122326>

Keywords: radar imaging; sparse reconstruction; autofocusing; 2D-ADMM-Net

Received: 7 May 2021
Accepted: 9 June 2021
Published: 13 June 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

High-resolution inverse synthetic aperture radar (ISAR) imaging plays a significant role in space situation awareness and air target surveillance [1,2]. Under ideal observational environments with high signal-to-noise ratios (SNRs) and complete echo matrices, well-focused imaging can be acquired by classic techniques such as the range-Doppler (RD) algorithm and the polar formatting algorithm (PFA) [3]. For a target with a small radar cross section (RCS) or a long observation distance, however, the SNR of the received echoes is low due to limited transmitted power. In addition, the existence of strong jamming and the resource scheduling of the cognitive radar may result in incomplete data along the range or/and azimuth direction(s). The complex observational environments discussed above, i.e., incomplete data and low SNRs, cause severe performance degradation or even invalidate the available imaging techniques. As ISAR images are generally sparse in the image domain, high-resolution ISAR imaging under complex observational environments based on the theory of sparse signal reconstruction has received intensive attention in the radar imaging community in recent years [4,5], in which the reconstruction of sparse images (i.e., the distribution of dominant scattering centers) from noisy or gapped echoes given the observation dictionary is sought.

In addition, the motion of the target can be decomposed into translational motion and rotational motion, where the former is not beneficial to imaging and needs to be

compensated by range alignment and autofocus. Traditional autofocus algorithms can obtain satisfying imaging results from complete radar echoes [6]. For complex observational environments, however, they cannot achieve good performance due to the deficiency of radar echoes and low SNRs. In recent years, parametric autofocus techniques [7] have been applied for sparse imaging. Although they are superior to traditional methods under sparse aperture conditions, their performance is dependent on imaging quality. Therefore, in turn, a better sparse reconstruction method is needed.

The available sparse ISAR imaging methods can be divided into three categories: (1) model-driven methods; (2) data-driven methods; and (3) combined model-driven and data-driven methods. Among them, model-driven methods construct the sparse observation model and obtain high-resolution images by l_0 -norm or l_1 -norm optimization. The l_0 -norm optimization, e.g., orthogonal MP (OMP) [8] and smoothed l_0 -norm method [9], cannot guarantee that the solution is sparsest and may converge to the local minima. The l_1 -norm optimization, e.g., the fast iterative shrinkage-thresholding algorithm (FISTA) [10,11] and alternating direction method of multipliers (ADMM) [12,13], is the convex approximation of the l_0 -norm [14]. However, the regularization parameter directly affects the performance and how to determine its optimum value remains an open problem [4]. Additionally, vectorized optimization requires long operating times and a large memory storage space. To improve the efficiency, methods based on matrix operations such as 2D-FISTA [15] and 2D-ADMM [16] are proposed.

Data-driven methods solve the nonlinear mapping from echoes to the 2D image by designing and training a deep network [17]. In the training process, the target echoes and corresponding ISAR images are adopted as the inputs and the label, respectively, and the loss function is the NMSE between the network output and the label. In order to minimize the loss function (i.e., to obtain the optimal network parameters), the network parameters are randomly initialized and updated by the gradient descent method iteratively until convergence [18]. Then, the trained network is applied to generate focused imaging of an unknown target. Facilitated by off-line network training, such methods achieve the reconstruction of multiple images rapidly, and typical networks include the complex-value deep neural network (CV-DNN) [19]. Nevertheless, the subjective network design process lacks unified criterion and theoretical support, which makes it difficult to analyze the influence of network structure and parameter settings on reconstruction performance. In addition, the large number of unknown parameters require massive training samples to avoid overfitting.

The combined model-driven and data-driven methods first expand the model-driven methods into a deep network [20], and then utilize only a few training samples to learn the optimal values of the adjustable parameters [21]. Finally, they output the focused image of an unknown target by the trained network. Such methods effectively solve the difficulties in: (1) setting proper parameters for model-driven methods; (2) clearly explaining the physical meaning of the network; and (3) generating a large number of training samples to avoid overfitting for data-driven methods. A common technique to expand the model-driven methods is unrolling [22], which utilizes a finite-layer hierarchical architecture to implement iterations. As a typical imaging network, the deep ADMM network [23] uses measured data for effective training, which is usually limited due to observation conditions, and autofocus is not considered for sparse aperture ISAR imaging.

To tackle the above-mentioned problems, this article proposes the 2D-ADMM-Net (2D-ADN) to achieve well-focused 2D imaging under complex observational environments, and its key contributions mainly include the following: (a) Mapping from the ISAR images to echoes in the wavenumber domain is established, which forms as a 2D sparse reconstruction problem. Then, the 2D-ADMM method is provided with phase error estimation for focused imaging. (b) Based on the 2D-ADMM, 2D-ADN is designed to include the reconstruction layers, nonlinear transform layers, and multiplier update layers. Then, the adjustable parameters are estimated by minimizing the loss function through back-propagation in the complex domain. (c) Simulation results demonstrate that the 2D-ADN, which is trained

by a small number of samples generated from point-scattering model, obtains the best reconstruction performance. For both the complete and incomplete data with low SNRs, the 2D-ADN combined with random phase error estimation obtains better-focused imaging of measured aircraft data with a clearer background than the available methods.

The remainder of this article is organized as follows. Section 2 establishes the sparse observation model for the high-resolution 2D imaging and provides the iterative formulae of 2D-ADMM with random phase error estimation. Section 3 introduces the construction of 2D-ADN in detail. Section 4 gives the network loss function and derives the back-propagation formulae in the complex domain. Section 5 carries out various experiments to prove the effectiveness of 2D-ADN. In Section 6, we discuss the performance of 2D-ADN, and Section 7 concludes the article with suggestions for future work.

2. Modeling and Solving

2.1. 2D Modeling

After translational motion compensation [24,25], echoes $\mathbf{Y} \in \mathbb{C}^{P \times Q}$ in the wavenumber domain satisfy:

$$\mathbf{Y} = \Phi_1 \mathbf{X} \Phi_2 \mathbf{E} + \mathbf{N} \quad (1)$$

where $\Phi_1 \in \mathbb{R}^{P \times U}$ is the over-complete range dictionary, $\mathbf{X} \in \mathbb{C}^{U \times V}$ is the 2D distribution of the scattering centers, $\Phi_2 \in \mathbb{R}^{V \times Q}$ is the over-complete Doppler dictionary, $\mathbf{N} \in \mathbb{C}^{P \times Q}$ is the complex noise matrix, and $\mathbf{E} \in \mathbb{C}^{Q \times Q}$ is the diagonal random phase error matrix.

$$\mathbf{E} = \begin{bmatrix} e^{j\varphi_1} & 0 & 0 & 0 & 0 \\ 0 & \ddots & 0 & 0 & 0 \\ 0 & 0 & e^{j\varphi_q} & 0 & 0 \\ 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & e^{j\varphi_Q} \end{bmatrix} \quad (2)$$

In (2), φ_q denotes phase error of the q th echo.

For convenience, we vectorized (1) as:

$$\begin{aligned} \mathbf{y} &= (\Phi_2 \mathbf{E})^T \otimes \Phi_1 \mathbf{x} + \mathbf{n} \\ &= \Phi \mathbf{x} + \mathbf{n} \end{aligned} \quad (3)$$

where $\mathbf{y} \in \mathbb{C}^{PQ}$ is the vector form of $\mathbf{Y}$, $\otimes$ is the Kronecker product, $\Phi = (\Phi_2 \mathbf{E})^T \otimes \Phi_1 \in \mathbb{C}^{PQ \times UV}$, $\mathbf{x} \in \mathbb{C}^{UV}$ is the vector form of $\mathbf{X}$, and $\mathbf{n} \in \mathbb{C}^{PQ}$ is the vector form of $\mathbf{N}$.

2.2. The 2D-ADMM Method

Finding the optimal solution to (3) is a linear inverse problem, which can be further converted into an unconstrained optimization by introducing the regularization term:

$$\min_{\mathbf{x}} \left\{ \frac{1}{2} \|\Phi \mathbf{x} - \mathbf{y}\|_2^2 + \lambda \|\mathbf{x}\|_1 \right\} \quad (4)$$

where λ is the regularization parameter.

According to the variable splitting technique [26], (4) is equivalent to:

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{z}} \left\{ \frac{1}{2} \|\Phi \mathbf{x} - \mathbf{y}\|_2^2 + \lambda \|\mathbf{z}\|_1 \right\} \\ \text{s.t. } \mathbf{x} = \mathbf{z} \end{aligned} \quad (5)$$

and the augmented Lagrangian function is:

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{z}, \alpha} L_\rho(\mathbf{x}, \mathbf{z}, \alpha) \\ L_\rho(\mathbf{x}, \mathbf{z}, \alpha) = \frac{1}{2} \|\Phi \mathbf{x} - \mathbf{y}\|_2^2 + \lambda \|\mathbf{z}\|_1 + \langle \alpha, \mathbf{x} - \mathbf{z} \rangle + \frac{\rho}{2} \|\mathbf{z} - \mathbf{x}\|_2^2 \end{aligned} \quad (6)$$

where $\langle \cdot, \cdot \rangle$ is the inner product, α is the vector of Lagrangian multiplier, and ρ is the penalty parameter.

ADMM decomposes (6) into three sub-problems by minimizing $L_\rho(\mathbf{x}, \mathbf{z}, \alpha)$ with respect to $\mathbf{x}$, $\mathbf{z}$, and α , respectively,

$$\begin{cases} \mathbf{x}^{(n)} = \underset{\mathbf{x}}{\operatorname{argmin}} \frac{1}{2} \|\Phi \mathbf{x} - \mathbf{y}\|_2^2 + \langle \alpha^{(n-1)}, \mathbf{x} - \mathbf{z}^{(n-1)} \rangle + \frac{\rho}{2} \|\mathbf{z}^{(n-1)} - \mathbf{x}\|_2^2 \\ \mathbf{z}^{(n)} = \underset{\mathbf{z}}{\operatorname{argmin}} \lambda \|\mathbf{z}\|_1 + \langle \alpha^{(n-1)}, \mathbf{x} - \mathbf{z}^{(n-1)} \rangle + \frac{\rho}{2} \|\mathbf{z} - \mathbf{x}^{(n)}\|_2^2 \\ \alpha^{(n)} = \alpha^{(n-1)} + \rho(\mathbf{x}^{(n)} - \mathbf{z}^{(n)}) \end{cases} \quad (7)$$

where n is the iteration index. Let $\mathbf{b} = \frac{\alpha}{\rho}$, the solutions to (7) satisfy:

$$\begin{cases} \mathbf{x}^{(n)} = \left(\mathbf{z}^{(n-1)} - \mathbf{b}^{(n-1)} \right) - \frac{1}{1+\rho} \Phi^H \left(\Phi \left(\mathbf{z}^{(n-1)} - \mathbf{b}^{(n-1)} \right) - \mathbf{y} \right) \\ \mathbf{z}^{(n)} = S \left(\mathbf{x}^{(n)} + \mathbf{b}^{(n-1)}; \lambda/\rho \right) \\ \mathbf{b}^{(n)} = \mathbf{b}^{(n-1)} + \eta \left(\mathbf{x}^{(n)} - \mathbf{z}^{(n)} \right) \end{cases} \quad (8)$$

where $S(\cdot)$ is the shrinkage function [27] defined by $S(\mathbf{w}, \tau) = \operatorname{sign}(\mathbf{w}) \max\{|\mathbf{w}| - \tau, 0\}$, τ is the threshold, and η is an update rate for the Lagrangian multiplier.

