



Article

Deep Unfolded Gridless DOA Estimation Networks Based on Atomic Norm Minimization

Hangui Zhu ¹ , Weike Feng ^{1,*} , Cunqian Feng ¹, Teng Ma ² and Bo Zou ¹ ¹ Air and Missile Defense College, Air Force Engineering University, Xi'an 710051, China² National Laboratory of Radar Signal Processing, Xidian University, Xi'an 710071, China

* Correspondence: fengweike007@163.com

Abstract: Deep unfolded networks have recently been regarded as an essential way to direction of arrival (DOA) estimation due to the fast convergence speed and high interpretability. However, few consider gridless DOA estimation. This paper proposes two deep unfolded gridless DOA estimation networks to resolve the above problem. We first consider the atomic norm-based 1D and decoupled atomic norm-based 2D gridless DOA models solved by the alternating iterative minimization of variables, respectively. Then, the corresponding deep networks are trained offline after constructing the corresponding complete training datasets. At last, the trained networks are applied to realize the 1D DOA and 2D estimation, respectively. Simulation results reveal that the proposed networks can secure higher 1D and 2D DOA estimation performances while maintaining a lower computational expenditure than typical methods.

Keywords: direction of arrival (DOA); sparse recovery; atomic norm; alternating iterative minimization; deep unfolding



Citation: Zhu, H.; Feng, W.; Feng, C.; Ma, T.; Zou, B. Deep Unfolded Gridless DOA Estimation Networks Based on Atomic Norm Minimization. *Remote Sens.* **2023**, *15*, 13. <https://doi.org/10.3390/rs15010013>

Academic Editor: Andrzej Stateczny

Received: 9 November 2022

Revised: 16 December 2022

Accepted: 19 December 2022

Published: 21 December 2022



Copyright: © 2022 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

The Direction of Arrival (DOA) estimation technique is widely used in radar, sonar, and other fields, and is an important research direction in array signal processing [1,2]. Typical DOA estimation methods include MUSIC (Multiple Signal Classification) [3] and ESPRIT (Estimation of Signal Parameters via Rotational Invariance Technique) [4]. Thanks to the successful application of 1D DOA estimation problems, the subspace super-resolution algorithms are successfully extended to 2D DOA estimation problems, such as 2D unitary estimation of signal parameters via rotational invariance techniques (U-ESPRIT) [5] and the 2D multiple signal classification (2D MUSIC) algorithm [6]. These algorithms can obtain high *angular* resolution under the requirement of multiple snapshot reception data and known coherent signal sources, but any of the conditions not satisfied will cause its estimation performance to degrade or even fail.

In recent years, effective alternative algorithms based on compressive sensing techniques have been introduced into the field of DOA estimation, and fruitful research results have been achieved. The traditional compressed sensing algorithms, for example, orthogonal matching pursuit (OMP) [7], sparse Bayesian learning (SBL) [8], and other algorithms, divide the possible space of the signal source into a finite number of grid points by dimension. It shows excellent estimation performance when the target angle falls exactly on the established grid, which can be applied to complex scenarios such as single snapshot, signal coherence, and missing data. On the contrary, if the real signal source does not fall on the established grid, it will cause a grid mismatch problem, and the estimation performance will be degraded or even fail. In addition, traditional compressed sensing algorithms must satisfy the pairwise isometry property (PIP) [9] and high-density gridding.

A gridless continuous compressive sensing technique based on the atomic norm theory and the Vandermonde decomposition theorem is proposed to overcome these problems,

called atomic norm minimization (ANM) [10–14]. ANM projects the Vandermonde structure of the steering matrix from the observed data to the corresponding semi-definite programming (SDP) model using the Toeplitz matrix and obtains the recovered signal by solving the SDP optimization model to achieve super-resolution. Compared with the conventional compressed sensing algorithm, it does not need to mesh the space domain, and effectively avoids the grid mismatch problem and the PIP limitation. By solving the SDP optimization model using the SDPT3-based CVX/alternating direction method of multipliers (ADMM), the ANM-based method can obtain the recovered signal to achieve super-resolution 1D DOA estimation. However, computational budgets of the CVX solver SDPT3-based method increase with the scale of the problem. The performance of the model-driven ADMM method also depends on the selected regularization parameters, and improper initial values of the parameters affect its convergence speed and accuracy.

According to Caratheodory's theory, the Vandermonde decomposition of the Toeplitz matrix does not hold in higher dimensional spaces, so the 1D ANM cannot be directly extended to the 2D DOA estimation problem. Fortunately, Chi et al. successfully apply vectorized ANM (VANM) to 2D DOA estimation with a dual Toeplitz matrix with a 2-dimensional Vandermonde structure [15]. However, the vectorization operations increase computational complexity. They cannot be applied to real-world scenarios. The dual 2D ANM starts from the dual problem of the VANM, but does not reduce the high computational costs of the VANM [16]. Luckily, Tian et al. proposed a new decoupled atomic norm minimization algorithm (DANM) [17–19] to reduce the heavy computational burden. DANM first replaces the vector atomic set in VANM with the matrix atomic set and then derives the corresponding SDP model, which naturally decouples the dual Toeplitz matrix in VANM into two Toeplitz matrices containing one-dimensional Vandermonde structures. Thereby, the 2D DOA estimation problem is converted to two 1D DOA estimation problems. This algorithm significantly reduces the computational complexity and is several orders of magnitude lower than VANM, which can also be solved by SDPT3-based CVX and ADMM algorithms [20,21]. However, ADMM algorithms are suffering from difficulties of parameter setting in practical applications. Meanwhile, inappropriate parameter settings will decrease the convergence speed and accuracy of the ADMM algorithm, thus adding to the computing complexity and degrading the DOA estimation performance. Even if the proper parameters can be chosen by theoretical analysis and cross-validation method, the fixed parameter settings fail to guarantee optimal convergence.

Recently, the Deep Unfolding/Unrolling (DU) method has been proposed as a solution to the above problems of model-driven SR algorithms inspired by deep learning techniques. With the DU method, a specific SR algorithm is unrolled into a deep neural network by taking the number of iterations of the algorithm as the number of layers and the algorithm's parameters as the network's learning parameters. SR-Net is trained on the training data set to determine the optimal parameters, thus improving the convergence speed and accuracy of the SR algorithm [22–31]. Some deep unfolding networks have successfully solved different DOA estimation problems. For example, The paper implemented fast DOA estimation under array deficiencies using the learned ISTA (LISTA) algorithm based on the iterative soft thresholding algorithm (ISTA) [32]; The paper proposed (position-enabled complex Toeplitz learned iterative shrinkage thresholding algorithm) PACT-LISTA algorithm, which integrates the intrinsic Toeplitz structure of the signal into LISTA, ignores the amplitude and phase information of the signal to be recovered and solves the mutual coupling problem between the antenna array elements using data [33]; In the paper [34], the iterative fixed-point continuation (FPC) algorithm was expanded into a deep neural network DeepFPC, to achieve single-bit DOA estimation. The paper in [35] modeled the 2D-DOA estimation problem for unmanned vehicles as a block sparse recovery problem and then used sparse bayesian learning network (SBLNet) to learn nonlinear features from data received by a large-scale (Multi-Input Multi-Output) MIMO multiple-input system or (Reconfigurable Intelligent Surface) RIS to achieve 2D-DOA and polarization parameter estimation at lower computational complexity. Our group extended the smoothed (SL0)

algorithm into SL0-Net to achieve super-resolution DOA estimation [36]. By combining model-driven and data-driven approaches effectively, these DU methods reduce computational complexity dramatically and provide high DOA performance compared with corresponding SR algorithms.

Based on the ideas above, this paper proposes the DU gridless DOA estimation method, and the main ideas and contributions are as follows:

- ① The ANM-ADMM and DANM-ADMM algorithms are implemented to solve the 1D ANM and 2D DANM gridless DOA model, respectively, based on sparse optimization theory to reduce the computational burden of the existing SDPT3-based CVX method to a certain extent.
- ② Aiming to reduce the high computational cost and tackle the problems of intractable parameter presettings of the algorithms, we unfold the algorithms to the multi-layer deep neural network named ANM-ADMM-Net and DANM-ADMM-Net, respectively, based on the DU method. They include several types of network layers and parameters. Depending on the ANM and DANM estimation models, the complete and reasonable dataset construction methods are proposed to train the network with suitable training methods to achieve fast and accurate 1D and 2D DOA estimation, respectively.
- ③ Traditional DU methods, such as FISTA and LAMP, are challenging to solve the ANM and DANM models. The proposed ANM-ADMM-Net and DANM-ADMM-Net as the novel DU-DOA framework is the only DU method that can optimize the above models and secure better gridless 1D and 2D DOA estimation performances with lower operational complexity.

The rest of this paper is structured as follows: In Section 2, the array received signal model and gridless DOA model, including the ANM and DANM models, are established briefly. Section 3 introduces the 1D ANM-ADMM and 2D DANM-ADMM algorithms first, then, the network structure, datasets construction method, and training methods of ANM-ADMM-Net and DANM-ADMM-Net are provided in detail. Section 4 verifies the performance and advantages of DU-DOA through simulation experiments compared with the SDPT3-based CVX and ADMM methods. Conclusions are drawn in Section 5.

2. Signal Model and ANM-DOA Model

2.1. 1D Signal Model and Its ANM-DOA Model

Assume that K_t far-field narrow-band signals incident into a uniform linear array (ULA) composed of M array elements. Then, the array received signal can be characterized as

$$\begin{aligned} \mathbf{Y} &= \mathbf{X} + \mathbf{N} \\ &= \sum_{k_t=1}^{K_t} \mathbf{a}(f_{k_t}) \mathbf{s}_{k_t} + \mathbf{N} \\ &= \sum_{k_t=1}^{K_t} c_{k_t} \mathbf{a}(f_{k_t}) \mathbf{b}_{k_t} + \mathbf{N} \end{aligned} \quad (1)$$

where $\mathbf{a}(f_{k_t}) = [1, \exp(j2\pi f_{k_t}), \dots, \exp(j2\pi(M-1)f_{k_t})]^T$ is the steering vector, $f_{k_t} = d \sin \theta_{k_t} / \lambda = \sin \theta_{k_t} / 2$, λ is the signal wavelength, $d = \lambda/2$ is the array spacing, θ_{k_t} is the angle between the k_t -th signal source and the array, and $[\cdot]^T$ represents the transposition; $\mathbf{s}_{k_t} = [s_{k_t1}, s_{k_t2}, \dots, s_{k_tL}] \in \mathbb{C}^{1 \times L}$ is the complex amplitude matrix; $c_{k_t} = \|\mathbf{s}_{k_t}\|_2 > 0$ and $\mathbf{b}_{k_t} = c_{k_t}^{-1} \mathbf{s}_{k_t}$ with $\|\mathbf{b}_{k_t}\|_2 = 1$; $\mathbf{N}$ is the zero-mean Gaussian matrix white noise.