The 2D-ADMM estimates the 2D image $\mathbf{X}^{(n)}$ by:

$$\mathbf{X}^{(n)} = \left(\mathbf{Z}^{(n-1)} - \mathbb{B}^{(n-1)} \right) - \frac{1}{1+\rho} \Phi_1^H \left(\Phi_1 \left(\mathbf{Z}^{(n-1)} - \mathbb{B}^{(n-1)} \right) \Phi_2 \mathbf{E} - \mathbf{Y} \right) (\Phi_2 \mathbf{E})^H \quad (9)$$

In addition, the matrix forms of $\mathbf{z}$ and $\mathbf{b}$, i.e., $\mathbf{Z}$ and $\mathbb{B}$, are calculated by,

$$\mathbf{Z}^{(n)} = S \left(\mathbf{X}^{(n)} + \mathbb{B}^{(n-1)}; \lambda/\rho \right) \quad (10)$$

$$\mathbb{B}^{(n)} = \mathbb{B}^{(n-1)} + \eta \left(\mathbf{X}^{(n)} - \mathbf{Z}^{(n)} \right) \quad (11)$$

2.3. Phase Error Estimation and Algorithm Summation

The phase error is estimated by optimizing the following objective function [28]:

$$\hat{\phi}_q = \underset{\phi_q}{\operatorname{argmin}} \|\mathbf{Y}(\cdot, q) - e^{j\phi_q} (\Phi_1 \mathbf{X} \Phi_2)(\cdot, q)\|_2^2 \quad (12)$$

where $\mathbf{Y}(\cdot, q)$ is the q th column of $\mathbf{Y}$, and $(\Phi_1 \mathbf{X} \Phi_2)(\cdot, q)$ is the q th column of $\Phi_1 \mathbf{X} \Phi_2$.

Let the derivative of (12) with respect to ϕ_q be zero, then:

$$\hat{\phi}_q = -\tan^{-1} \frac{\operatorname{Re} \left(((\Phi_1 \mathbf{X} \Phi_2)(\cdot, q))^H \mathbf{Y}(\cdot, q) \right)}{\operatorname{Im} \left(((\Phi_1 \mathbf{X} \Phi_2)(\cdot, q))^H \mathbf{Y}(\cdot, q) \right)} \quad (13)$$

where $\operatorname{Re}(\cdot)$ and $\operatorname{Im}(\cdot)$ represent the real and the imaginary parts, respectively, and the phase error matrix $\hat{\mathbf{E}}$ is constructed by (2).

Algorithm 1 summarizes the high-resolution 2D ISAR imaging and autofocus method based on 2D-ADMM and random phase error estimation, where N denotes the total number of iterations. Analysis and experiments have shown that the choice of λ and ρ has a great influence on the imaging quality, and improper initialization may generate a defocused image with a noisy background. In addition, λ and ρ cannot be adaptively adjusted in each iteration, demonstrating a lack of flexibility.

Algorithm 1. Autofocusing by 2D-ADMM

```

1. Initialize  $\hat{\mathbf{E}} = \mathbf{I}$ ,  $\lambda$ , and  $\rho$ 
2. For  $t = 1 : T$ 
  For  $n = 1 : N$ 
    Update  $\mathbf{X}_t^{(n)}$  by (9);
    Update  $\mathbf{Z}_t^{(n)}$  by (10);
    Update  $\mathbb{B}_t^{(n)}$  by (11);
  End
  Calculate  $\hat{\mathbf{E}}$  by (13) and (2);
End
3. Output  $\mathbf{X}_T^{(N+1)}$  and  $\hat{\mathbf{E}}$ 

```

3. Structure of 2D-ADN

To tackle the aforementioned problems, we modified 2D-ADMM and expanded it into 2D-ADN. There are similarities between deep networks and iterative algorithms [22] such as 2D-ADMM. Particularly, the matrix multiplication is similar to the linear mapping of the deep network, the shrinkage function is similar to the nonlinear operation, and the adjustable parameters are similar to network parameters. Therefore, the 2D-ADMM algorithm can be unfolded into 2D-ADN. As shown in Figure 1, the network has N stages, and stage n , $n \in [1, N]$, represents the n th iteration described by Algorithm 1. Typically, one stage consists of three layers, i.e., the reconstruction layer, the nonlinear transform layer, and the multiplier update layer, which correspond to (9), (10), and (11), respectively. The inputs of 2D-ADN are echoes in the 2D wavenumber domain, and the output is the reconstructed 2D high-resolution image. Below, we will derive the forward-propagation formulae of each layer.

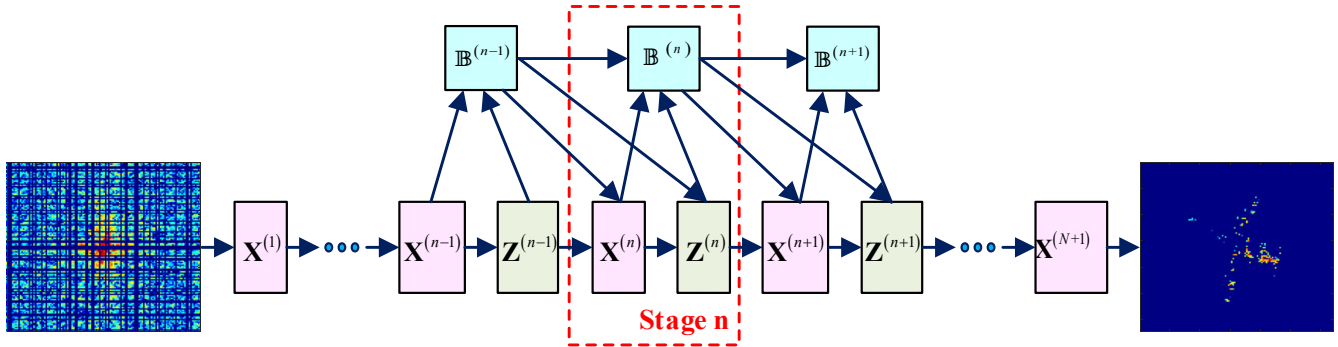


Figure 1. Structure of the 2D-ADN.

3.1. Reconstruction Layer

As shown in Figure 2, the inputs of the reconstruction layer are $\mathbf{Z}^{(n-1)}$ and $\mathbb{B}^{(n-1)}$, and the output is:

$$\mathbf{X}^{(n)} = \left(\mathbf{Z}^{(n-1)} - \mathbb{B}^{(n-1)} \right) - \frac{1}{1 + \rho^{(n)}} \Phi_1^H \left(\Phi_1 \left(\mathbf{Z}^{(n-1)} - \mathbb{B}^{(n-1)} \right) \Phi_2 \mathbf{E} - \mathbf{Y} \right) (\Phi_2 \mathbf{E})^H \quad (14)$$

where the penalty parameter $\rho^{(n)}$ is the adjustable parameter.

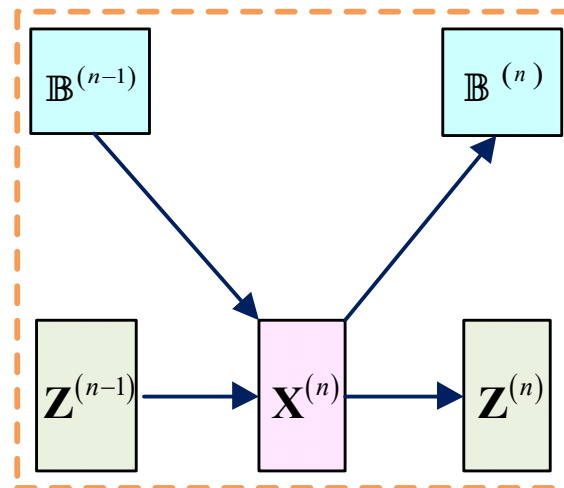


Figure 2. Reconstruction layer.

For $n = 1$, $\mathbf{Z}^{(0)}$ and $\mathbb{B}^{(0)}$ are initialized to zero matrices and thus the output is:

$$\mathbf{X}^{(1)} = -\frac{1}{1 + \rho^{(1)}} \Phi_1^H(-\mathbf{Y})(\Phi_2 \mathbf{E})^H \quad (15)$$

For $n \in [1, N]$, the output serves as the input of $\mathbf{Z}^{(n)}$ and $\mathbb{B}^{(n)}$. For $n = N + 1$, the output is adopted as the input of the loss function.

3.2. Nonlinear Transform Layer

As shown in Figure 3, the inputs of the nonlinear transform layer are $\mathbf{X}^{(n)}$ and $\mathbb{B}^{(n-1)}$, and the output is:

$$\mathbf{Z}^{(n)} = S_{PLF} \left(\mathbf{X}^{(n)} + \mathbb{B}^{(n-1)}; \left\{ p_i, q_i^{(n)} \right\}_{i=1}^{N_c} \right) \quad (16)$$

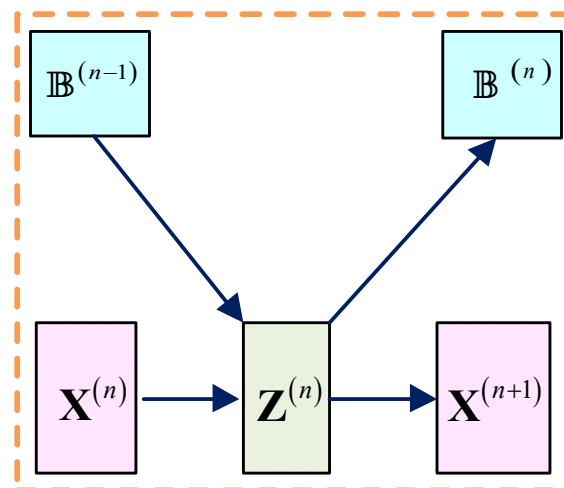


Figure 3. Nonlinear transform layer.

To learn a more flexible non-linear activation function, we substituted a piecewise linear function $S_{PLF}(\cdot)$ for the shrinkage function [29]. Specifically, $S_{PLF}(\cdot)$ is determined by N_c control points $\left\{ p_i, q_i^{(n)} \right\}_{i=1}^{N_c}$, where p_i and $q_i^{(n)}$ denote the predefined position and adjustable value of the i th point, respectively. In particular, we performed the $S_{PLF}(\cdot)$ on the real and imaginary parts of the complex signal, respectively.

For $n = 1$, $\mathbb{B}^{(0)}$ is initialized as a zero matrix and the output is:

$$\mathbf{Z}^{(1)} = S_{PLF} \left(\mathbf{X}^{(1)}; \left\{ p_i, q_i^{(1)} \right\}_{i=1}^{N_c} \right) \quad (17)$$

The output of this layer serves as the input of $\mathbf{X}^{(n+1)}$ and $\mathbb{B}^{(n)}$.