According to ANM theory [10], the received signal $\mathbf{X}$ can be seen as a linear combination of some K_t atoms in the set $\mathbf{A}_M$. The set $\mathbf{A}_M$ of atoms can be defined as

$$\mathbf{A}_M \triangleq \{\mathbf{A}(f_{k_t}, \mathbf{b}_{k_t}) = \mathbf{a}(f_{k_t}) \mathbf{b}_{k_t}, f_{k_t} \in [0, 0.5]\} \quad (2)$$

Then, the atomic norm l_0 of $\mathbf{X}$ is defined as

$$\|\mathbf{X}\|_{\mathbf{A}_M, 0} = \inf_{K_t} \left\{ K_t \mid \mathbf{X} = \sum_{k_t=1}^{K_t} c_{k_t} \mathbf{A}(f_{k_t}, \mathbf{b}_{k_t}), f_{k_t} \in \mathbf{A}_M, c_{k_t} \geq 0 \right\} \quad (3)$$

So, the frequency can be recovered by dealing with the following optimization problem

$$\underset{\mathbf{X}}{\operatorname{argmin}} \|\mathbf{X}\|_{A_M,0} \text{ s.t. } \|\mathbf{Y} - \mathbf{X}\|_2^2 \leq \varepsilon \quad (4)$$

Usually, the solution of (4) can be obtained by its dual problem [34,37]. By exploiting the semi-positive, Toeplitz, and low-rank properties of the array covariance matrix $\mathbf{R} = \mathbb{E}\{\mathbf{Y}\mathbf{Y}^H\} = T(\mathbf{u})$ [10], also (4) can be formulated into the following SDP problem.

$$\min_{\mathbf{W}, \mathbf{u}, \mathbf{X}} \frac{1}{2} [\operatorname{Tr}(\mathbf{W}) + \operatorname{Tr}(T(\mathbf{u}))] \text{ s.t. } \begin{bmatrix} T(\mathbf{u}) & \mathbf{X} \\ \mathbf{X}^H & \mathbf{W} \end{bmatrix} \geq 0, \|\mathbf{Y} - \mathbf{X}\|_F^2 \leq \varepsilon \quad (5)$$

where $\mathbb{E}\{\cdot\}$ and $\operatorname{Tr}(\cdot)$ denote the expectation operator and the trace operation, and $T(\cdot)$ denotes a mapping operation from a vector $\mathbf{u} \in \mathbb{C}^{M \times 1}$ to a Hermitian Toeplitz matrix $T(\mathbf{u}) \in \mathbb{C}^{M \times M}$, and $\mathbf{W} \in \mathbb{C}^{L \times L}$ denotes an auxiliary variable matrix.

For problem (5), CVX solver SDPT3 can be utilized to secure optimal $T(\mathbf{u})$ efficiently [38]. Then, the results of DOA are estimated by Vandermonde composition of $T(\mathbf{u})$ [34,37]. Although completely avoiding gridding, the dual ANM and ANM-CVX methods have extensive computational complexity, failing to meet real-time requirements.

2.2. 2D Signal Model and Its DANM-DOA Model

Consider K_t spatial far-field narrowband signals acting on a $N \times M$ uniform rectangular array (URA) with array element spacing of half wavelength, as shown in Figure 1. The pitch angle ϕ_{k_t} and azimuth angle θ_{k_t} of the k_t -th incident signals are related to the angles α_{k_t} and β_{k_t} between the x and y axes as follows:

$$\phi_{k_t} = \arcsin(\sin 2(\alpha_{k_t}) + \sin 2(\beta_{k_t})) \quad (6)$$

$$\theta_{k_t} = \arctan\left(\frac{\sin(\beta_{k_t})}{\sin(\alpha_{k_t})}\right) \quad (7)$$

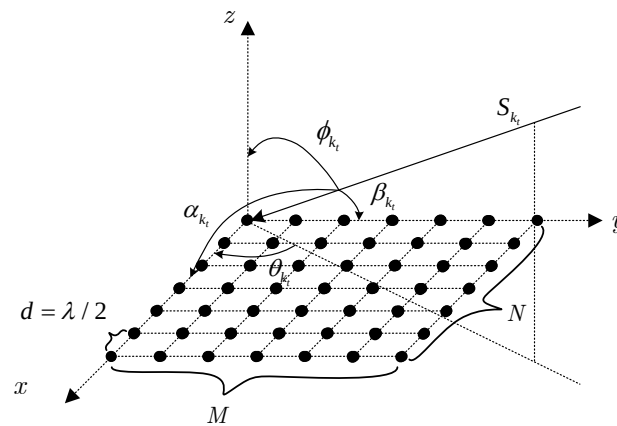


Figure 1. URA signal model.

By finding the angles α_{k_t} and β_{k_t} , we can obtain θ_{k_t} and ϕ_{k_t} according to Equations (6) and (7). Therefore, this paper uses α_{k_t} and β_{k_t} for signal modeling analysis. Then, the array guiding vector, the guiding matrix of dimensions x and y are

$$\begin{cases} \mathbf{a}_x(w_{x,k_t}) = [1, e^{-jw_{x,k_t}}, \dots, e^{-j(N-1)w_{x,k_t}}]^T \\ \mathbf{a}_y(w_{y,k_t}) = [1, e^{-jw_{y,k_t}}, \dots, e^{-j(M-1)w_{y,k_t}}]^T \end{cases} \quad (8)$$

$$\begin{cases} \mathbf{A}_x = [\mathbf{a}_x(w_{x,1}), \mathbf{a}_x(w_{x,2}), \dots, \mathbf{a}_x(w_{x,K_t})] \\ \mathbf{A}_y = [\mathbf{a}_y(w_{y,1}), \mathbf{a}_y(w_{y,2}), \dots, \mathbf{a}_y(w_{y,K_t})] \end{cases} \quad (9)$$

The single snapshot data can be expressed as:

$$\mathbf{X} = \sum_{k_t=1}^{K_t} \mathbf{s}_{k_t} \mathbf{a}_x(w_{x,k_t}) \mathbf{a}_y^T(w_{y,k_t}) = \mathbf{A}_x \mathbf{S} \mathbf{A}_y^T \quad (10)$$

where $\mathbf{S} = \text{diag}(s_1, s_2, \dots, s_{K_t})$ is a diagonal amplitude matrix, and s_{k_t} is the amplitude of the k_t -th target signal corresponding to the true frequencies of interest $w_{x,k_t} = \pi \sin \phi_{k_t} \cos \theta_{k_t} = \pi \sin \alpha_{k_t}$ and $w_{y,k_t} = \pi \sin \phi_{k_t} \sin \theta_{k_t} = \pi \sin \beta_{k_t}$. Two dimensional frequencies sets are $\mathbf{w}_x = (w_{x,1}, w_{x,2}, \dots, w_{x,K_t})$, $\mathbf{w}_y = (w_{y,1}, w_{y,2}, \dots, w_{y,K_t})$, and the corresponding angles are $\hat{\boldsymbol{\alpha}} = (\alpha_1, \alpha_2, \dots, \alpha_{K_t})$, $\hat{\boldsymbol{\beta}} = (\beta_1, \beta_2, \dots, \beta_{K_t})$.

2D DOA estimation means that w_{x,k_t} and w_{y,k_t} are first recovered from the observations data $\mathbf{X}$, then arriving at the angles α_{k_t} and β_{k_t} for acquiring pitch angle ϕ_{k_t} and azimuth angle θ_{k_t} .

According to [17], the set of atoms $\mathbf{A}_V$ in a matrix form of the received signal data $\mathbf{X}$ can be expressed as

$$\begin{aligned} \mathbf{A}_M &= \left\{ \mathbf{a}_x(w_{x,k_t}) \mathbf{a}_y^T(w_{y,k_t}), w_{x,k_t}, w_{y,k_t} \in [-\pi, \pi] \right\} \\ &= \{ \mathbf{A}_\gamma, \gamma \in [-\pi, \pi] \times [-\pi, \pi] \} \end{aligned} \quad (11)$$

where each atom is a rank 1 matrix. The atomic norm $\mathbf{l}_1$ of $\mathbf{X}$ is defined as

$$\|\mathbf{X}\|_{\mathbf{A}_M} = \inf \left\{ \sum_{k_t}^{K_t} |\mathbf{s}_{k_t,p}| \mid \mathbf{X} = \sum_{k_t}^{K_t} \mathbf{A}(\gamma) \mathbf{s}_{k_t}, \mathbf{A}(\gamma) \in \mathbf{A}_M \right\} \quad (12)$$

Define the one-level Toeplitz matrices $T(\mathbf{u}_x)$ and $T(\mathbf{u}_y)$ constructed by their first rows $\mathbf{u}_x \in \mathbb{C}^{M \times 1}$, and $\mathbf{u}_y \in \mathbb{C}^{N \times 1}$, respectively. The problem of minimizing Equation (12) can be transformed into addressing the following semidefinite programming (SDP) problem [19]

$$\begin{aligned} &\underset{\mathbf{u}_x, \mathbf{u}_y, \mathbf{X}}{\text{argmin}} \frac{\lambda}{2\sqrt{NM}} \text{Tr}(T(\mathbf{u}_x)) + \text{Tr}(T(\mathbf{u}_y)) \\ &\text{s.t. } \boldsymbol{\Theta} = \begin{bmatrix} T(\mathbf{u}_x) & \mathbf{X}^H \\ \mathbf{X} & T(\mathbf{u}_y) \end{bmatrix} \geq 0, \|\mathbf{Y} - \mathbf{X}\|_F^2 \leq \varepsilon \end{aligned} \quad (13)$$

where $\mathbf{Y}$ denotes the received data with noise.

After obtaining $T(\mathbf{u}_x)$ and $T(\mathbf{u}_y)$, the x and y -dimensional DOA can be obtained by the following decomposition

$$T(\mathbf{u}_x) = \mathbf{A}_x \mathbf{D}_x \mathbf{A}_x^H, T(\mathbf{u}_y) = \mathbf{A}_y \mathbf{D}_y \mathbf{A}_y^H \quad (14)$$

where $\mathbf{D}_x$ and $\mathbf{D}_y$ are diagonal matrices. The final 2D DOA is obtained by the pairing procedure after obtaining the DOA in each dimension [39].

For the DANM optimization model (13), an off-the-shelf CVX solver SDPT3 can be utilized to secure $T(\mathbf{u}_x)$ and $T(\mathbf{u}_y)$, where the dimensionality of the semi-definite constraint matrix is $(N + M) \times (N + M)$ [21]. The above directly leads to high computational costs when the dimensions N and M are unacceptably large. Although the ADMM algorithm in [20,40] is an alternative to solving (13), computational load still needs reconsidered.

3. DU-DOA Estimation

In order to lighten the computational load and enhance the estimation performance efficiently, this paper intends to the DU-DOA method to solve (5) for 1D DOA and (13) for the 2D DOA estimations in this section. According to [22], the DU approach can be seen as an iterative step based on the SR algorithm for designing the structure and parameters of the deep neural network. DU methods such as LISTA, LAMP, and LePOM [22–24] can all theoretically achieve the 2D DOA estimation. However, these methods are unsuitable for solving the AMM-DOA and DAMM-DOA models, i.e., (5) and (13). This paper analyzes

the ANM-ADMM and the DANM-ADMM algorithms to address the above problem in Sections 3.1 and 3.3, respectively. The algorithms are expanded into the deep neural network ANM-ADMM-Net and DANM-ADMM-Net to achieve fast and accurate 1D and 2D DOA estimation in Sections 3.2 and 3.4, respectively.

3.1. 1D ANM-ADMM DOA

To facilitate the solution, (5) is rewritten as (15).

$$\begin{aligned} [\mathbf{X}, \mathbf{u}] &= \underset{\mathbf{X}, \mathbf{W}, \mathbf{u}, \boldsymbol{\Theta}}{\operatorname{argmin}} \frac{\tau}{2} [\operatorname{Tr}(\mathbf{W}) + \operatorname{Tr}(T(\mathbf{u}))] + \frac{1}{2} \|\mathbf{Y} - \mathbf{X}\|_{\text{F}}^2 \\ \text{s.t. } \boldsymbol{\Theta} &= \begin{bmatrix} T(\mathbf{u}) & \mathbf{X} \\ \mathbf{X}^H & \mathbf{W} \end{bmatrix} \geq 0 \end{aligned} \quad (15)$$

where $\tau \in \mathbb{R}$ is the regularization factor.

The augmented Lagrangian function of the problem (15) can be defined via [41].

$$\begin{aligned} &\underset{\mathbf{X}, \mathbf{W}, \mathbf{u}, \boldsymbol{\Theta} \geq 0, \boldsymbol{\Lambda}}{\operatorname{argmin}} \frac{\tau}{2} [\operatorname{Tr}(\mathbf{W}) + \operatorname{Tr}(T(\mathbf{u}))] + \frac{1}{2} \|\mathbf{Y} - \mathbf{X}\|_{\text{F}}^2 \\ &+ \left\langle \boldsymbol{\Lambda}, \boldsymbol{\Theta} - \begin{bmatrix} T(\mathbf{u}) & \mathbf{X} \\ \mathbf{X}^H & \mathbf{W} \end{bmatrix} \right\rangle + \frac{\rho}{2} \left\| \boldsymbol{\Theta} - \begin{bmatrix} T(\mathbf{u}) & \mathbf{X} \\ \mathbf{X}^H & \mathbf{W} \end{bmatrix} \right\|_{\text{F}}^2 \\ &= \underset{\mathbf{X}, \mathbf{W}, \mathbf{u}, \boldsymbol{\Theta} \geq 0, \boldsymbol{\Lambda}}{\operatorname{argmin}} \frac{\tau}{2} [\operatorname{Tr}(\mathbf{W}) + \operatorname{Tr}(T(\mathbf{u}))] + \frac{1}{2} \|\mathbf{Y} - \mathbf{X}\|_{\text{F}}^2 + \Delta \end{aligned} \quad (16)$$

where $\boldsymbol{\Lambda} \in \mathbb{C}^{(M+L) \times (M+L)}$ is the Lagrangian multiplier, and $\rho > 0$ is the penalty factor.

Note (16) is an unconstrained optimization problem, so that given the received signal $\mathbf{Y}$, unknown signal components $\mathbf{X}$, and vector $\mathbf{u}$ can be estimated by alternatively minimizing the cost function of $\mathbf{X}$, $\mathbf{W}$, $\mathbf{u}$, $\boldsymbol{\Theta}$ and $\boldsymbol{\Lambda}$ [14]. The specific iterative process can be found in Appendix A.

In conclusion, the ADMM algorithm for solving (5) is provided in Algorithm 1.

Algorithm 1: ANM-ADMM-DOA algorithm.

Input: $\mathbf{Y}$, iterations K , penalty factor ρ , regularization factor τ .

Initialization: $\boldsymbol{\Theta}^{(0)} = \mathbf{0}_{M+L}$ and $\boldsymbol{\Lambda}^{(0)} = \mathbf{0}_{M+L}$.

For $k = 0 : K - 1$ **do**

(1) $\mathbf{X}^{(k+1)} = 1/(1 + 2\rho)(\mathbf{Y} + 2\boldsymbol{\Lambda}_X^{(k)} + 2\rho\boldsymbol{\Theta}_X^{(k)})$;

(2) $\mathbf{W}^{(k+1)} = \rho^{-1}\boldsymbol{\Lambda}_W^{(k)} + \boldsymbol{\Theta}_W^{(k)} - \tau/(2\rho)\mathbf{I}_L$;

(3) $\mathbf{u}^{(k+1)} = \Gamma(T^*(\rho^{-1}\boldsymbol{\Lambda}_{T(\mathbf{u})}^{(k)} + \boldsymbol{\Theta}_{T(\mathbf{u})}^{(k)}) - \tau/(2\rho)\mathbf{M}\mathbf{e}_1)$;

(4) $\begin{cases} \hat{\boldsymbol{\Theta}}^{(k+1)} = \begin{bmatrix} T(\mathbf{u}^{(k+1)}) & \mathbf{X}^{(k+1)} \\ (\mathbf{X}^{(k+1)})^H & \mathbf{W}^{(k+1)} \end{bmatrix} - \rho^{-1}\boldsymbol{\Lambda}^{(k)} \\ \hat{\boldsymbol{\Theta}}^{(k+1)} = \mathbf{G}\operatorname{diag}(\{\delta_g\})\mathbf{G}^{-1}, \boldsymbol{\Theta}^{(k+1)} = \mathbf{G}\operatorname{diag}(\{\delta_g\}_+) \mathbf{G}^{-1} \end{cases};$

(5) $\boldsymbol{\Lambda}^{(k+1)} = \boldsymbol{\Lambda}^{(k)} + \rho \left(\boldsymbol{\Theta}^{(k+1)} - \begin{bmatrix} T(\mathbf{u}^{(k+1)}) & \mathbf{X}^{(k+1)} \\ (\mathbf{X}^{(k+1)})^H & \mathbf{W}^{(k+1)} \end{bmatrix} \right)$;

End

Output: Recovered signal of interest $\mathbf{X}^{(K)}$, the optimal estimate $\mathbf{u}^{(K)}$.

Then: Estimate f_{k_t} and s_{k_t} by Vandermonde decomposition of $T(\mathbf{u}^{(K)}\sqrt{M}/2)$ and compute DOA by $\theta_{k_t} = \operatorname{asind}(2f_{k_t})$.

According to the above derivation, the parameters of the model-driven ADMM algorithm, including the penalty factor ρ and the regularization factor τ , need to be set in advance, which is a challenge for practical applications. Meanwhile, inappropriate parameter settings will decrease the convergence speed and accuracy of the ADMM algorithm, thus adding to the computing complexity of (5) and degrading the DOA estimation performance. Even if the proper parameters can be chosen by theoretical analysis and cross-validation method [41], the fixed parameter settings fail to guarantee the optimal

convergence of the ANM-ADMM algorithm. Based on the idea of the DU method, this paper expands the algorithm into a deep neural network ANM-ADMM-Net and learns its optimal parameters from the constructed data obeying a particular distribution, thus solving the above problems.

It is necessary to note that the optimal estimates $T(\mathbf{u}^{(K)})$ of the algorithm are the critical points of the following estimation. Therefore, we can consider the output as a label and construct a proper loss function in the next subsection.

3.2. 1D ANM-ADMM-Net DOA

For the optimization problem shown in (5), when the system parameters are given and the complex amplitude and noise all obey a particular distribution, the received data $\mathbf{Y}$ will also have a particular distribution. At this point, assume that an optimal set of parameter sequences exists so that (5) can be solved quickly and accurately by the ADMM algorithm for all received signals, with the DOA obeying a specific distribution. Therefore, this subsection constructs the network ANM-ADMM-Net to tackle the problems of the ADMM algorithm based on its iterative steps. This network couples the interpretability of the model-driven algorithm and the nonlinear fitting capability of the data-driven deep learning method. Training networks based on a sufficient and complete training data set can obtain optimal iterative parameters, thereby reducing the number of iterations and further enabling higher speed and ameliorated performance of DOA estimation. In the following, the four parts of ANM-ADMM-Net: network structures, dataset construction, network initialization, and training are described thoroughly.

3.2.1. Network Structure

According to the steps in Algorithm 1, the ANM-ADMM algorithm can be mapped to a K layer network ANM-ADMM-Net shown in Figure 2, whose inputs are $\mathbf{Y}$, $\Theta^{(0)} = \mathbf{0}_{M+L}$, and $\Lambda^{(0)} = \mathbf{0}_{M+L}$, and the learnable parameters are $\Omega = \{\Omega^{(k+1)}\}_{k=0}^{K-1} = \{\rho_{k+1}, \tau_{k+1}, \eta_{k+1}\}_{k=0}^{K-1}$, to arrive at the signal components $\mathbf{X}^{(K)}$ and the Toeplitz matrix $T(\mathbf{u}^{(K)})$. The $(k+1)$ -th layer operation of ANM-ADMM-Net can be expressed as

$$[\mathbf{X}^{(k+1)}, \mathbf{u}^{(k+1)}] = F_{k+1}\{\mathbf{Y}, \mathbf{X}^{(k)}, \mathbf{W}^{(k)}, \mathbf{u}^{(k)}, \theta^{(k)}, \Lambda^{(k)}, \Omega^{(k+1)}\} \quad (17)$$

where $F_{k+1}\{\cdot\}$ contains five main structural sub-layers, including reconstruction sub-layer $\mathbf{A}^{(k+1)}$, the auxiliary variable update sub-layer $\mathbf{B}^{(k+1)}$, the Toeplitz transform sub-layer $\mathbf{C}^{(k+1)}$, the nonlinear sub-layer $\mathbf{D}^{(k+1)}$, and the multiplier update sub-layer $\mathbf{E}^{(k+1)}$ corresponding to (17). The specific description is as follows.

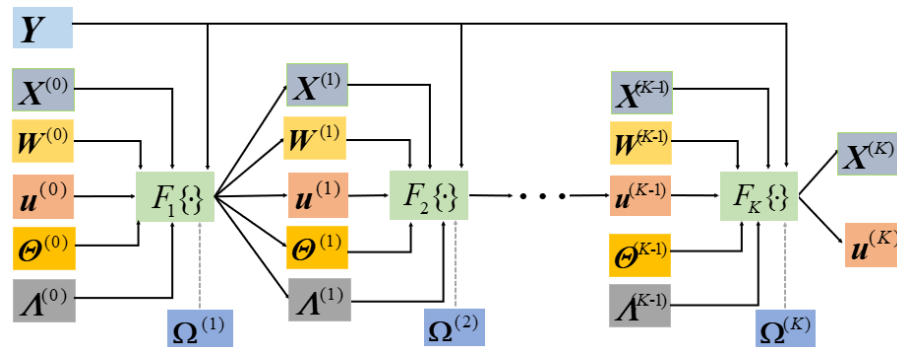


Figure 2. The network structure of ANM-ADMM-Net.

- (1) Reconstruction sub-layer $\mathbf{A}^{(k+1)}$: Taking the output $\Lambda_x^{(k)}$ of the sub-layer $\mathbf{E}^{(k)}$ and $\Theta_x^{(k)}$ of the sub-layer $\mathbf{D}^{(k)}$ of k -th layer and received signal $\mathbf{Y}$ as the input, the output $\mathbf{X}^{(k+1)}$ is updated as

$$\mathbf{X}^{(k+1)} = 1/(1 + 2\rho_{k+1})(\mathbf{Y} + 2\Lambda_{\mathbf{X}}^{(k)} + 2\rho_{k+1}\Theta_{\mathbf{X}}^{(k)}) \quad (18)$$

where ρ_{k+1} is the learnable parameter. The output $\mathbf{X}^{(k+1)}$ of $\mathbf{A}^{(k+1)}$ can be used as the input of the sub-layer $\mathbf{D}^{(k+1)}$ and $\mathbf{E}^{(k+1)}$ in the $(k+1)$ -th layer.

- (2) Auxiliary variable update sub-layer $\mathbf{B}^{(k+1)}$: Taking the output $\Theta_{\mathbf{W}}^{(k)}$ of the sub-layer $\mathbf{D}^{(k+1)}$ and $\Lambda_{\mathbf{W}}^{(k)}$ of the sub-layer $\mathbf{E}^{(k)}$ in the k -th layer are used as inputs, the output of $\mathbf{B}^{(k+1)}$ is given by

$$\mathbf{W}^{(k+1)} = \rho_{k+1}^{-1}\Lambda_{\mathbf{W}}^{(k)} + \Theta_{\mathbf{W}}^{(k)} - \tau_{k+1}/(2\rho_{k+1})\mathbf{I}_L \quad (19)$$

where τ_{k+1} is the learnable parameter. The output $\mathbf{W}^{(k+1)}$ is the input of the sub-layer $\mathbf{D}^{(k+1)}$ and $\mathbf{E}^{(k+1)}$ in the $(k+1)$ -th layer.

- (3) Toeplitz transform sub-layer $\mathbf{C}^{(k+1)}$: The output $\Theta_{T(u)}^{(k)}$ of the sub-layer $\mathbf{D}^{(k+1)}$ and $\Lambda_{T(u)}^{(k)}$ of the sub-layer $\mathbf{E}^{(k)}$ in the k -th layer are used as inputs, and the output is represented by

$$\mathbf{u}^{(k+1)} = \Gamma(T^*(\rho_{k+1}^{-1}\Lambda_{T(u)}^{(k)} + \Theta_{T(u)}^{(k)}) - \tau_{k+1}/(2\rho_{k+1})\mathbf{M}\mathbf{e}_1) \quad (20)$$

where $\mathbf{u}^{(k+1)}$ can be the input of the sub-layer $\mathbf{D}^{(k+1)}$ and $\mathbf{E}^{(k+1)}$ in the $(k+1)$ -th layer.