3.3. Multiplier Update Layer

As shown in Figure 4, the inputs of the multiplier update layer are $\mathbb{B}^{(n-1)}$, $\mathbf{X}^{(n)}$, and $\mathbf{Z}^{(n)}$, and the output is:

$$\mathbb{B}^{(n)} = \mathbb{B}^{(n-1)} + \eta^{(n)} \left(\mathbf{X}^{(n)} - \mathbf{Z}^{(n)} \right) \quad (18)$$

where the learning rate $\eta^{(n)}$ is an adjustable parameter.

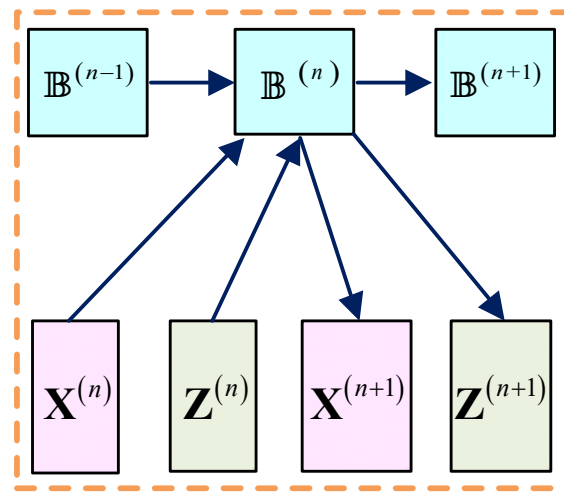


Figure 4. Multiplier update layer.

For $n = 1$, $\mathbb{B}^{(0)}$ is initialized as a zero matrix and the output is:

$$\mathbb{B}^{(1)} = \eta^{(1)} \left(\mathbf{X}^{(1)} - \mathbf{Z}^{(1)} \right) \quad (19)$$

For $n \in [1, N - 1]$, the output of this layer serves as the input of $\mathbf{X}^{(n+1)}$, $\mathbf{Z}^{(n+1)}$, and $\mathbb{B}^{(n+1)}$. For $n = N$, the output is adopted as the input of the reconstruction layer.

4. Training of 2D-ADN

4.1. Loss Function

In this article, the loss function $E(\Theta)$ is defined as the normalized mean square error (NMSE) between the network output $\hat{\mathbf{X}}$ and the label image $\mathbf{X}^{gt}$, i.e., the ground truth of the scattering center distribution:

$$E(\Theta) = \frac{1}{\gamma} \sum_{(\mathbf{Y}, \mathbf{X}^{gt}) \in \Gamma} \frac{\sqrt{\|\hat{\mathbf{X}}(\mathbf{Y}, \Theta) - \mathbf{X}^{gt}\|_F^2}}{\sqrt{\|\mathbf{X}^{gt}\|_F^2}} \quad (20)$$

where $\mathbf{Y}$ are the input echoes in the wavenumber domain defined by (14), $\Theta = \left\{ \rho^{(n)}, q_i^{(n)}, \eta^{(n)} \right\}$ is the set of the adjustable parameters, $\|\cdot\|_F$ is the Frobenius norm, and $\Gamma = \left\{ (\mathbf{Y}, \mathbf{X}^{gt}) \right\}$ is the training set with $\text{card}\{\Gamma\} = \gamma$.

4.2. Back-Propagation

We optimized the parameters of 2D-ADN utilizing the gradient-based limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) algorithm. To this end, we computed the gradients of the loss function with respect to $\Theta = \{\rho^{(n)}, q_i^{(n)}, \eta^{(n)}\}$ through back-propagation over the deep architectures. Following the structures and writing styles of the available literatures dealing with deep networks [30] and the conjugate complex derivative of composite function [31], we derived the back-propagation of the three layers, respectively. To be consistent with the previous definition, the gradients of the matrices were expressed in matrix forms for convenience, and they were also calculated in matrix forms for efficiency.

As shown in Figure 5, the gradient transferred to the reconstruction layer through $\mathbb{B}^{(n)}$ and $\mathbf{Z}^{(n)}$ for $n \in [1, N]$ satisfies:

$$\begin{aligned} \frac{\partial \mathbf{E}}{\partial \mathbf{X}^{(n)\top}} = & \operatorname{Re} \left(\frac{\partial \mathbf{E}}{\partial \mathbf{Z}^{(n)\top}} \right) \operatorname{Re} \left(\frac{\partial \mathbf{Z}^{(n)}}{\partial \mathbf{X}^{(n)\top}} \right) + j \operatorname{Im} \left(\frac{\partial \mathbf{E}}{\partial \mathbf{Z}^{(n)\top}} \right) \operatorname{Im} \left(\frac{\partial \mathbf{Z}^{(n)}}{\partial \mathbf{X}^{(n)\top}} \right) \\ & + \operatorname{Re} \left(\frac{\partial \mathbf{E}}{\partial \mathbb{B}^{(n)\top}} \right) \operatorname{Re} \left(\frac{\partial \mathbb{B}^{(n)}}{\partial \mathbf{X}^{(n)\top}} \right) + j \operatorname{Im} \left(\frac{\partial \mathbf{E}}{\partial \mathbb{B}^{(n)\top}} \right) \operatorname{Im} \left(\frac{\partial \mathbb{B}^{(n)}}{\partial \mathbf{X}^{(n)\top}} \right) \end{aligned} \quad (21)$$

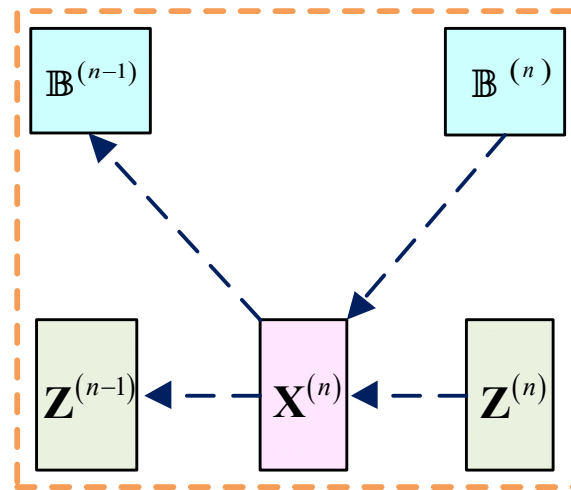


Figure 5. Back-propagation of the reconstruction layer.

For $n = N + 1$, the gradient of the loss function transferred to $\mathbf{X}^{(n)}$ equals to:

$$\frac{\partial \mathbf{E}}{\partial \mathbf{X}^{(n)\top}} = \frac{\hat{\mathbf{X}}(\mathbf{Y}', \Theta) - \mathbf{X}^{gt}}{\sqrt{\|\mathbf{X}^{gt}\|_F^2} \sqrt{\|\hat{\mathbf{X}}(\mathbf{Y}', \Theta) - \mathbf{X}^{gt}\|_F^2}} \quad (22)$$

The gradient of $\rho^{(n)}$ is:

$$\frac{\partial \mathbf{E}}{\partial \rho^{(n)}} = \sum \left(\operatorname{Re} \left(\frac{\partial \mathbf{E}}{\partial \mathbf{X}^{(n)\top}} \right) \operatorname{Re} \left(\frac{\partial \mathbf{X}^{(n)}}{\partial \rho^{(n)}} \right) \right) + \sum \left(\operatorname{Im} \left(\frac{\partial \mathbf{E}}{\partial \mathbf{X}^{(n)\top}} \right) \operatorname{Im} \left(\frac{\partial \mathbf{X}^{(n)}}{\partial \rho^{(n)}} \right) \right) \quad (23)$$

where:

$$\frac{\partial \mathbf{X}^{(n)}}{\partial \rho^{(n)}} = \frac{1}{(1 + \rho^{(n)})^2} \Phi_1^H \left(\Phi_1 \left(\mathbf{Z}^{(n-1)} - \mathbb{B}^{(n-1)} \right) \Phi_2 \mathbf{E} - \mathbf{Y} \right) (\Phi_2 \mathbf{E})^H \quad (24)$$

and the gradients transferred to $\mathbf{Z}^{(n-1)}$ and $\mathbb{B}^{(n-1)}$ are calculated by:

$$\frac{\partial \mathbf{E}}{\partial \mathbf{Z}^{(n-1)\top}} = \text{Re} \left(\frac{\partial \mathbf{E}}{\partial \mathbf{X}^{(n)\top}} \right) \text{Re} \left(\frac{\partial \mathbf{X}^{(n)}}{\partial \mathbf{Z}^{(n-1)\top}} \right) + j \text{Im} \left(\frac{\partial \mathbf{E}}{\partial \mathbf{X}^{(n)\top}} \right) \text{Im} \left(\frac{\partial \mathbf{X}^{(n)}}{\partial \mathbf{Z}^{(n-1)\top}} \right) \quad (25)$$

where:

$$\frac{\partial \mathbf{X}^{(n)}}{\partial \mathbf{Z}^{(n-1)\top}} = 1 - \frac{\Phi_1^H (\Phi_1(1) \Phi_2 \mathbf{E}) (\Phi_2 \mathbf{E})^H}{1 + \rho^{(n)}} \quad (26)$$

where 1 represents the matrix with all ones:

$$\frac{\partial \mathbf{E}}{\partial \mathbb{B}^{(n-1)\top}} = \text{Re} \left(\frac{\partial \mathbf{E}}{\partial \mathbf{X}^{(n)\top}} \right) \text{Re} \left(\frac{\partial \mathbf{X}^{(n)}}{\partial \mathbb{B}^{(n-1)\top}} \right) + j \text{Im} \left(\frac{\partial \mathbf{E}}{\partial \mathbf{X}^{(n)\top}} \right) \text{Im} \left(\frac{\partial \mathbf{X}^{(n)}}{\partial \mathbb{B}^{(n-1)\top}} \right) \quad (27)$$

where:

$$\frac{\partial \mathbf{X}^{(n)}}{\partial \mathbb{B}^{(n-1)\top}} = - \frac{\partial \mathbf{X}^{(n)}}{\partial \mathbf{Z}^{(n-1)}} \quad (28)$$

As shown in Figure 6, the gradient transferred to the nonlinear transform layer through $\mathbb{B}^{(n)}$ and $\mathbf{X}^{(n+1)}$ satisfies:

$$\begin{aligned} \frac{\partial \mathbf{E}}{\partial \mathbf{Z}^{(n)\top}} = & \text{Re} \left(\frac{\partial \mathbf{E}}{\partial \mathbb{B}^{(n)\top}} \right) \text{Re} \left(\frac{\partial \mathbb{B}^{(n)}}{\partial \mathbf{Z}^{(n)\top}} \right) + j \text{Im} \left(\frac{\partial \mathbf{E}}{\partial \mathbb{B}^{(n)\top}} \right) \text{Im} \left(\frac{\partial \mathbb{B}^{(n)}}{\partial \mathbf{Z}^{(n)\top}} \right) \\ & + \text{Re} \left(\frac{\partial \mathbf{E}}{\partial \mathbf{X}^{(n+1)\top}} \right) \text{Re} \left(\frac{\partial \mathbf{X}^{(n+1)}}{\partial \mathbf{Z}^{(n)\top}} \right) + j \text{Im} \left(\frac{\partial \mathbf{E}}{\partial \mathbf{X}^{(n+1)\top}} \right) \text{Im} \left(\frac{\partial \mathbf{X}^{(n+1)}}{\partial \mathbf{Z}^{(n)\top}} \right) \end{aligned} \quad (29)$$

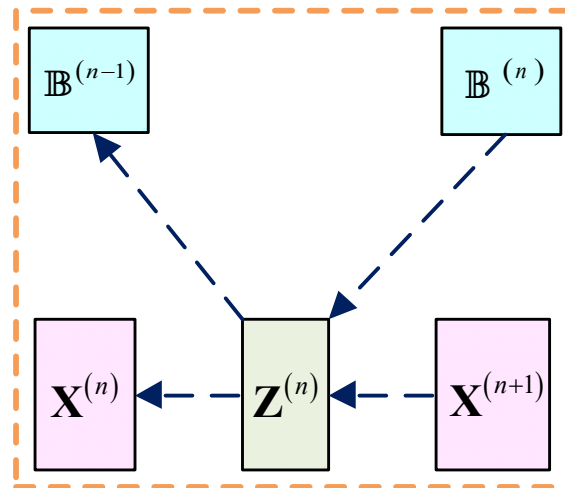


Figure 6. Back-propagation of the nonlinear transform layer.