- (4) Nonlinear sub-layer $\mathbf{D}^{(k+1)}$: Taking the output $\Lambda_{T(u)}^{(k)}$ of the sub-layer $\mathbf{E}^{(k)}$ of the k -th layer, the output $\mathbf{X}^{(k+1)}$ of the sub-layer $\mathbf{A}^{(k)}$, and the output $\mathbf{u}^{(k+1)}$ of the sub-layer $\mathbf{B}^{(k+1)}$ as the input, the output of $\mathbf{D}^{(k+1)}$ can be given as

$$\begin{cases} \hat{\Theta}^{(k+1)} = \begin{bmatrix} T(\mathbf{u}^{(k+1)}) & \mathbf{X}^{(k+1)} \\ (\mathbf{X}^{(k+1)})^H & \mathbf{W}^{(k+1)} \end{bmatrix} - \rho_{k+1}^{-1}\Lambda^{(k)} \\ \hat{\Theta}^{(k+1)} = \mathbf{G}\text{diag}(\{\delta_g\})\mathbf{G}^{-1}, \Theta^{(k+1)} = \mathbf{G}\text{diag}(\{\delta_g\}_+) \mathbf{G}^{-1} \end{cases} \quad (21)$$

where the output $\Theta^{(k+1)}$ of $\mathbf{D}^{(k+1)}$ can be used as the input of the sub-layer $\mathbf{A}^{(k+2)}$, $\mathbf{B}^{(k+2)}$, and $\mathbf{C}^{(k+2)}$ in the $(K+2)$ -th layer.

- (5) Multiplier update sub-layer $\mathbf{E}^{(k+1)}$: Taking the output $\Lambda^{(k)}$ of the sub-layer $\mathbf{D}^{(k)}$ of the k -th layer, the output $\mathbf{X}^{(k+1)}$, $\mathbf{W}^{(k+1)}$, $\mathbf{u}^{(k+1)}$ and $\Theta^{(k+1)}$ of the sub-layer $\mathbf{A}^{(k+1)}$, $\mathbf{B}^{(k+1)}$, $\mathbf{C}^{(k+1)}$ and $\mathbf{D}^{(k+1)}$ respectively as input, the output of $\mathbf{E}^{(k+1)}$ can be updated by

$$\Lambda^{(k+1)} = \Lambda^{(k)} + \eta_{k+1} \left(\Theta^{(k+1)} - \begin{bmatrix} T(\mathbf{u}^{(k+1)}) & \mathbf{X}^{(k+1)} \\ (\mathbf{X}^{(k+1)})^H & \mathbf{W}^{(k+1)} \end{bmatrix} \right) \quad (22)$$

where the multiplier update rate η_{k+1} is the learnable parameter, and the output $\Lambda^{(k+1)}$ of $\mathbf{E}^{(k+1)}$ can be used as the input in the $(k+2)$ -th layer. It is essential to emphasize that the new parameters η_{k+1} are added to enhance the learning capability further and the performance of ANM-ADMM-Net compared to updating multipliers by ρ of ANM-ADMM (as shown in Algorithm 1).

Considering that each sub-layer's parameters are learned and tuned, there will be $3K$ parameters for K layer ANM-ADMM-Net generally, i.e., $\{\rho_1, \dots, \rho_k\}$, $\{\tau_1, \dots, \tau_k\}$, $\{\eta_1, \dots, \eta_k\}$. Compared with the ANM-ADMM algorithm, where the parameters $\rho_1 = \dots = \rho_k$ and $\tau_1 = \dots = \tau_k$ are fixed, this parameter learning strategy of ANM-ADMM-Net has the advantages of superior flexibility and brilliant nonlinear fitting capability [36]. More importantly, the design of the network is guided by the model and is highly interpretable.

3.2.2. Data Construction

The proposed ANM-ADMM-Net is a sparse recovery approach jointly driven by model and data. The key to its effectiveness is constructing a reasonable dataset with generalization capability. By building an adequate and complete dataset, ANM-ADMM-Net is less prone to overfitting during the training process, performing better DOA estimation. Thus, this paper randomly generates the signal data $\mathbf{Y}$ obeying a particular distribution and forms the received data $\mathbf{X}$. Specifically:

- (1) Given the array element number M , the frequency range $(f_{\min}, f_{\max}]$, corresponding angle range $(\theta_{\min}, \theta_{\max}] = (\text{asind}(2f_{\min}), \text{asind}(2f_{\max}))$, number of snapshots L .
- (2) Given the maximum value of the number of sources K_t and randomly generate the sources number k_t .
- (3) For each $k_t (1, 2, \dots, K_t)$, its frequency $f_{1:k_t}$ has a uniform distribution $U(f_{\min}, f_{\max})$, and the frequency interval between any two sources in the case of multiple sources needs to satisfy $\min_{i \neq j} |f_i - f_j| > 1/M$ [38], then the angle of the source can be obtained as $\theta_{1:k_t}$. The received signal $\mathbf{Y}$ is generated according to (1), where $\mathbf{X} = \mathbf{A}\mathbf{S}$, $\mathbf{A} = [\mathbf{a}(f_1), \mathbf{a}(f_2), \dots, \mathbf{a}(f_{K_t})]$, the complex amplitude matrix $\mathbf{S} = [\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_{K_t}] \in \mathbb{C}^{K_t \times L}$ obeys the standard normal distribution and $\mathbf{N}$ is Gaussian noise with SNR.
- (4) Repeat the above to obtain the set of D received signals $\{\mathbf{Y}_d, \mathbf{X}_d\}_{d=1}^D$, the set of the frequencies and the angles $\{\{f_{1:k_t}, \theta_{1:k_t}\}_d\}_{d=1}^D$. The ideal label set vector $\{\mathbf{u}_d\}_{d=1}^D$ can be obtained according to Vandermonde decomposition after the dual atomic norm minimization method [34,37].
- (5) Divide randomly the above set into a training dataset $\{\mathbf{Y}_q^{\text{train}}, \mathbf{u}_q^{\text{train}}\}_{q=1}^{Q=0.8D}$ and a testing dataset $\{\mathbf{Y}_o^{\text{test}}, \mathbf{u}_o^{\text{test}}\}_{o=1}^{O=0.2D}$.

3.2.3. Network Initialization and Training

Consider the difficulty of mapping $T(\cdot)$ in (5), $T^*(\cdot)$ and eigen-decomposition in Algorithm 1 when training ANM-ADMM-Net. Proper initialization of the parameters $\mathbf{\Omega} = \{\rho_{k+1}, \lambda_{k+1}, \gamma_{k+1}\}_{k=0}^{K-1}$ and an appropriate training method, including loss function, optimizer, and learning schedules, will make it easier to reach convergence and avoid falling into a locally optimal solution to a certain extent.

(1) Network initialization

The initial values of the parameters in each layer are set as $\rho_{1:K} = \rho_0$, $\tau_{1:K} = \tau_0$, and $\eta_{1:K} = \eta_0$ to enhance the proposed method's flexibility based on the theoretical analysis [41]. Compared with the ANM-ADMM algorithm with fixed parameter settings, ANM-ADMM-Net will substantially increase the convergence rate (i.e., reduce the number of iterations) and shorten the time to solve (5), with guaranteed convergence performance.

(2) Network training

The Adam algorithm is adopted for learning and tuning the parameters with an initial learning rate of 1×10^{-3} to achieve the possible global optimum rapidly. Based on the training dataset $\{\mathbf{Y}_q^{\text{train}}, \mathbf{u}_q^{\text{train}}\}_{q=1}^{Q=0.8D}$ constructed in Section 3.2.2 and given the network layer K , the optimal parameters $\mathbf{\Omega}^* = \{\rho_{k+1}^*, \tau_{k+1}^*, \eta_{k+1}^*\}_{k=0}^{K-1}$ can be obtained by minimizing the following normalized mean square error (NMSE) loss function using the principle of Back Propagation (BP) [42], i.e.,

$$\mathbf{\Omega}^* = \arg\min \left(\sum_{q=1}^Q L_u^q \right) / Q \quad (23)$$

where $L_u^q = ||\mathbf{u}^{(K)}(\boldsymbol{\Omega}, \boldsymbol{\Theta}^{(0)}, \boldsymbol{\Lambda}^{(0)}, \mathbf{Y}_q^{\text{train}})\sqrt{M}/2 - \mathbf{u}_q^{\text{train}}||_{\text{F}}^2 / ||\mathbf{u}_q^{\text{train}}||_{\text{F}}^2$, $\mathbf{u}^{(K)}(\boldsymbol{\Omega}, \boldsymbol{\Theta}^{(0)}, \boldsymbol{\Lambda}^{(0)}, \mathbf{Y}_q^{\text{train}})\sqrt{M}/2$ denotes the estimated components of the K -th Toeplitz transform sub-layer output of the network with parameters $\boldsymbol{\Omega}$, $\boldsymbol{\Theta}^{(0)} = \mathbf{0}_{M+L}$, $\boldsymbol{\Lambda}^{(0)} = \mathbf{0}_{M+L}$ and $\mathbf{Y}_q^{\text{train}}$ as inputs, respectively.

Then, according to the given testing dataset $\{\mathbf{Y}_o^{\text{test}}, \mathbf{u}_o^{\text{test}}\}_{o=1}^{O=0.2D}$, after obtaining the optimal parameters $\boldsymbol{\Omega}^*$, the recovered signal $\mathbf{X}_o^{\text{test}}$ and the Toeplitz matrix $T(\mathbf{u}_o^{\text{test}})$ can be estimated online by

$$\begin{cases} \mathbf{X}_o^{\text{test}} = \mathbf{X}^{(K)} \left\{ \boldsymbol{\Omega}^*, \boldsymbol{\Theta}^{(0)}, \boldsymbol{\Lambda}^{(0)}, \mathbf{Y}_o^{\text{test}} \right\} \\ \mathbf{u}_o^{\text{test}} = \mathbf{u}^{(K)} \left\{ \boldsymbol{\Omega}^*, \boldsymbol{\Theta}^{(0)}, \boldsymbol{\Lambda}^{(0)}, \mathbf{Y}_o^{\text{test}} \right\} \end{cases} \quad (24)$$

At the end of the ANM-ADMM-Net DOA method, $T(\mathbf{u}_o^{\text{test}}\sqrt{M}/2)$ can be the input of the Vandermonde decomposition to obtain the frequency f_{k_t} and amplitude values s_{k_t} , thereby achieving the estimated DOA $\theta_{k_t} = \text{asind}(2f_{k_t})$.

3.3. 2D DANM-ADMM DOA

To facilitate the solution, (13) is rewritten as (25):

$$\begin{aligned} [T(\mathbf{u}_x), T(\mathbf{u}_y)] = & \arg \min_{\mathbf{u}_x, \mathbf{u}_y, \mathbf{X}} \frac{\lambda}{2\sqrt{NM}} \text{Tr}(T(\mathbf{u}_x)) + \text{Tr}(T(\mathbf{u}_y)) \\ & + \frac{1}{2} ||\mathbf{Y} - \mathbf{X}||_2^2 \text{ s.t. } \boldsymbol{\Theta} = \begin{bmatrix} T(\mathbf{u}_x) & \mathbf{X}^H \\ \mathbf{X} & T(\mathbf{u}_y) \end{bmatrix} \geq 0 \end{aligned} \quad (25)$$

where $\lambda \in \mathbb{R}$ is the regularization factor.