The gradient calculations of $q_i^{(n)}$, $\mathbb{B}^{(n-1)}$, and $\mathbf{X}^{(n)}$ are consistent with the derivation of the piecewise linear function defined in [29].

As shown in Figure 7, the gradient transferred to the multiplier update layer through $\mathbf{X}^{(n+1)}$, $\mathbf{Z}^{(n+1)}$, and $\mathbb{B}^{(n+1)}$ for $n \in [1, N - 1]$ satisfies:

$$\begin{aligned} \frac{\partial \mathbf{E}}{\partial \mathbb{B}^{(n)\mathrm{T}}} = & \operatorname{Re} \left(\frac{\partial \mathbf{E}}{\partial \mathbb{B}^{(n+1)\mathrm{T}}} \right) \operatorname{Re} \left(\frac{\partial \mathbb{B}^{(n+1)}}{\partial \mathbb{B}^{(n)\mathrm{T}}} \right) + j \operatorname{Im} \left(\frac{\partial \mathbf{E}}{\partial \mathbb{B}^{(n+1)\mathrm{T}}} \right) \operatorname{Im} \left(\frac{\partial \mathbb{B}^{(n+1)}}{\partial \mathbb{B}^{(n)\mathrm{T}}} \right) \\ & + \operatorname{Re} \left(\frac{\partial \mathbf{E}}{\partial \mathbf{Z}^{(n+1)\mathrm{T}}} \right) \operatorname{Re} \left(\frac{\partial \mathbf{Z}^{(n+1)}}{\partial \mathbb{B}^{(n)\mathrm{T}}} \right) + j \operatorname{Im} \left(\frac{\partial \mathbf{E}}{\partial \mathbf{Z}^{(n+1)\mathrm{T}}} \right) \operatorname{Im} \left(\frac{\partial \mathbf{Z}^{(n+1)}}{\partial \mathbb{B}^{(n)\mathrm{T}}} \right) \\ & + \operatorname{Re} \left(\frac{\partial \mathbf{E}}{\partial \mathbf{X}^{(n+1)\mathrm{T}}} \right) \operatorname{Re} \left(\frac{\partial \mathbf{X}^{(n+1)}}{\partial \mathbb{B}^{(n)\mathrm{T}}} \right) + j \operatorname{Im} \left(\frac{\partial \mathbf{E}}{\partial \mathbf{X}^{(n+1)\mathrm{T}}} \right) \operatorname{Im} \left(\frac{\partial \mathbf{X}^{(n+1)}}{\partial \mathbb{B}^{(n)\mathrm{T}}} \right) \end{aligned} \quad (30)$$

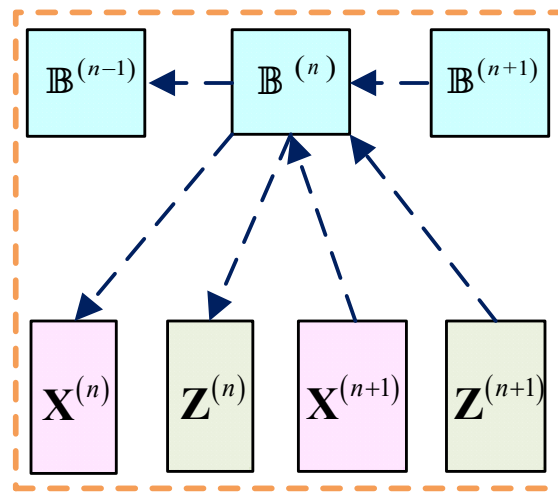


Figure 7. Back-propagation of the multiplier update layer.

For $n = N$, the gradient transferred from the last reconstruction layer to $\mathbb{B}^{(n)}$ is:

$$\frac{\partial \mathbf{E}}{\partial \mathbb{B}^{(n)\mathrm{T}}} = \operatorname{Re} \left(\frac{\partial \mathbf{E}}{\partial \mathbf{X}^{(n+1)\mathrm{T}}} \right) \operatorname{Re} \left(\frac{\partial \mathbf{X}^{(n+1)}}{\partial \mathbb{B}^{(n)\mathrm{T}}} \right) + j \operatorname{Im} \left(\frac{\partial \mathbf{E}}{\partial \mathbf{X}^{(n+1)\mathrm{T}}} \right) \operatorname{Im} \left(\frac{\partial \mathbf{X}^{(n+1)}}{\partial \mathbb{B}^{(n)\mathrm{T}}} \right) \quad (31)$$

The gradient of $\eta^{(n)}$ is:

$$\frac{\partial \mathbf{E}}{\partial \eta^{(n)}} = \sum \left(\operatorname{Re} \left(\frac{\partial \mathbf{E}}{\partial \mathbb{B}^{(n)\mathrm{T}}} \right) \operatorname{Re} \left(\frac{\partial \mathbb{B}^{(n)}}{\partial \eta^{(n)}} \right) \right) + \sum \left(\operatorname{Im} \left(\frac{\partial \mathbf{E}}{\partial \mathbb{B}^{(n)\mathrm{T}}} \right) \operatorname{Im} \left(\frac{\partial \mathbb{B}^{(n)}}{\partial \eta^{(n)}} \right) \right) \quad (32)$$

where:

$$\frac{\partial \mathbb{B}^{(n)}}{\partial \eta^{(n)}} = \mathbf{X}^{(n)} - \mathbf{Z}^{(n)} \quad (33)$$

and the gradient of this layer transferred to $\mathbb{B}^{(n-1)}$, $\mathbf{X}^{(n)}$ and $\mathbf{Z}^{(n)}$ are calculated by:

$$\frac{\partial \mathbf{E}}{\partial \mathbb{B}^{(n-1)\mathrm{T}}} = \operatorname{Re} \left(\frac{\partial \mathbf{E}}{\partial \mathbb{B}^{(n)\mathrm{T}}} \right) \operatorname{Re} \left(\frac{\partial \mathbb{B}^{(n)}}{\partial \mathbb{B}^{(n-1)\mathrm{T}}} \right) + j \operatorname{Im} \left(\frac{\partial \mathbf{E}}{\partial \mathbb{B}^{(n)\mathrm{T}}} \right) \operatorname{Im} \left(\frac{\partial \mathbb{B}^{(n)}}{\partial \mathbb{B}^{(n-1)\mathrm{T}}} \right) \quad (34)$$

where:

$$\frac{\partial \mathbb{B}^{(n)}}{\partial \mathbb{B}^{(n-1)\mathrm{T}}} = 1 \quad (35)$$

$$\frac{\partial \mathbf{E}}{\partial \mathbf{X}^{(n)\mathrm{T}}} = \operatorname{Re} \left(\frac{\partial \mathbf{E}}{\partial \mathbb{B}^{(n)\mathrm{T}}} \right) \operatorname{Re} \left(\frac{\partial \mathbb{B}^{(n)}}{\partial \mathbf{X}^{(n)\mathrm{T}}} \right) + j \operatorname{Im} \left(\frac{\partial \mathbf{E}}{\partial \mathbb{B}^{(n)\mathrm{T}}} \right) \operatorname{Im} \left(\frac{\partial \mathbb{B}^{(n)}}{\partial \mathbf{X}^{(n)\mathrm{T}}} \right) \quad (36)$$

where:

$$\frac{\partial \mathbb{B}^{(n)}}{\partial \mathbf{X}^{(n)T}} = \eta^{(n)} \mathbf{1} \quad (37)$$

$$\frac{\partial \mathbf{E}}{\partial \mathbf{Z}^{(n)T}} = \text{Re}\left(\frac{\partial \mathbf{E}}{\partial \mathbb{B}^{(n)T}}\right) \text{Re}\left(\frac{\partial \mathbb{B}^{(n)}}{\partial \mathbf{Z}^{(n)T}}\right) + j \text{Im}\left(\frac{\partial \mathbf{E}}{\partial \mathbb{B}^{(n)T}}\right) \text{Im}\left(\frac{\partial \mathbb{B}^{(n)}}{\partial \mathbf{Z}^{(n)T}}\right) \quad (38)$$

where:

$$\frac{\partial \mathbb{B}^{(n)}}{\partial \mathbf{Z}^{(n)T}} = -\eta^{(n)} \mathbf{1} \quad (39)$$

The training stage can be summarized as follows:

Step1: Define the NMSE between the network output and the label image as the loss function;

Step2: Back-propagation. Calculate the gradients of the loss function with respect to the penalty parameter, the piecewise linear function, and the learning rate of each stage;

Step3: Utilize the L-BFGS algorithm to update the network parameters according to their current values and the gradients;

Step4: Repeat Step2~Step4 until the difference between the loss functions of the two adjacent iterations is less than 10^{-6} .

For multiple training samples, we utilized the average gradient and loss function.

4.3. 2D High-Resolution ISAR Imaging Based on 2D-ADN

According to the above discussions, high-resolution 2D ISAR imaging based on 2D-ADN includes the following steps:

Step1: Training set generation. Initialize the phase error matrix $\mathbf{E} = \mathbf{I}$, construct Φ_1 and Φ_2 according to the radar parameters and data missing pattern, generate randomly distributed scattering centers $\mathbf{X}^{gt}$ with Gaussian amplitudes, calculate $\mathbf{Y}$ according to (1), and obtain the data set $\Gamma = \{(\mathbf{Y}, \mathbf{X}^{gt})\}$.

Step2: Network training. Initialize the adjustable parameters $\Theta = \{\rho^{(n)}, q_i^{(n)}, \eta^{(n)}\}$, and utilize $\{(\mathbf{Y}, \mathbf{X}^{gt})\}$ to train the network according to Section IV-B.

Step3: Testing. For simulated data, feed echoes in the wavenumber domain into the trained 2D-ADN and obtain the high-resolution image. For measured data with random phase errors, estimate the high-resolution image and random phase errors by the trained 2D-ADN and (13) iteratively until convergence.

As the distribution and the amplitudes of simulated scattering centers mimic true ISAR targets, optimal network parameters suitable for measured data imaging can be learned after network training. By this means, the issue of insufficient measured training data is effectively tackled.