Then, the augmented Lagrangian function of the problem (25) can be defined via [41]

$$\begin{aligned} & \underset{\mathbf{X}, \mathbf{u}_x, \mathbf{u}_y, \boldsymbol{\Theta}, \boldsymbol{\Lambda}}{\text{argmin}} \quad \Pi(\mathbf{X}, \mathbf{u}_x, \mathbf{u}_y, \boldsymbol{\Theta}, \boldsymbol{\Lambda}) \\ & = \underset{\mathbf{X}, \mathbf{u}_x, \mathbf{u}_y, \boldsymbol{\Theta}, \boldsymbol{\Lambda}}{\text{argmin}} \quad \frac{\lambda}{2\sqrt{NM}} \text{Tr}(T(\mathbf{u}_x)) + \text{Tr}(T(\mathbf{u}_y)) + \frac{1}{2} ||\mathbf{Y} - \mathbf{X}||_{\text{F}}^2 \\ & + \left\langle \boldsymbol{\Lambda}, \boldsymbol{\Theta} - \begin{bmatrix} T(\mathbf{u}_x) & \mathbf{X}^H \\ \mathbf{X} & T(\mathbf{u}_y) \end{bmatrix} \right\rangle + \frac{\rho}{2} ||\boldsymbol{\Theta} - \begin{bmatrix} T(\mathbf{u}_x) & \mathbf{X}^H \\ \mathbf{X} & T(\mathbf{u}_y) \end{bmatrix}||_{\text{F}}^2 \\ & = \underset{\mathbf{X}, \mathbf{u}_x, \mathbf{u}_y, \boldsymbol{\Theta}, \boldsymbol{\Lambda}}{\text{argmin}} \quad \frac{\lambda}{2\sqrt{NM}} \text{Tr}(T(\mathbf{u}_x)) + \text{Tr}(T(\mathbf{u}_y)) + \frac{1}{2} ||\mathbf{Y} - \mathbf{X}||_{\text{F}}^2 + \Delta \end{aligned} \quad (26)$$

where $\boldsymbol{\Lambda} \in \mathbb{C}^{(N+M) \times (N+M)}$ is the Lagrangian multiplier, and $\rho > 0$ is the penalty factor. Note (26) is an unconstrained optimization problem, so that, given the received signal $\mathbf{Y}$, unknown signal components $\mathbf{X}$ and Toeplitz matrix $T(\mathbf{u}_x)$ and $T(\mathbf{u}_y)$ can be estimated by minimizing the objective function of $\mathbf{X}, \mathbf{u}_x, \mathbf{u}_y, \boldsymbol{\Theta}, \boldsymbol{\Lambda}$ alternatively.

By the same token as in Algorithm 1, the ADMM algorithm for solving (13) is provided in Algorithm 2.

Algorithm 2: DANM-ADMM DOA algorithm.**Input:** $\mathbf{Y}$, number of iterations K , penalty factor ρ , regularization factor λ .**Initialization:** $\Theta^{(0)} = 0_{M+N}$ and $\Lambda^{(0)} = 0_{M+N}$.**For** $k = 0 : K - 1$ **do**

$$(1) \mathbf{X}^{(k+1)} = 1/(1 + \rho)(\mathbf{Y} + \Lambda_X^{(k)} + \rho\Theta_X^{(k)});$$

$$(2) \begin{cases} (T(\mathbf{u}_x))^+ = \rho^{-1}\Lambda_{T(\mathbf{u}_x)}^{(k)} + \Theta_{T(\mathbf{u}_x)}^{(k)} - \lambda/(2\sqrt{NM})\mathbf{N}\mathbf{I}_N \\ \mathbf{u}_x^{(k+1)} = \Gamma(T^*(\Lambda_{T(\mathbf{u}_x)}^{(k)} + \rho\Theta_{T(\mathbf{u}_x)}^{(k)}) - \lambda/(2\sqrt{NM})\mathbf{N}e_1)/\rho \end{cases};$$

$$(3) \begin{cases} (T(\mathbf{u}_y))^+ = \rho^{-1}\Lambda_{T(\mathbf{u}_y)}^{(k)} + \Theta_{T(\mathbf{u}_y)}^{(k)} - \lambda/(2\sqrt{NM})\mathbf{M}\mathbf{I}_M \\ \mathbf{u}_y^{(k+1)} = \Gamma(T^*(\Lambda_{T(\mathbf{u}_y)}^{(k)} + \rho\Theta_{T(\mathbf{u}_y)}^{(k)}) - \lambda/(2\sqrt{NM})\mathbf{M}e_1)/\rho \end{cases};$$

$$(4) \begin{cases} \hat{\Theta}^{(k+1)} = \begin{bmatrix} T(\mathbf{u}_x^{(k+1)}) & (\mathbf{X}^{(k+1)})^H \\ \mathbf{X}^{(k+1)} & T(\mathbf{u}_y^{(k+1)}) \end{bmatrix} - \rho^{-1}\Lambda^{(k)} \\ \hat{\Theta}^{(k+1)} = P\text{diag}(\{\delta_g\})P^{-1}, \Theta^{(k+1)} = P\text{diag}(\{\delta_g\}_+)P^{-1} \end{cases};$$

$$(5) \Lambda^{(k+1)} = \Lambda^{(k)} + \rho \left(\Theta^{(k+1)} - \begin{bmatrix} T(\mathbf{u}_x^{(k+1)}) & (\mathbf{X}^{(k+1)})^H \\ \mathbf{X}^{(k+1)} & T(\mathbf{u}_y^{(k+1)}) \end{bmatrix} \right);$$

End**Output:** Recovered signal of interest $\mathbf{X}^{(K)}$, the optimal estimate $\mathbf{u}_x^{(K)}$ and $\mathbf{u}_y^{(K)}$.**Then:** Compute matrix $T(\mathbf{u}_x^{(K)})$ and $T(\mathbf{u}_y^{(K)})$; Retrieve two dimensional frequencies $(\hat{w}_x, \hat{w}_y)$ without pairing by root music algorithm or matrix pencil method [38]. Then, final 2D DOA estimation can be obtained by following pairing technique.
$$j_i = \underset{j}{\operatorname{argmax}} |a_x^T(\hat{w}_{x,i})\mathbf{X}^{(K)}a_y^*(\hat{w}_{y,j})|, \text{ where } j_i \text{ denotes the frequency index of } \hat{w}_y \text{ that matched to } \hat{w}_{x,i}, \text{ i.e., } \hat{w}_y^*.$$
Finally: Obtain the paired $(\hat{w}_x, \hat{w}_y^*)$ and compute the corresponding angles $(\hat{\alpha}_x, \hat{\beta}_y^*)$, final pitch angle ϕ_{k_t} and azimuth angle θ_{k_t} can be calculated by (6) and (7).

where Θ are Λ Hermitian matrices, and $\Theta_X, \Lambda_X \in \mathbb{C}^{N \times M}$, $\Theta_{T(\mathbf{u}_x)} \in \mathbb{C}^{N \times N}$, $\Lambda_{T(\mathbf{u}_y)} \in \mathbb{C}^{M \times M}$, $\Theta = \begin{bmatrix} \Theta_{T(\mathbf{u}_x)} & \Theta_X^H \\ \Theta_X & \Theta_{T(\mathbf{u}_y)} \end{bmatrix}$, $\Lambda = \begin{bmatrix} \Lambda_{T(\mathbf{u}_x)} & \Lambda_X^H \\ \Lambda_X & \Lambda_{T(\mathbf{u}_y)} \end{bmatrix} \in \mathbb{C}^{(N+M) \times (N+M)}$; $g = 1, 2, \dots, M + N$, $\{\delta_g\}_+$ denotes that all elements smaller than zero are set to zero; and P is an orthogonal matrix satisfying $PP^T = \mathbf{I}_{M+N}$.

According to the above derivation, the parameters of the model-driven ADMM algorithm, including the penalty factor ρ and the regularization factor λ , need to be set in advance, which is a challenge for practical applications. Based on the idea of the DU method, this paper also expands the algorithm into a deep neural network DANM-ADMM-Net as Section 3.2, thus solving the above problems. The optimal estimates $T(\mathbf{u}_x^{(K)})$ and $T(\mathbf{u}_y^{(K)})$ can be considered as labels and construct a proper loss function in the next subsection.

3.4. 2D DANM-ADMM-Net DOA

In the following, the four parts of DANM-ADMM-Net: network structures, dataset construction, network initialization, and training, are described thoroughly.

3.4.1. Network Structure

According to the steps in Algorithm 2, the ADMM algorithm can be mapped to a K layer network DANM-ADMM-Net, shown in Figure 3, whose inputs are $\mathbf{Y}$, $\Theta^{(0)} = 0_{M+N}$ and $\Lambda^{(0)} = 0_{M+N}$, and the learnable parameters are $\Omega = \{\Omega^{(k+1)}\}_{k=0}^{K-1} = \{\rho_{k+1}, \lambda_{k+1}, \gamma_{k+1}\}_{k=0}^{K-1}$, to arrive at the signal components $\mathbf{X}^{(K)}$ and two Toeplitz matrixes $T(\mathbf{u}_x^{(K)})$, $T(\mathbf{u}_y^{(K)})$. The $(k+1)$ -th layer operation of DANM-ADMM-Net can be expressed as

$$[\mathbf{X}^{(k+1)}, \mathbf{u}_x^{(k+1)}, \mathbf{u}_y^{(k+1)}] = F_{k+1}\{\mathbf{Y}, \Theta^{(k)}, \Lambda^{(k)}, \Omega^{(k+1)}\} \quad (27)$$

where $F_{k+1}\{\cdot\}$ contains five main structure sub-layers, including reconstruction sub-layer $\mathbf{A}^{(k+1)}$, the Toeplitz transform sub-layer $\mathbf{B}^{(k+1)}$, the Toeplitz transform sub-layer $\mathbf{C}^{(k+1)}$, the nonlinear sub-layer $\mathbf{D}^{(k+1)}$, and the multiplier update sub-layer $\mathbf{E}^{(k+1)}$. The specific description is as follows.

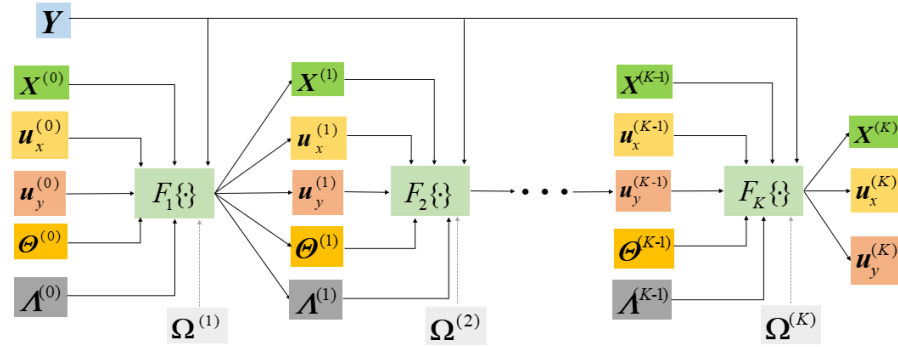


Figure 3. The network structure of DANM-ADMM-Net.

- (1) Reconstruction sub-layer $\mathbf{A}^{(k+1)}$: Taking the output $\Lambda_X^{(k)}$ of the sub-layer $\mathbf{E}^{(k)}$ and $\Theta_x^{(k)}$ of the sub-layer $\mathbf{D}^{(k)}$ of k -th layer and received signal $\mathbf{Y}$ as the inputs, the output $\mathbf{X}^{(k+1)}$ is updated as

$$\mathbf{X}^{(k+1)} = 1/(1 + \rho_{k+1})(\mathbf{Y} + \Lambda_X^{(k)} + \rho_{k+1}\Theta_x^{(k)}) \quad (28)$$

where ρ_{k+1} is the learnable parameter. The output $\mathbf{X}^{(k+1)}$ of $\mathbf{A}^{(k+1)}$ can be used as the input of the sub-layer $\mathbf{D}^{(k+1)}$ and $\mathbf{E}^{(k+1)}$ in the $(k + 1)$ -th layer.

- (2) Toeplitz transform sub-layer $\mathbf{B}^{(k+1)}$: Taking the output $\Theta_{T(u_x)}^{(k)}$ of the sub-layer $\mathbf{D}^{(k+1)}$ and $\Lambda_{T(u_x)}^{(k)}$ of the sub-layer $\mathbf{E}^{(k)}$ in the k -th layer as inputs, the output of $\mathbf{B}^{(k+1)}$ is represented by

$$\mathbf{u}_x^{(k+1)} = \Gamma(T^*(\Lambda_{T(u_x)}^{(k)} + \rho_{k+1}\Theta_{T(u_x)}^{(k)}) - \lambda_{k+1}/(2\sqrt{NM})Ne_1)/\rho_{k+1} \quad (29)$$

where λ_{k+1} is the learnable parameter. The output $\mathbf{u}_x^{(k+1)}$ is the input of the sub-layer $\mathbf{D}^{(k+1)}$ and $\mathbf{E}^{(k+1)}$ in the $(k + 1)$ -th layer.