In 2D-ADN the number of stages N determines the network depth. It is observed that the loss function first decreases rapidly and then tends to be stable with the increment of N . Therefore, we choose N according to the convergence condition given below:

$$\frac{\|E_N(\Theta) - E_{N-1}(\Theta)\|_F^2}{\|E_{N-1}(\Theta)\|_F^2} < \varepsilon \quad (40)$$

where $E_N(\Theta)$ is the loss function of the trained network with N stages, $E_{N-1}(\Theta)$ is the loss function of the trained network with $N - 1$ stages, and ε is a threshold.

For a single iteration, 2D-ADN and 2D-FISTA share the same computational complexity of $\mathcal{O}(UPQ)$. As the number of stages in 2D-ADN is much smaller than the number of iterations required for 2D-FISTA to converge, the computational time of 2D-ADN is shorter.

5. Experimental Results

In this section, we will demonstrate the effectiveness of 2D-ADN by high-resolution ISAR imaging of complete data, incomplete range data, incomplete azimuth data, and

2D incomplete data. The SNR of the range-compressed echoes is set to 0 dB by adding Gaussian noise, and the loss rate of the incomplete data is 50%.

For network training, 40 samples are generated following Step1 in Section IV-C, and a typical label image is shown in Figure 8a. Specifically, the first 20 samples constitute the training set and the rest constitute the test set. Later experiments will demonstrate that a small training set is adequate as unfolded deep networks have the potential to developing efficient high-performance architectures from reasonably sized training sets [22].

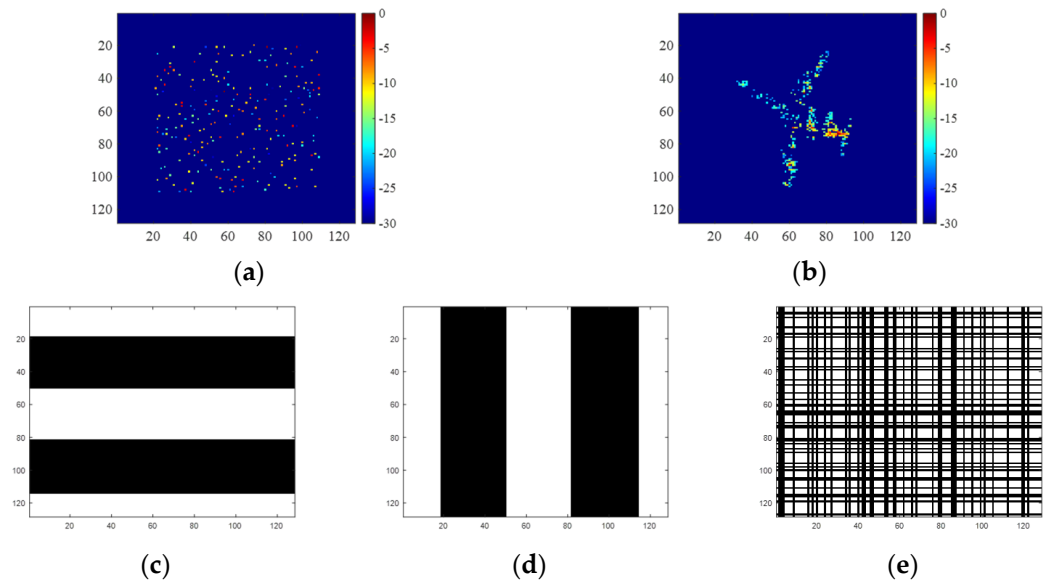


Figure 8. (a) Label image of a test sample. (b) RD image of the Yak-42 aircraft with complete data and high SNR. (c) Data missing pattern of the incomplete range data. (d) Data missing pattern of the incomplete azimuth data. (e) Data missing pattern of the 2D incomplete data.

In the training stage, the adjustable parameters are initialized as $\rho^{(n)} = 0.2$ and $\eta^{(n)} = 1$. In addition, the piecewise linear function is initialized as a soft threshold function with $\tau = 1/20$, and the control points are equally spaced with $N_c = 101$. Then, the 2D-ADN is trained following Section 4.2, and the number of stages is set to 7 according to (40).

For the simulated test data, we fed the test samples into the trained 2D-ADN, and calculated the NMSE, the peak signal-to-noise ratio (PSNR), the structure similarity index measure (SSIM), and the entropy of the image (ENT) according to the output and the label for quantitative performance evaluation. In addition, we compared the imaging results of the 2D-FISTA, untrained 2D-ADN, UNet [32], and trained 2D-ADN. In particular, as a data-driven method, the UNet has much more trainable parameters than the 2D-ADN. To avoid overfitting, we generated 1000 samples as the simulated data set, where 800 samples constituted the training set and the rest constituted the test set. The network was trained by 28 epochs, and the training time was 22 min. For 2D-ADN, the training terminated when the relative error of the loss fell below 10^{-6} , and the training time was 19 min. For 2D-FISTA, the parameters were initialized as $\lambda = 0.05$ and the algorithm terminated when the image NMSE in adjacent iterations fell below 10^{-6} .

Additionally, we fed the measured data of a Yak-42 aircraft with random phase errors into the trained 2D-ADN and obtained the high-resolution imaging in various observation conditions. The original RD image with complete data and high SNR is shown in Figure 8b. The algorithm terminated when the NMSE between adjacent iterations fell below 10^{-4} .

The imaging results were obtained with MATLAB coding without optimization, using an Intel i9-10920X 3.50-GHz computer with a 12-core processor.

5.1. Complete Data

For the test sample illustrated in Figure 8a, imaging results are shown in Figure 9 and the corresponding metrics are shown in Table 1. It was observed that the image obtained by 2D-FISTA and untrained 2D-ADN are noisy with spurious scattering centers. On the contrary, the image obtained by trained 2D-ADN has the smallest NMSE and ENT, and the highest PSNR and SSIM, demonstrating its superior imaging performance. In addition, 2D-FISTA has the longest running time due to slow convergence. The UNet obtains satisfying denoising performance and has the shortest running time of only 0.01 s.

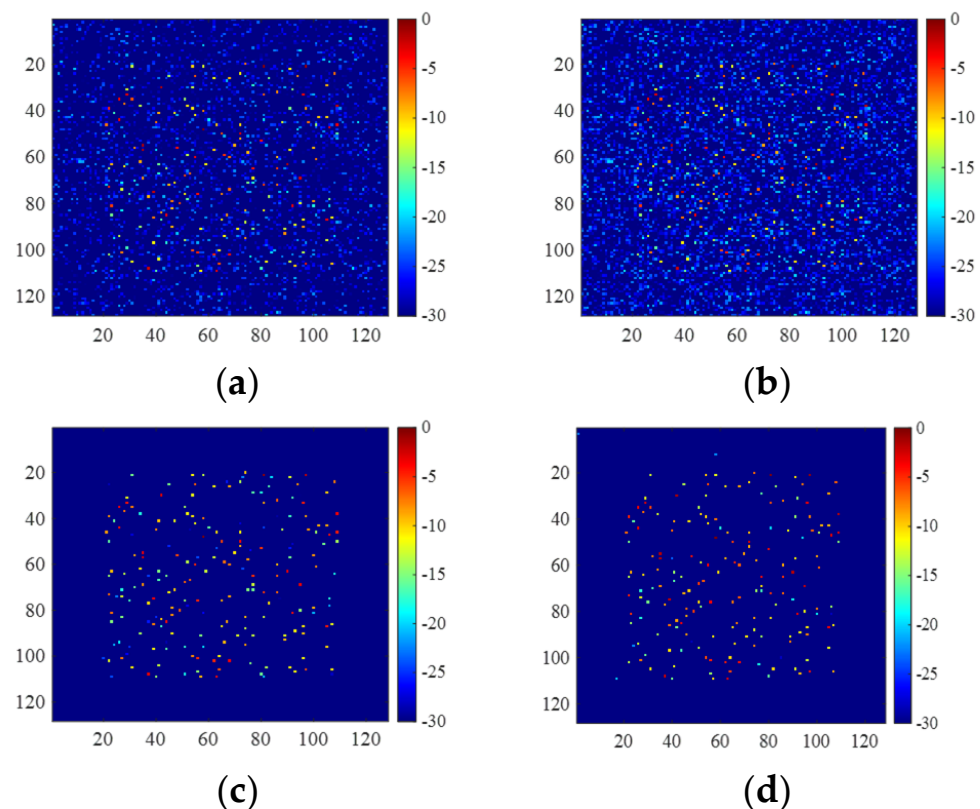


Figure 9. Images of the complete data obtained by (a) 2D-FISTA, (b) untrained 2D-ADN, (c) UNet, and (d) trained 2D-ADN.

Table 1. Quantitative performance evaluation for the complete simulation data.

Method	NMSE	PSNR	SSIM	ENT	Time (s)
2D-FISTA	0.5273	33.3754	0.4312	3.4538	0.51
Untrained 2D-ADN	0.7454	30.3146	0.2548	4.1646	0.14
UNet	0.1323	45.3112	0.9844	0.3361	0.01
Trained 2D-ADN	0.1281	45.5825	0.9858	0.3194	0.14

For measured data of the Yak-42 aircraft, imaging results are shown in Figure 10 and the corresponding entropies and running times are shown in Table 2. Compared with the available methods, the trained 2D-ADN obtained better-focused images with a clearer background. In addition, the running time increased because 2D-ADN was implemented multiple times for phase error estimation. The untrained 2D-ADN had the longest running time since the strong background noise hindered fast convergence.

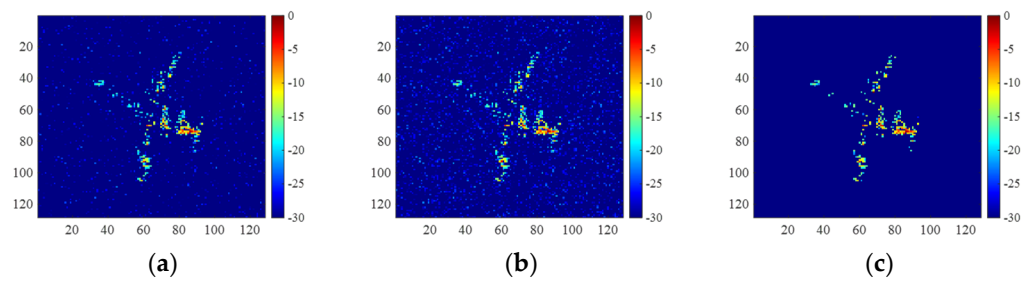


Figure 10. Yak-42 images of complete data obtained by (a) 2D-FISTA, (b) untrained 2D-ADN, and (c) trained 2D-ADN.

Table 2. Quantitative performance evaluation for the complete measured data.

Method	ENT	Time (s)
2D-FISTA	2.8055	2.81
Untrained 2D-ADN	3.8244	20.47
Trained 2D-ADN	0.2764	2.15

5.2. Incomplete Range Data

The data missing pattern of the incomplete range data is shown in Figure 8c, where the white bars denote the available echoes and the black ones denote the missing echoes.

For the same test sample, the reconstruction results are shown in Figure 11, and the metrics for quantitative comparison are shown in Table 3. Still, the trained 2D-ADN demonstrated the best reconstruction performance.

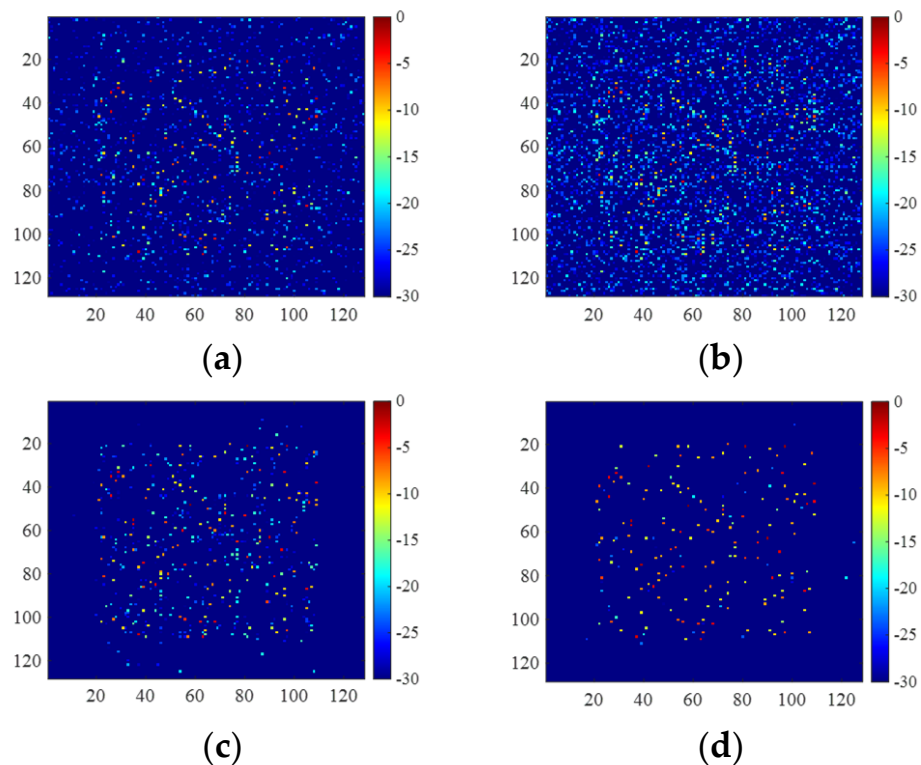
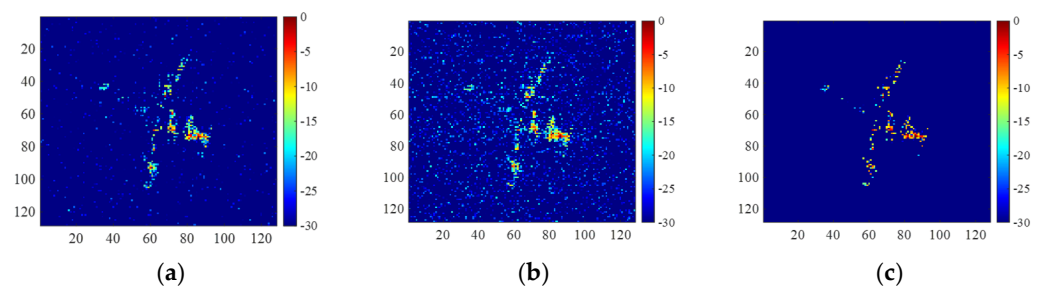


Figure 11. Images of the incomplete range data with data missing pattern shown in Figure 8c, which are obtained by (a) 2D-FISTA, (b) untrained 2D-ADN, (c) UNet, and (d) trained 2D-ADN.

Table 3. Quantitative performance evaluation for the incomplete range simulation data.

Method	NMSE	PSNR	SSIM	ENT	Time (s)
2D-FISTA	0.4575	34.6875	0.6935	2.0177	1.53
Untrained 2D-ADN	0.6922	30.9651	0.3633	3.6624	0.13
UNet	0.3606	36.6405	0.8836	1.3901	0.01
Trained 2D-ADN	0.2399	40.2297	0.9594	0.8104	0.14

For the same measured data of the Yak-42 aircraft, the reconstruction results are shown in Figure 12, where the trained 2D-ADN generated better-focused images with a clearer background than other methods. The corresponding entropies and running times are shown in Table 4, where 2D-FISTA has the longest time.

**Figure 12.** Yak-42 images of the incomplete range data obtained by (a) 2D-FISTA, (b) untrained 2D-ADN, and (c) trained 2D-ADN.**Table 4.** Quantitative performance evaluation for the incomplete range measured data.

Method	ENT	Time (s)
2D-FISTA	1.4823	43.82
Untrained 2D-ADN	3.3128	10.52
Trained 2D-ADN	0.6762	2.88

5.3. Incomplete Azimuth Data

The data missing pattern of the incomplete azimuth data is shown Figure 8d. For the same test sample, the reconstruction results are shown in Figure 13, and the metrics for quantitative comparison are shown in Table 5.

Table 5. Quantitative performance evaluation for the incomplete azimuth simulation data.

Method	NMSE	PSNR	SSIM	ENT	Time (s)
2D-FISTA	0.4468	34.9667	0.6989	2.0116	0.72
Untrained 2D-ADN	0.6822	31.0992	0.3684	3.6556	0.14
UNet	0.2993	38.2874	0.9178	1.1730	0.01
Trained 2D-ADN	0.2284	40.5901	0.9623	0.7856	0.14

For the same measured data, the imaging results are shown in Figure 14, and the corresponding entropies and running times are shown in Table 6. Similarly, 2D-ADN achieved well-focused imaging with the shortest running time.

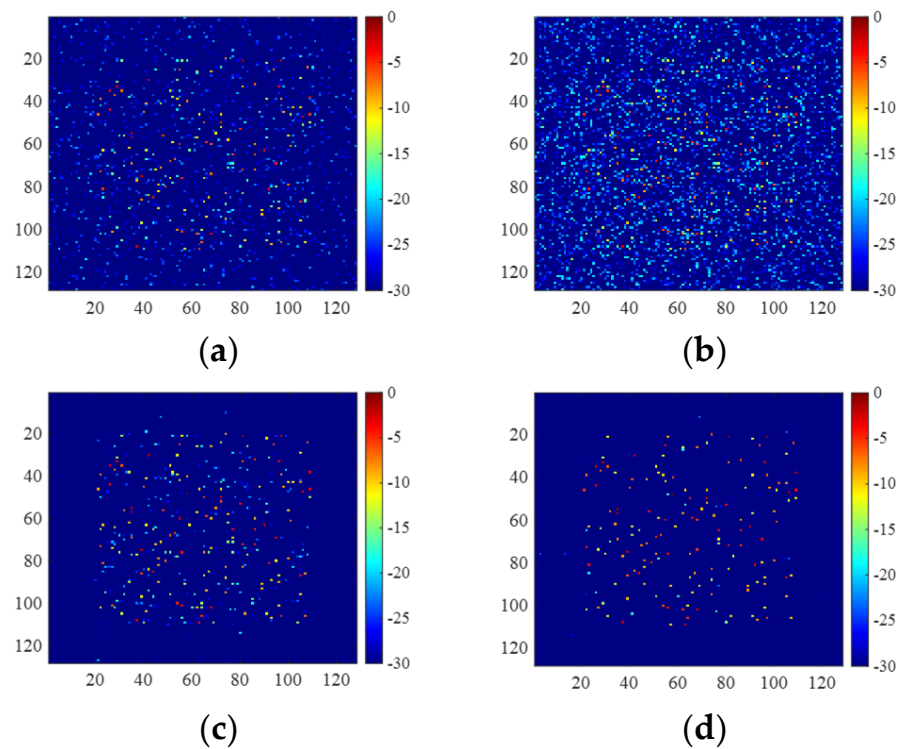


Figure 13. Images of the incomplete azimuth data with data missing pattern shown in Figure 8d, which are obtained by (a) 2D-FISTA, (b) untrained 2D-ADN, (c) UNet, and (d) trained 2D-ADN.

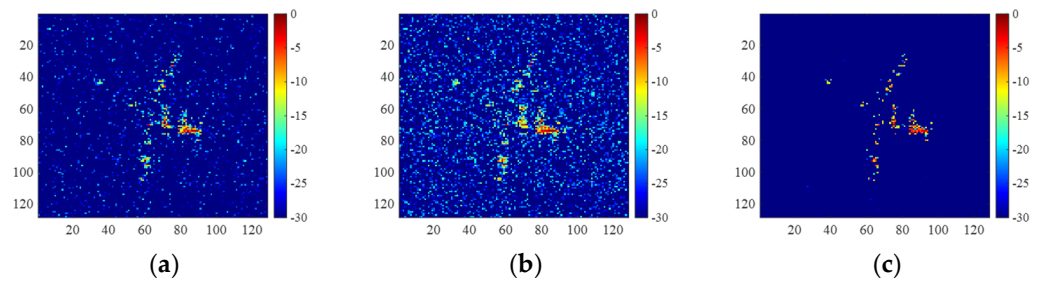


Figure 14. Yak-42 images of the incomplete azimuth data obtained by (a) 2D-FISTA, (b) untrained 2D-ADN, and (c) trained 2D-ADN.

Table 6. Quantitative performance evaluation for the incomplete azimuth measured data.

Method	ENT	Time (s)
2D-FISTA	1.9434	26.65
Untrained 2D-ADN	3.6184	8.40
Trained 2D-ADN	0.8041	3.21

5.4. 2D Incomplete Data

The data missing pattern of the 2D incomplete data is shown in Figure 8e. For the same test sample, the reconstruction results are shown in Figure 15, and the metrics for the quantitative comparison are shown in Table 7.

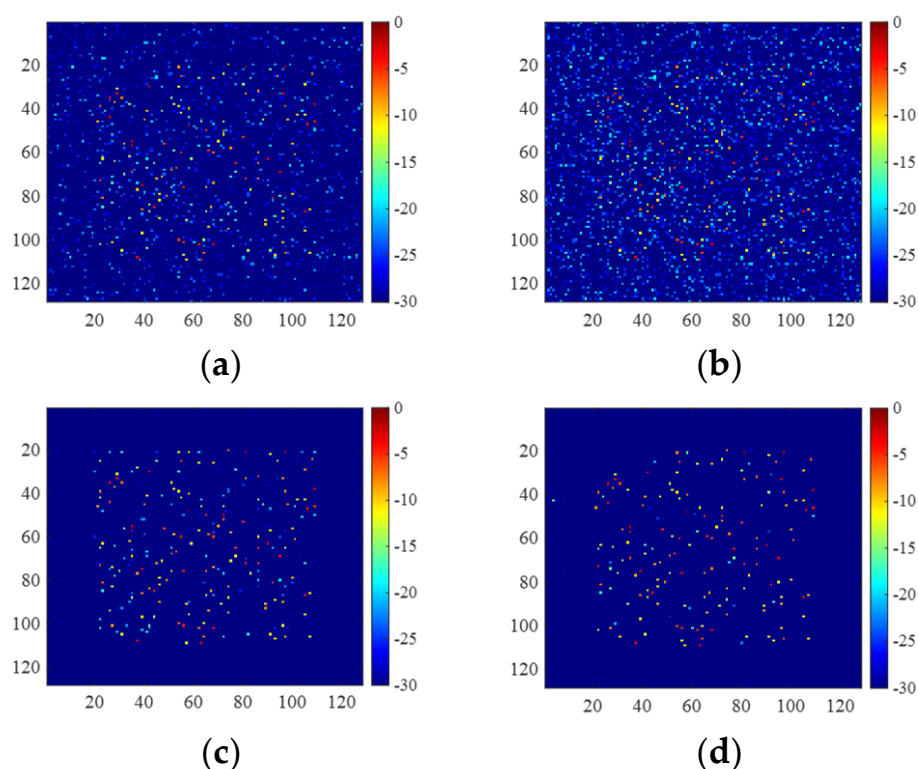


Figure 15. Images of the 2D incomplete data with data missing pattern shown in Figure 8e, which are obtained by (a) 2D-FISTA, (b) untrained 2D-ADN, (c) UNet, and (d) trained 2D-ADN.