- (3) Toeplitz transform sub-layer $\mathbf{C}^{(k+1)}$: Taking the output $\Theta_{T(u_y)}^{(k)}$ of the sub-layer $\mathbf{D}^{(k+1)}$ and $\Lambda_{T(u_y)}^{(k)}$ of the sub-layer $\mathbf{E}^{(k)}$ in the k -th layer as inputs, the output of $\mathbf{C}^{(k+1)}$ can be expressed as

$$\mathbf{u}_y^{(k+1)} = \Gamma(T^*(\Lambda_{T(u_y)}^{(k)} + \rho_{k+1}\Theta_{T(u_y)}^{(k)}) - \lambda_{k+1}/(2\sqrt{NM})Me_1)/\rho_{k+1} \quad (30)$$

where the output $\mathbf{u}_y^{(k+1)}$ can be used as the input of the sub-layer $\mathbf{D}^{(k+1)}$ and $\mathbf{E}^{(k+1)}$ in the $(k + 1)$ -th layer.

- (4) Nonlinear sub-layer $\mathbf{D}^{(k+1)}$: Taking the output $\Lambda_{T(u_x)}^{(k)}$ of the sub-layer $\mathbf{E}^{(k)}$ of the k -th layer, the output $\mathbf{X}^{(k+1)}$ of the sub-layer $\mathbf{A}^{(k)}$, the output $\mathbf{u}_x^{(k+1)}$ of the sub-layer $\mathbf{B}^{(k+1)}$ and the output $\mathbf{u}_y^{(k+1)}$ of the sub-layer $\mathbf{C}^{(k+1)}$ as inputs, the output of $\mathbf{D}^{(k+1)}$ can be given as

$$\begin{cases} \hat{\Theta}^{(k+1)} = \begin{bmatrix} T(\mathbf{u}_x^{(k+1)}) & (\mathbf{X}^{(k+1)})^H \\ \mathbf{X}^{(k+1)} & T(\mathbf{u}_y^{(k+1)}) \end{bmatrix} - \rho_{k+1}^{-1}\Lambda^{(k)} \\ \hat{\Theta}^{(k+1)} = P\text{diag}(\{\delta_g\})P^{-1}, \Theta^{(k+1)} = P\text{diag}(\{\delta_g\}_+)P^{-1} \end{cases} \quad (31)$$

where the output $\Theta^{(k+1)}$ of $\mathbf{D}^{(k+1)}$ can be used as the input of the sub-layer $\mathbf{A}^{(k+2)}$, $\mathbf{B}^{(k+2)}$, and $\mathbf{C}^{(k+2)}$ in the $(K+2)$ -th layer.

- (5) Multiplier update sub-layer $\mathbf{E}^{(k+1)}$: Taking the output $\Lambda^{(k)}$ of the sub-layer $\mathbf{E}^{(k)}$ of the k -th layer, the output $\mathbf{X}^{(k+1)}$, $\mathbf{u}_x^{(k+1)}$, $\mathbf{u}_y^{(k+1)}$, $\Theta^{(k+1)}$ of the sub-layer $\mathbf{A}^{(k+1)}$, $\mathbf{B}^{(k+1)}$, $\mathbf{C}^{(k+1)}$, $\mathbf{D}^{(k+1)}$ respectively, as inputs, the output of $\mathbf{E}^{(k+1)}$ can be updated by

$$\Lambda^{(k+1)} = \Lambda^{(k)} + \gamma_{k+1} \left(\Theta^{(k+1)} - \begin{bmatrix} T(\mathbf{u}_x^{(k+1)}) & (\mathbf{X}^{(k+1)})^H \\ \mathbf{X}^{(k+1)} & T(\mathbf{u}_y^{(k+1)}) \end{bmatrix} \right) \quad (32)$$

where multiplier update rate γ_{k+1} is the learnable parameter, the output $\Lambda^{(k+1)}$ of $\mathbf{E}^{(k+1)}$ can be used as the input in the $(k+2)$ -th layer. The new parameters γ_{k+1} are also added to enhance the learning capability further and the performance of DANM-ADMM-Net compared to updating multipliers by ρ of DANM-ADMM (as shown in Algorithm 2).

Considering that each sub-layer's parameters are learned and tuned, there will be $3K$ parameters for K layer DANM-ADMM-Net generally, i.e., $\{\rho_1, \dots, \rho_k\}$, $\{\lambda_1, \dots, \lambda_k\}$, $\{\gamma_1, \dots, \gamma_k\}$. The parameters $\rho_1 = \dots = \rho_k$ and $\lambda_1 = \dots = \lambda_k$ of the DANM-ADMM algorithm are fixed.

3.4.2. Data Construction

This subsection randomly generates the received data $\mathbf{Y}^{k_t}$ and constructs the dataset. Specifically:

- (1) Given the array element number M , the pulse number N , pitch angle $\phi_k \in [\phi_{\min}, \phi_{\max}]$, azimuth angle $\theta_k \in [\theta_{\min}, \theta_{\max}]$, and number of samples D .
- (2) Given the maximum value of the number of sources K_t and randomly generate the sources number k_t .
- (3) For each $k_t (1, 2, \dots, K_t)$, compute $w_{x,k_t} = \pi \sin \phi_{k_t} \cos \theta_{k_t}$ and $w_{y,k_t} = \pi \sin \phi_{k_t} \sin \theta_{k_t}$ satisfying $\min_{i \neq j} |w_{x,i} - w_{x,j}| \geq 1/\lfloor 4(N-1) \rfloor$ and $\min_{i \neq j} |w_{y,i} - w_{y,j}| \geq 1/\lfloor 4(M-1) \rfloor$ [19].

Then received signal $\{\{\mathbf{Y}_p^{k_t}, \mathbf{X}_p^{k_t}\}_{p=1}^P\}_{k_t=1}^{K_t} = \{\mathbf{Y}^d, \mathbf{X}^d\}_{d=1}^{D=K_t \times P}$ is produced by $\mathbf{Y}_p^{k_t} = \mathbf{X}_p^{k_t} + \mathbf{N}_p$, where $\mathbf{X}_p^{k_t} = \sum_{k_t} \mathbf{s}_{k_t} \mathbf{a}_x(w_{x,k_t}) \mathbf{a}_y^T(w_{y,k_t}) = \mathbf{A}_x \mathbf{S} \mathbf{A}_y^T$, and amplitude s_{k_t} obeys complex standard normal distribution. $\mathbf{N}_p$ is the complex Gaussian white noise with SNR.

- (4) Randomly divide the received signal data $\{\mathbf{Y}^d, \mathbf{X}^d\}_{d=1}^D$ into Q training data $\{\mathbf{Y}_q^{\text{train}}, \mathbf{X}_q^{\text{train}}\}_{q=1}^{Q=0.8D}$ and $O(D-Q)$ testing data $\{\mathbf{Y}_o^{\text{test}}, \mathbf{X}_o^{\text{test}}\}_{o=1}^{O=0.2D}$.
- (5) Use the DANM-CVX method to address (13), thereby obtaining the training label set $\{\mathbf{X}_q^{\text{train}}, T(\mathbf{u}_{x,q}^{\text{train}}), T(\mathbf{u}_{y,q}^{\text{train}})\}_{q=1}^Q$, the testing label set $\{\mathbf{X}_o^{\text{test}}, T(\mathbf{u}_{x,o}^{\text{test}}), T(\mathbf{u}_{y,o}^{\text{test}})\}_{o=1}^O$ according to the pairing technique in Algorithm 2.

3.4.3. Network Initialization and Training

The initial values of the parameters in each layer are set as $\rho_{1:K} = \rho_0$, $\lambda_{1:K} = \lambda_0$, and $\gamma_{1:K} = \gamma_0$. The Adam algorithm is adopted for learning and tuning the parameters with an initial learning rate of 2×10^{-3} to achieve the possible global optimum rapidly. Based on the training dataset $\{\mathbf{X}_q^{\text{train}}, T(\mathbf{u}_{x,q}^{\text{train}}), T(\mathbf{u}_{y,q}^{\text{train}})\}_{q=1}^Q$ constructed in Section 3.4.2 and given the number of network layers K , the optimal parameters $\Omega^* = \{\rho_{k+1}^*, \lambda_{k+1}^*, \gamma_{k+1}^*\}_{k=0}^{K-1}$ can be obtained by minimizing the following NMSE loss function, i.e.,

$$\Omega^* = \operatorname{argmin} \frac{1}{Q} \sum_{q=1}^Q 0.5 L_{T(\mathbf{u}_x)}^q + 0.5 L_{T(\mathbf{u}_y)}^q \quad (33)$$

where $\begin{cases} L_{T(\mathbf{u}_x)}^q = \|T(\mathbf{u}_x)^{(K)}(\boldsymbol{\Omega}, \boldsymbol{\Theta}^{(0)}, \boldsymbol{\Lambda}^{(0)}, \mathbf{X}_q^{\text{train}}) - T(\mathbf{u}_x)_{\text{train}}^q\|_2^2 / \|T(\mathbf{u}_x)_{\text{train}}^q\|_F^2 \\ L_{T(\mathbf{u}_y)}^q = \|T(\mathbf{u}_y)^{(K)}(\boldsymbol{\Omega}, \boldsymbol{\Theta}^{(0)}, \boldsymbol{\Lambda}^{(0)}, \mathbf{X}_q^{\text{train}}) - T(\mathbf{u}_y)_{\text{train}}^q\|_2^2 / \|T(\mathbf{u}_y)_{\text{train}}^q\|_F^2 \end{cases}$, $T(\mathbf{u}_x)^{(K)}(\boldsymbol{\Omega}, \boldsymbol{\Theta}^{(0)}, \boldsymbol{\Lambda}^{(0)}, \mathbf{X}_q^{\text{train}})$, and $T(\mathbf{u}_y)^{(K)}(\boldsymbol{\Omega}, \boldsymbol{\Theta}^{(0)}, \boldsymbol{\Lambda}^{(0)}, \mathbf{X}_q^{\text{train}})$ denotes the estimated components of the K -th Toeplitz transform sub-layer output of the network with parameters $\boldsymbol{\Omega}$, $\boldsymbol{\Theta}^{(0)} = \mathbf{0}_{M+N}$, $\boldsymbol{\Lambda}^{(0)} = \mathbf{0}_{M+N}$ and $\mathbf{X}_q^{\text{train}}$ as inputs, respectively.

Then, according to the given testing dataset $\{\mathbf{X}_o^{\text{test}}, T(\mathbf{u}_{x,o}^{\text{test}}), T(\mathbf{u}_{y,o}^{\text{test}})\}_{o=1}^O$, after obtaining the optimal parameters $\boldsymbol{\Omega}^*$, the recovered signal $\mathbf{X}_o^{\text{test}}$ and the Toeplitz matrix $T(\mathbf{u}_{x,o}^{\text{test}})$, $T(\mathbf{u}_{y,o}^{\text{test}})$ can be estimated online by

$$\begin{cases} \mathbf{X}_o^{\text{test}} = \mathbf{X}^{(K)}\{\boldsymbol{\Omega}^*, \mathbf{X}_o^{\text{test}}, \boldsymbol{\Theta}^{(0)}, \boldsymbol{\Lambda}^{(0)}\} \\ T(\mathbf{u}_{x,o}^{\text{test}})^{(K)} = T(\mathbf{u}_x)^{(K)}\{\boldsymbol{\Omega}^*, \mathbf{X}_o^{\text{test}}, \boldsymbol{\Theta}^{(0)}, \boldsymbol{\Lambda}^{(0)}\} \\ T(\mathbf{u}_{y,o}^{\text{test}})^{(K)} = T(\mathbf{u}_y)^{(K)}\{\boldsymbol{\Omega}^*, \mathbf{X}_o^{\text{test}}, \boldsymbol{\Theta}^{(0)}, \boldsymbol{\Lambda}^{(0)}\} \end{cases} \quad (34)$$

Follow the procedures of the DANM-ADMM algorithm in Algorithm 2 and thereby achieve 2D DOA estimation perfectly at the end of the DANM-ADMM-Net method.