Table 7. Quantitative performance evaluation for the 2D incomplete simulation data.

Method	NMSE	PSNR	SSIM	ENT	Time (s)
2D-FISTA	0.4438	35.0002	0.6856	2.0916	0.75
Untrained 2D-ADN	0.6397	31.6518	0.3739	3.6191	0.12
UNet	0.2876	38.5924	0.9315	0.8511	0.01
Trained 2D-ADN	0.1898	42.1850	0.9679	0.4976	0.12

For the same measured data, the reconstruction results are shown in Figure 16, and the corresponding metrics are shown in Table 8, which demonstrate the superiority of 2D-ADN over the other available methods under complex observation conditions.

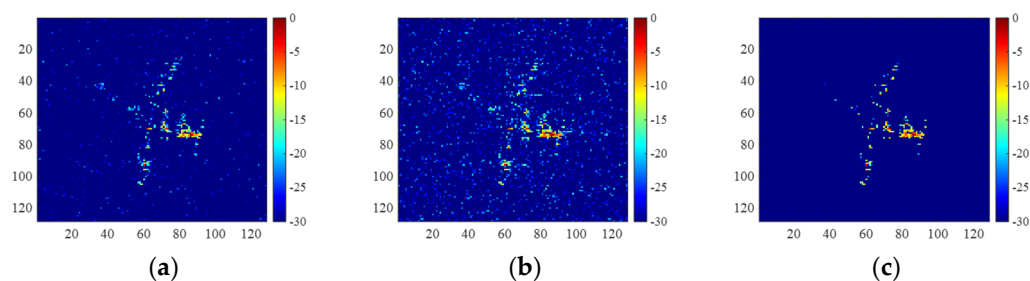


Figure 16. Yak-42 images of the 2D incomplete data obtained by (a) 2D-FISTA, (b) untrained 2D-ADN, and (c) trained 2D-ADN.

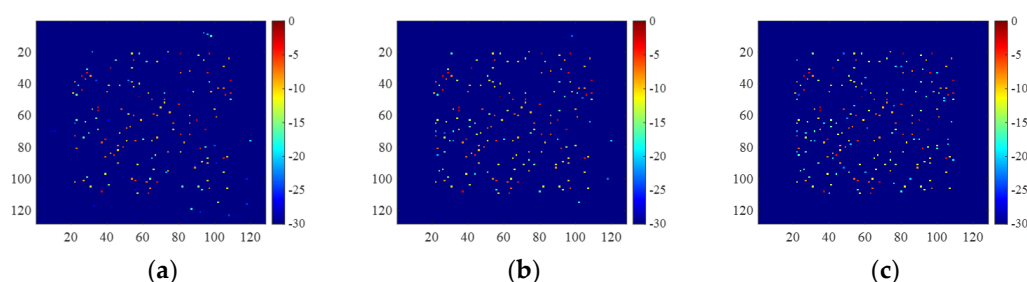
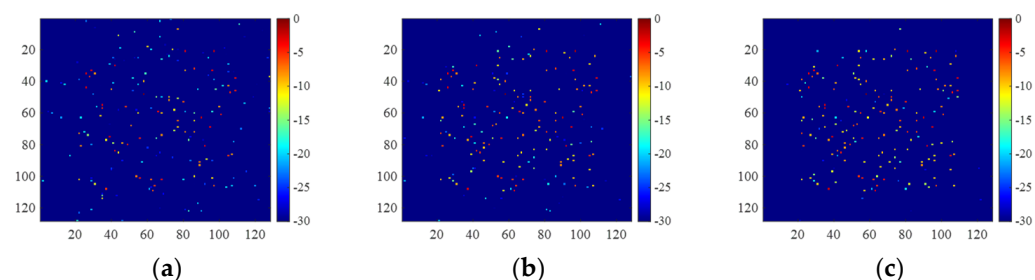
Table 8. Quantitative performance evaluation for the 2D incomplete measured data.

Method	ENT	Time (s)
2D-FISTA	1.5333	13.22
Untrained 2D-ADN	3.2766	11.70
Trained 2D-ADN	0.4747	1.68

6. Discussion

6.1. Influence of the Data Loss Rate and SNR

To further analyze the reconstruction performance of the proposed method, we designed more experiments using only 25% and 10% of the available data, where the SNRs were set to 0 dB, 5 dB, and 10 dB, respectively. The imaging results are shown in Figures 17 and 18. The corresponding metrics are shown in Tables 9 and 10.

**Figure 17.** Images generated using 25% of available data with SNRs of (a) 0 dB, (b) 5 dB, and (c) 10 dB.**Figure 18.** Images generated using 10% of available data with SNRs of (a) 0 dB, (b) 5 dB, and (c) 10 dB.**Table 9.** Quantitative performance evaluations using 25% available data.

SNR(dB)	NMSE	PSNR	SSIM	ENT	Time (s)
0	0.2976	38.2745	0.9247	1.0796	0.08
5	0.1657	43.3618	0.9715	0.6683	0.08
10	0.0837	49.3105	0.9911	0.3664	0.08

Table 10. Quantitative performance evaluations using 10% available data.

SNR(dB)	NMSE	PSNR	SSIM	ENT	Time (s)
0	0.5571	32.8445	0.7836	1.5451	0.08
5	0.3256	37.5059	0.8807	1.4266	0.08
10	0.2073	41.4280	0.9291	1.3158	0.08

It was observed that the imaging quality degraded heavily with the decrease of the available data for SNR of 0 dB. If we raised the SNR to 5 dB or 10 dB, however, the imaging performance improved rapidly. Therefore, the SNR has a greater impact on the imaging quality than the data loss rate. Furthermore, although it inherits the reconstruction performance of 2D-ADMM and utilizes a more flexible piecewise linear function as the denoiser, the 2D-ADN is still sensitive to low SNR.

6.2. Choice of the Optimal Regularization Parameter

In 2D-ADMM, it is necessary to perform multiple manual adjustments of the regularization parameter λ and penalty parameter ρ to obtain the best results. In addition, the adjustable parameters of the 2D-ADMM are fixed during iteration, which lacks flexibility. On the contrary, the 2D-ADN learns the optimal parameters of each layer separately, thus having more flexibility and a better reconstruction performance than the 2D-ADMM.

For a data loss rate of 50% and an SNR of 0dB, we obtained the optimal parameters with the minimum NSME by manual tuning, i.e., $\lambda = 0.05$ and $\rho = 0.8$. Imaging results of the optimal 2D-ADMM are shown in Figure 19 and Table 11. It was observed that the quality of the 2D-ADN image was better than the 2D-ADMM image obtained by parameter tuning.

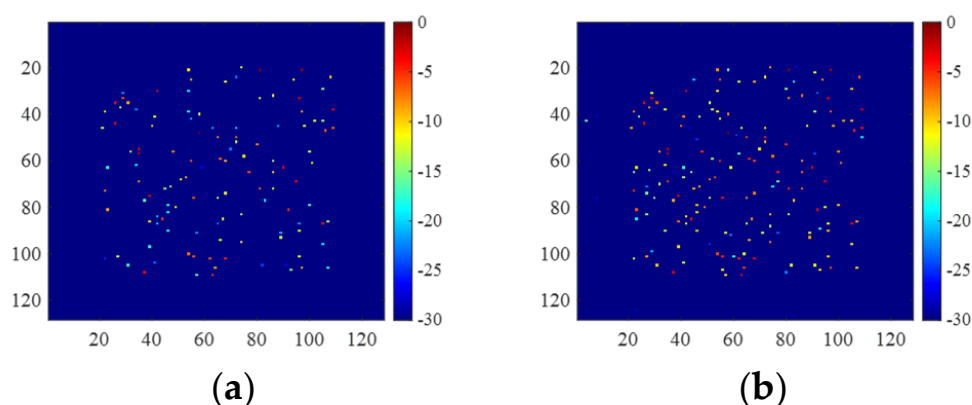


Figure 19. Images obtained by (a) optimal 2D-ADMM and (b) trained 2D-ADN.

Table 11. Quantitative performance evaluation for different methods.

Method	NMSE	PSNR	SSIM	ENT	Time (s)
Optimal 2D-ADMM	0.3231	37.8502	0.9340	0.5334	13.24
Trained 2D-ADN	0.1898	42.1850	0.9679	0.4976	0.12

6.3. Difference Between 2D-ADMM and 2D-ADN

Through the use of the unrolling method, 2D-ADN deviates from the original 2D-ADMM algorithm. Figure 20 shows the variations of NMSE for 2D-ADMM and 2D-ADN, respectively. Figure 21 shows the outputs of each stage for 2D-ADN. It was observed that the NMSE of the 2D-ADMM gradually decreased; while the NMSE of the 2D-ADN fluctuated and then reached the minimum. Therefore, end-to-end training guarantees the rapid dropping of NMSE, and the flexible network structure boosts the reconstruction performance.

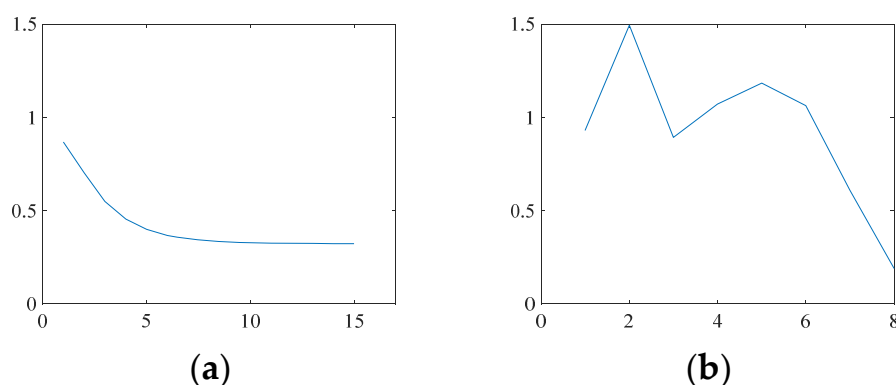


Figure 20. NMSE of (a) 2D-ADMM and (b) 2D-ADN.

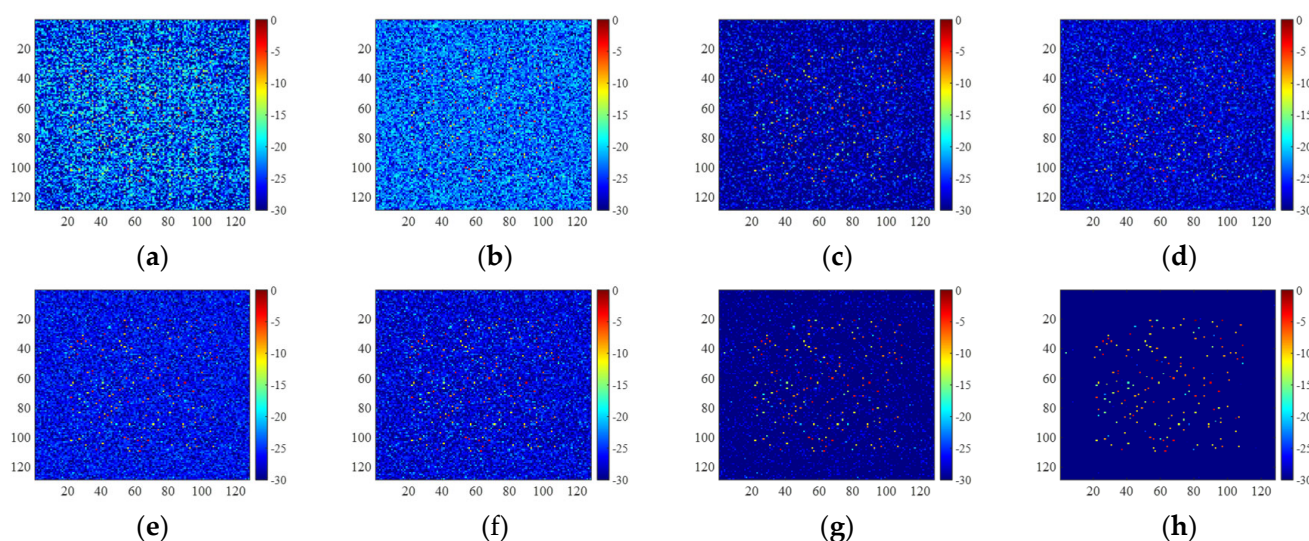


Figure 21. Images of 2D-ADN at different stages: (a) $n = 1$, (b) $n = 2$, (c) $n = 3$, (d) $n = 4$, (e) $n = 5$, (f) $n = 6$, (g) $n = 7$, (h) $n = 8$.

7. Conclusions

This article proposed 2D-ADN for high-resolution 2D ISAR imaging and autofocus under complex observational environments. Firstly, the 2D mapping from the ISAR images to echoes in the wavenumber domain was established. Then, iteration formulae based on the 2D-ADMM were derived for high-resolution ISAR imaging and, combined with the phase error estimation method, an imaging and autofocus method was proposed. On this basis, the 2D-ADMM was generalized and unrolled into an N -stage 2D-ADN, which consisted of reconstruction layers, nonlinear transform layers, and multiplier update layers. The 2D-ADN effectively tackles the parameter adjustment problem of model-driven methods and possesses more interpretability than data-driven methods. Experiments have shown that after the end-to-end training by randomly generated samples off-line, the 2D-ADN achieves the better-focused 2D imaging of measured data with random phase errors than the available methods while maintaining computational efficiency.

Future work will be focused on designing network architectures which incorporate residual translational motion compensation and 2D imaging, and on designing noise and jamming-robust network architectures in a Bayesian framework.

Author Contributions: X.L. proposed the method, designed the experiment, and wrote the manuscript; X.B. and F.Z. revised the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This paper was funded in part by the National Natural Science Foundation of China under Grant No. 61971332, 61801344, and 61631019.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kang, L.; Luo, Y.; Zhang, Q.; Liu, X.-W.; Liang, B.-S. 3-D Scattering Image Sparse Reconstruction via Radar Network. *IEEE Trans. Geosci. Remote Sens.* **2020**. [\[CrossRef\]](#)
2. Bai, X.R.; Zhou, X.N.; Zhang, F.; Wang, L.; Zhou, F. Robust pol-ISAR target recognition based on ST-MC-DCNN. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 9912–9927. [\[CrossRef\]](#)
3. Carrara, W.G.; Goodman, R.S.; Majewski, R.M. *Spotlight Synthetic Aperture Radar: Signal Processing Algorithms*; Artech House: Boston, MA, USA, 1995; Chapter 2.
4. Zhao, L.F.; Wang, L.; Yang, L.; Zoubir, A.M.; Bi, G.A. The race to improve radar imagery: An overview of recent progress in statistical sparsity-based techniques. *IEEE Signal Process. Mag.* **2016**, *33*, 85–102. [\[CrossRef\]](#)
5. Bai, X.R.; Zhang, Y.; Zhou, F. High-resolution radar imaging in complex environments based on Bayesian learning with mixture models. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 972–984. [\[CrossRef\]](#)

6. Li, R.Z.; Zhang, S.H.; Zhang, C.; Liu, Y.X.; Li, X. Deep Learning Approach for Sparse Aperture ISAR Imaging and Autofocusing Based on Complex-Valued ADMM-Net. *IEEE Sens. J.* **2021**, *21*, 3437–3451. [\[CrossRef\]](#)
7. Shao, S.; Zhang, L.; Liu, H.W. High-Resolution ISAR Imaging and Motion Compensation With 2-D Joint Sparse Reconstruction. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 6791–6811. [\[CrossRef\]](#)
8. Kang, M.; Lee, S.; Lee, S.; Kim, K. ISAR imaging of high-speed maneuvering target using gapped stepped-frequency waveform and compressive sensing. *IEEE Trans. Image Process.* **2017**, *26*, 5043–5056. [\[CrossRef\]](#)
9. Hu, P.J.; Xu, S.Y.; Wu, W.Z.; Chen, Z.P. Sparse subband ISAR imaging based on autoregressive model and smoothed l^0 algorithm. *IEEE Sens. J.* **2018**, *18*, 9315–9323. [\[CrossRef\]](#)
10. Beck, A.; Teboulle, M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imag. Sci.* **2009**, *2*, 183–202. [\[CrossRef\]](#)
11. Li, S.; Amin, M.; Zhao, G.; Sun, H. Radar imaging by sparse optimization incorporating MRF clustering prior. *IEEE Geosci. Remote Sens. Lett.* **2020**, *17*, 1139–1143. [\[CrossRef\]](#)
12. Afonso, M.V.; Bioucas-Dias, J.M.; Figueiredo, M.A.T. Fast image recovery using variable splitting and constrained optimization. *IEEE Trans. Image Process.* **2010**, *19*, 2345–2356. [\[CrossRef\]](#)
13. Bai, X.R.; Zhou, F.; Hui, Y. Obtaining JTF-signature of space-debris from incomplete and phase-corrupted data. *IEEE Trans. Aerosp. Electron. Syst.* **2017**, *53*, 1169–1180. [\[CrossRef\]](#)
14. Bai, X.R.; Wang, G.; Liu, S.Q.; Zhou, F. High-Resolution Radar Imaging in Low SNR Environments Based on Expectation Propagation. *IEEE Trans. Geosci. Remote Sens.* **2020**, *59*, 1275–1284. [\[CrossRef\]](#)
15. Li, S.Y.; Zhao, G.Q.; Zhang, W.; Qiu, Q.W.; Sun, H.J. ISAR imaging by two-dimensional convex optimization-based compressive sensing. *IEEE Sens. J.* **2016**, *16*, 7088–7093. [\[CrossRef\]](#)
16. Hashempour, H.R. Sparsity-Driven ISAR Imaging Based on Two-Dimensional ADMM. *IEEE Sens. J.* **2020**, *20*, 13349–13356. [\[CrossRef\]](#)
17. Pu, W. Deep SAR Imaging and Motion Compensation. *IEEE Trans. Image Process.* **2021**, *30*, 2232–2247. [\[CrossRef\]](#)
18. Pu, W. Shuffle GAN with Autoencoder: A Deep Learning Approach to Separate Moving and Stationary Targets in SAR Imagery. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**. [\[CrossRef\]](#)
19. Hu, C.Y.; Wang, L.; Li, Z.; Sun, L.; Loffeld, O. Inverse synthetic aperture radar imaging using complex-value deep neural network. *J. Eng.* **2019**, *2019*, 7096–7099. [\[CrossRef\]](#)
20. Zhang, J.; Ghanem, B. ISTA-Net: Interpretable optimization-inspired deep network for image compressive sensing. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 19–21 June 2018; pp. 1828–1837.
21. Ramírez, J.M.; Torre, J.I.M.; Fuentes, H.A. LADMM-Net: An Unrolled Deep Network for Spectral Image Fusion from Compressive Data. 2021. Available online: <https://arxiv.org/abs/2103.00940> (accessed on 10 June 2021).
22. Monga, V.; Li, Y.; Eldar, Y.C. Algorithm Unrolling: Interpretable, Efficient Deep Learning for Signal and Image Processing. *IEEE Signal Process. Mag.* **2021**, *38*, 18–44. [\[CrossRef\]](#)
23. Hu, C.Y.; Li, Z.; Wang, L.; Guo, J.; Loffeld, O. Inverse synthetic aperture radar imaging using a Deep ADMM Network. In Proceedings of the 20th International Radar Symposium (IRS), Ulm, Germany, 26–28 June 2019; pp. 1–9.
24. Qiu, W.; Zhao, H.; Zhou, J.; Fu, Q. High-resolution fully polarimetric ISAR imaging based on compressive sensing. *IEEE Trans. Geosci. Remote Sens.* **2014**, *52*, 6119–6131. [\[CrossRef\]](#)
25. Liu, L.; Zhou, F.; Tao, M.L.; Sun, P.G.; Zhang, Z.J. Adaptive translational motion compensation method for ISAR imaging under low SNR based on particle swarm optimization. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2015**, *8*, 5146–5157. [\[CrossRef\]](#)
26. Wang, Y.; Yang, J.; Yin, W.; Zhang, Y. A new alternating minimization algorithm for total variation image reconstruction. *SIAM J. Imag. Sci.* **2009**, *1*, 248–272. [\[CrossRef\]](#)
27. Combettes, P.; Wajs, V. Signal recovery by proximal forward-backward splitting. *Siam J. Multiscale Modeling Simul.* **2005**, *4*, 1168–1200. [\[CrossRef\]](#)
28. Zhao, L.; Wang, L.; Bi, G.A.; Yang, L. An autofocus technique for high-resolution inverse synthetic aperture radar imagery. *IEEE Trans. Geosci. Remote Sens.* **2014**, *52*, 6392–6403. [\[CrossRef\]](#)
29. Sun, J.; Li, H.B.; Xu, Z.B. Deep ADMM-Net for compressive sensing MRI. In Proceedings of the 30th International Conference on Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; pp. 10–18.
30. Yang, Y.; Sun, J.; Li, H.B.; Xu, Z.B. ADMM-CSNet: A deep learning approach for image compressive sensing. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 521–538. [\[CrossRef\]](#) [\[PubMed\]](#)
31. Petersen, K.B.; Pedersen, M.S. *The Matrix Cookbook*; Version 20121115; Technical University of Denmark: Copenhagen, Denmark, 2012; p. 24. Available online: <http://www2.compute.dtu.dk/pubdb/pubs/3274-full.html> (accessed on 10 June 2021).
32. Yang, T.; Shi, H.Y.; Lang, M.Y.; Guo, J.W. ISAR imaging enhancement: Exploiting deep convolutional neural network for signal reconstruction. *Int. J. Remote Sens.* **2020**, *41*, 9447–9468. [\[CrossRef\]](#)