4. Experiment Results

In this section, we evaluate the DU-DOA method based on the ANM-ADMM-Net and DANM-ADMM-Net through simulation experiments. Considering that the ADMM algorithm is the only iterative algorithm for solving the ANM and DANM models, we compare it to traditional 1D and 2D DOA estimation methods with fixed parameters. For the convenience of training and testing, all offline training procedures are implemented based on Python 3.8 with the configuration of Intel(R) Core i7-6246 3.30GHz CPU and NVIDIA Quadro GV100 GPU. Once the optimal parameters after training obtaining, all testing simulations will be implemented based on MATLAB 2020b online.

Since the noise level is generally unknown in the practical application, only the training data without noise are used in the training process of the ADMM network in this paper. Then noise is added to the data in the test data to verify the ADMM algorithm's performance under different SNRs. Therefore, the training and testing datasets for DOA estimation can be constructed with parameters in Tables 1 and 2 according to Sections 3.2.2 and 3.4.2, respectively.

Table 1. Simulation parameters of 1D DOA estimation.

Parameter	Notation	Value
Antenna elements number	M	10
Frequency range	$[f_{\min}, f_{\max}]$	(0,0.5]
Maximum signal sources	K_t	3
Number of snapshots	L	1/3/5
Number of samples	D	600
Signal-to-noise Ratio	SNR	Inf
Number of training samples	Q	480
Number of validation samples	O	120

Table 2. Simulation parameters of 2D DOA estimation.

Parameter	Notation	Value
Antenna elements number	$N \times M$	$\times 20$
Pitch angle range	ϕ_{k_t}	$[0^\circ, 90^\circ]$
Azimuth angle range	θ_{k_t}	$[0^\circ, 90^\circ]$
Maximum signal sources	K_t	3
Number of samples in each source	P	200
Number of samples	D	600
Signal-to-noise Ratio	SNR	Inf
Number of training samples	Q	480
Number of validation samples	O	120

4.1. Network Convergence Analysis

This subsection investigates the convergence performance of the ANM-ADMM-Net and DANM-ADMM-Net under different layers and compares it with the traditional algorithms with fixed iterative parameters.

The iteration parameters of ANM-ADMM are set as $\rho_0 = 0.5$, $\tau_0 = 0.01$, $\eta_0 = 0.5$. Set different network layers $K = 10 \sim 40$, initialize and train network (450 epochs), and the results are shown in Figure 4. Among them, Figure 4a,b shows training NMSEs and testing NMSEs when $K = 40$, and Figure 4c shows the NMSEs of two methods. Based on Figure 4a,b, the training and testing NMSEs of the proposed method decrease with the increase in training time and effectively reach convergence. Figure 4c shows that, as the number of network layers (the number of iterations) increases, the NMSEs of both ANM-ADMM-Net and ANM-ADMM algorithms gradually decrease, and the former is much smaller than the latter. In addition, only when the number of iterations of the ANM-ADMM algorithm is at least fifty to sixty times higher than the ANM-ADMM-Net, their NMSE is equal, which implies the computing complexity required for convergence is reduced.

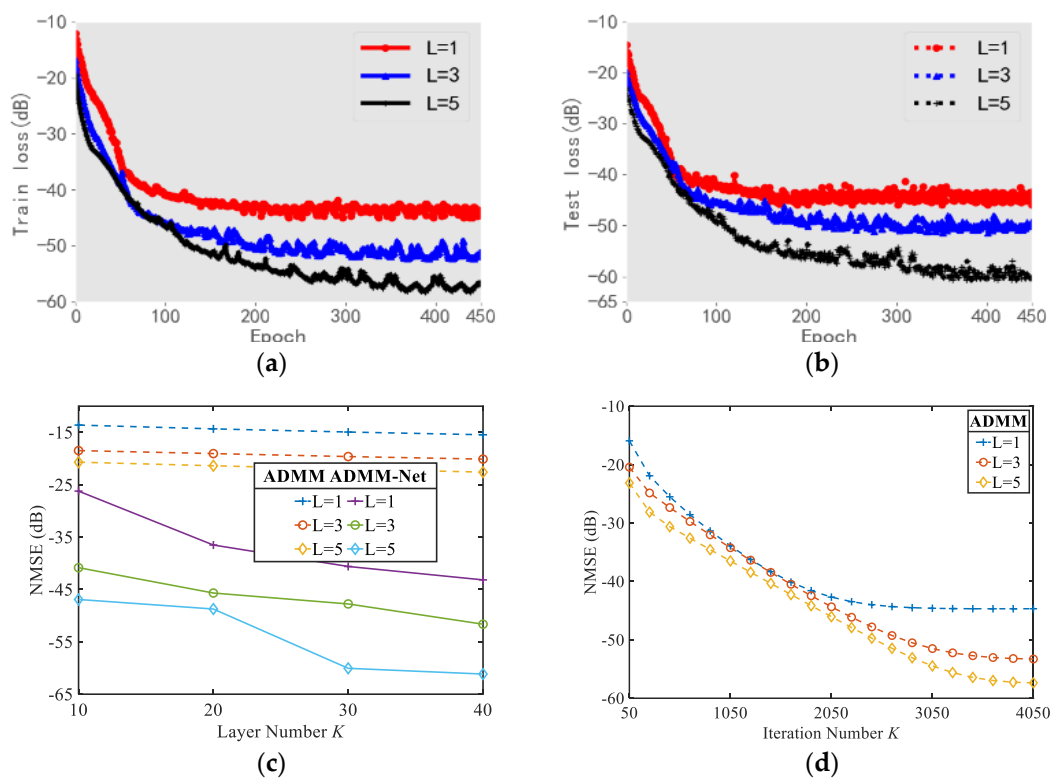


Figure 4. Convergence performance of ANM-ADMM-Net and its comparison with ANM-ADMM algorithm. (a) Train NMSEs when $M = 10$ (b) Test NMSEs when $M = 10$ (c) Test NMSEs when $K = 10 - 40$ (d) Test NMSEs when $K = 50 - 4050$.

The iteration parameters of DANM-ADMM algorithm are set as $\rho_0 = 0.1$, $\lambda_0 = 0.1$, and $\gamma_0 = 0.1$. Set different network layers $K = 5 \sim 25$, initialize and train DANM-ADMM-Net (600 epochs), and the results are shown in Figure 5. Among them, Figure 5a,b is the training NMSEs and testing NMSEs of DANM-ADMM-Net when $K = 5/10$, and Figure 5c shows the NMSEs of both when the number of network layers (iterations) $K = 5 \sim 25$. Figure 5d reveals the NMSEs of algorithm when iterations $K = 5 \sim 300$. Based on Figure 5a,b, the training and testing NMSEs of DANM-ADMM-Net decrease with the increase in training time and effectively reach convergence after 500 training epochs. In addition, Figure 5c shows that as the number of network layers (the number of iterations) increases, the NMSEs of both the DANM-ADMM-Net and DANM-ADMM algorithm decrease. However, the NMSE of the former is much smaller than that of the latter. From Figure 5c,d, the two

have similar NMSE only when the number of iterations of the algorithm is fifty times higher than that of network when $K = 10$. Therefore, it can be concluded that the proposed can learn the optimal iteration parameters from the constructed dataset and obtain better convergence performance.

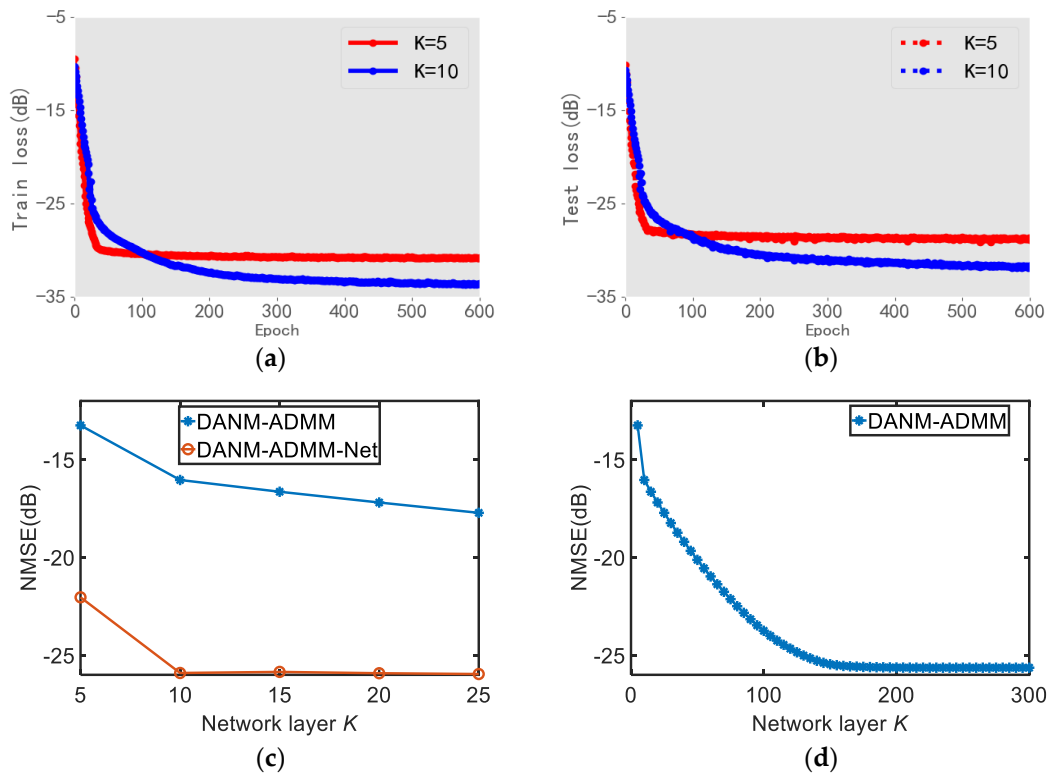


Figure 5. Convergence performance of DANM-ADMM-Net and its comparison with DANM-ADMM algorithm. (a) Train NMSEs when $K = 5/10$. (b) Test NMSEs when $K = 5/10$. (c) Test NMSEs when $K = 5 \sim 25$. (d) Test NMSEs when $K = 5 \sim 300$.

In practical applications, the networks can be trained offline based on different simulation conditions to determine the range of network layers that can obtain better DOA estimation performance and lower computational complexity. Then, the network layers can be selected according to the actual situation.

4.2. DOA Estimation Results Analysis

4.2.1. 1D DOA Estimation Results Analysis

This subsection investigates the performance of ANM-ADMM-Net for DOA estimation and its upsides over the SBL algorithm, dual ANM, ANM-CVX and ANM-ADMM methods with simulation experiments. The root mean square error (RMSE) is evaluated by $M_c = 100$ Monte Carlo trials.

$$\text{RMSE} = \frac{1}{O} \sum_{o=1}^O \sqrt{\frac{1}{K_t M_c} \sum_{j=1}^{M_c} \sum_{k_t=1}^{K_t} (\hat{\theta}_{o,k_t} - \theta_{o,k_t})^2} \quad (35)$$

The results of different methods are shown in Figure 6, where grid numbers $N = 100$, $K = 200$ and $\lambda = 1e-3$ for the SBL algorithm [8]; $K = 30$, $L = 5$ for ANM-ADMM and ANM-ADMM-Net. The results indicate that unsuitable grid division inevitably causes the estimated DOA offset of the SBL algorithm. dual ANM and ANM-CVX methods are always the optimal estimation results. ANM-ADMM cannot recover the amplitude and degree of the signal perfectly due to limited iterations. However, the proposed ANM-ADMM-Net has a better estimation of amplitude and degree than the ANM-ADMM algorithm with

fixed parameters, demonstrating the effectiveness of the combination of model-driven and data-driven DU methods.

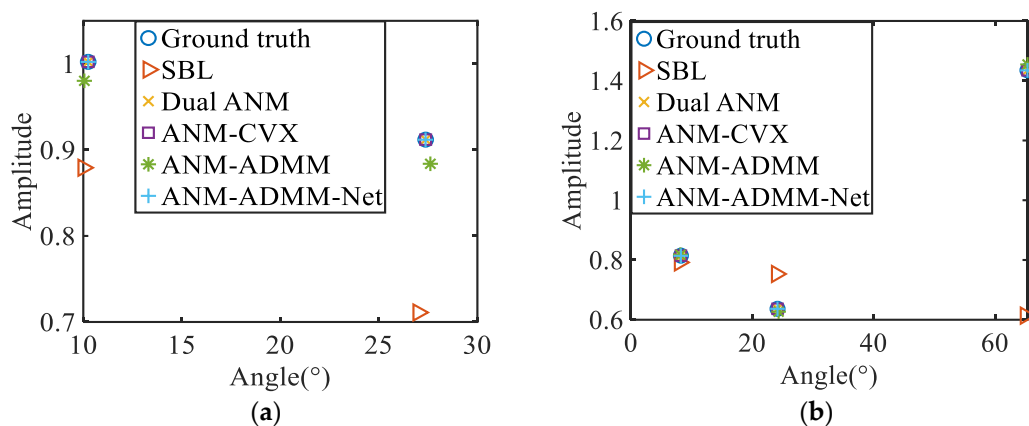


Figure 6. DOA estimation results of different methods. (a) Two targets. (b) Three targets.

Figure 7 shows the test RMSEs when the number of network layers $K = 10 \sim 40$. The RMSEs of ANM-ADMM and ANM-ADMM-Net gradually decrease as the number of network layers increases (number of iterations), and the latter is much smaller than the former with fixed parameters. The above results demonstrate that the more layers of the network, the more optimal iterative parameters can be learned from the constructed dataset due to its powerful nonlinear fitting capability and superior flexibility, resulting in better DOA estimation performance. In other words, the DU-gridless DOA method based on ANM-ADMM-Net is suitable for efficiently resolving the different sparsity problems (different target signal problems) at a lower computational cost.

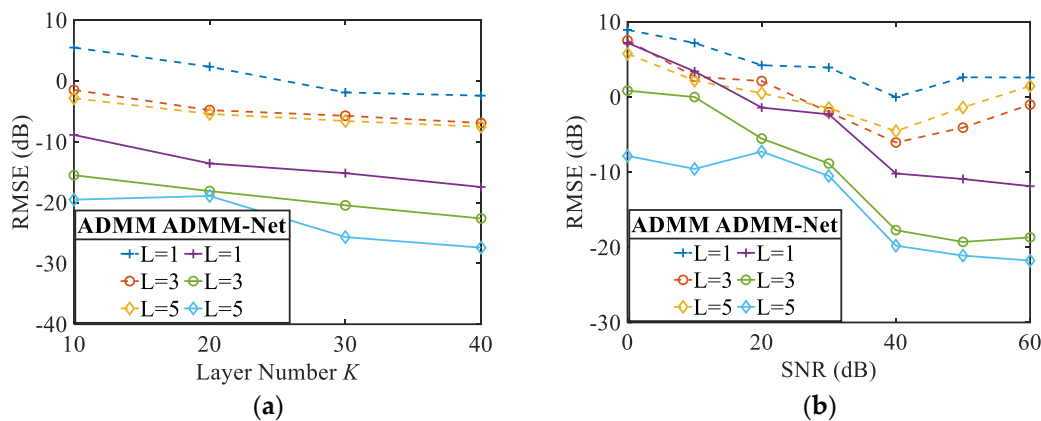


Figure 7. Test RMSEs versus K and SNR. (a) versus network layer K ; (b) versus SNR.

The testing dataset when $\text{SNR} = 0 \sim 60 \text{ dB}$ is constructed to verify the noise robustness according to Section 3.2.2 (i.e., generate testing data for every SNR and noise regularization factor of dual ANM is 0.1), and the test results when $K = 40$ are shown in Figure 7b. It can be seen that when $\text{SNR} < 20 \text{ dB}$, ANM-ADMM is unable to be implemented for estimation. However, the proposed method still performs better under limited network layers/iterations, demonstrating the higher noise robustness of the latter with the optimal parameters. In addition, when $\text{SNR} > 40 \text{ dB}$, the test RMSE of ANM-ADMM-Net tends to be stable and close to the results obtained in the noise-free case. The above implies that even if noise-free data train the network, the proposed method still obtains better parameter results and can perform DOA estimation for the actual array received data containing noise.

4.2.2. 2D DOA Estimation Results Analysis

This subsection investigates the performance of DANM-ADMM-Net for 2D DOA estimation and its upsides over the conventional 2D DOA methods with simulation experiments. The root mean square error (RMSE) is also evaluated by $M_c = 100$ Monte Carlo trials.

$$\text{RMSE} = \frac{1}{O} \sum_{o=1}^O \sqrt{\frac{1}{K_t M_c} \sum_{j=1}^{M_c} \sum_{k_t=1}^{K_t} (\hat{\phi}_{o,k_t} - \phi_{o,k_t})^2 + (\hat{\theta}_{o,k_t} - \theta_{o,k_t})^2} \quad (36)$$

The 2D DOA results estimated by different methods when $M = 20, K = 20$ are shown in Figure 8, where the parameters of the 2D-MUSIC method are set to SNR = 20 dB, and the number of snapshots $L = 20$. The results in Figure 8b show that the DANM-ADMM algorithm fails to precisely converge to the ground truth precisely at a finite number of iterations due to improper parameter settings, resulting in some deviations in both azimuth and pitch angles estimated for individual targets. In contrast, the parameters of the proposed method are optimized for data and networks, thus converging to a better result completely at a finite number of layers.

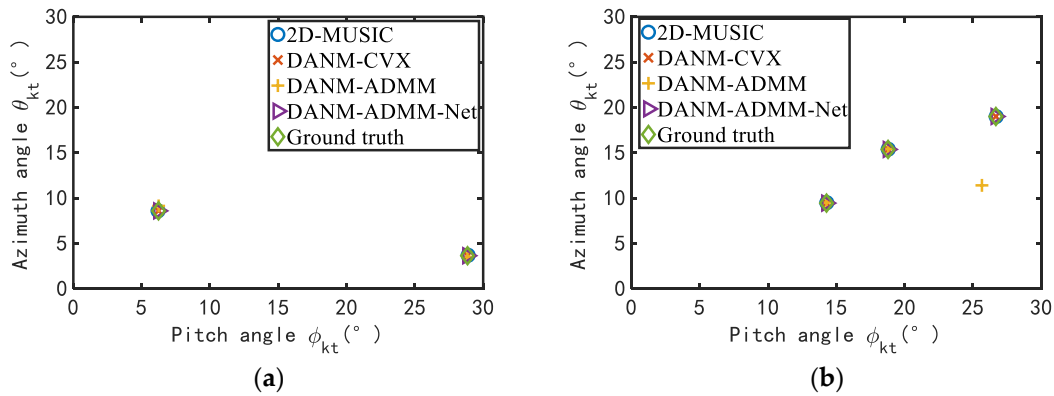


Figure 8. 2D DOA estimation results of different methods. (a) Two targets. (b) Three targets.

Figure 9a gives the results of the test RMSEs when the number of array elements $M = 20$ and the number of network layers $K = 5 \sim 25$. The RMSEs of DANM-ADMM and DANM-ADMM-Net gradually decrease as the number of network layers increases (number of iterations). When $K = 25$, the RMSE of the latter is reduced by 20dB compared to the former with fixed parameters, which indicates that the more layers of the network, the more optimal iterative parameters can be learned from the constructed dataset, resulting in better 2D DOA estimation performance.

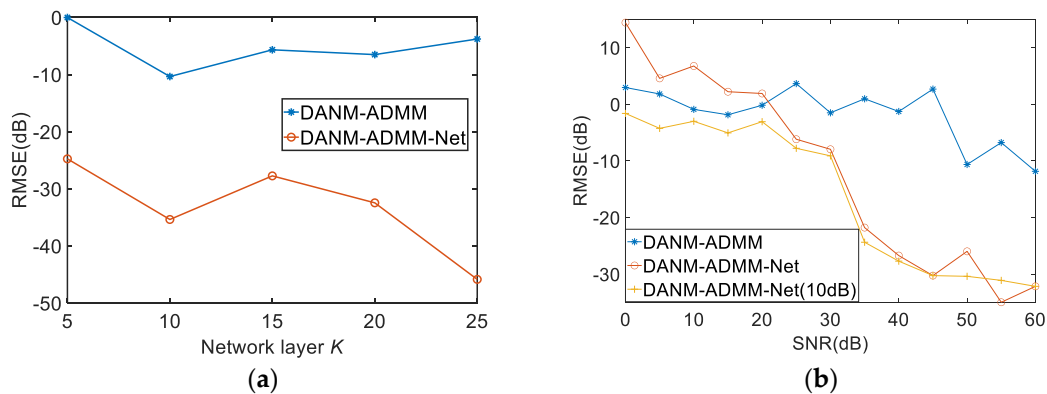


Figure 9. Test RMSEs versus K and SNR; (a) versus network layer K . (b) versus SNR.

The test dataset is constructed by Section 3.4.3 when SNR = 0~60dB (i.e., test data are generated for each SNR) to verify the noise robustness of DANM-ADMM-Net. The results when $K = 25$ are shown in Figure 9b. The tested RMSE of both methods decreased with increased SNR under a limited number of network layers/iterations. If we train the network with data when noise exists, for example, SNR = 10 dB, DANM-ADMM-Net performs better than DANM-ADMM under any circumstances. Therefore, to ensure higher noise robustness, one can estimate the signal-to-noise ratio first, and then use the corresponding data containing the noise to train the network.

4.3. Computational Complexity and Running Time Comparison

This subsection analyzes the computational complexity and running time of SDPT3 based ANM-CVX method and ANM-ADMM-Net at different array elements when network layers $K = 40$. The running time results when $M = 10 \sim 50$, $L = 5$ and $M = 10$, $L = 1 \sim 15$ are shown in Figure 10. The results of the comparison of DANM-CVX and DANM-ADMM-Net when network layers $K = 25$ are also shown in Figure 11.

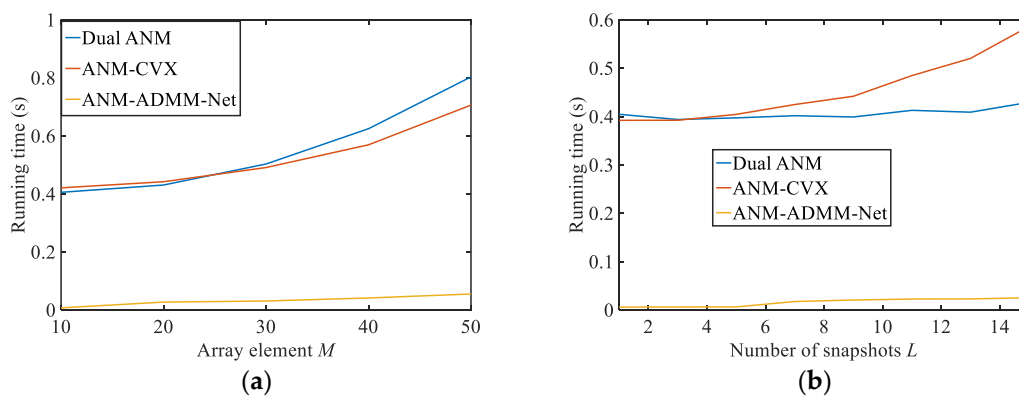


Figure 10. Running time versus array element number M and snapshot number L . (a) when $M = 10 \sim 50$; (b) when $L = 1 \sim 15$.

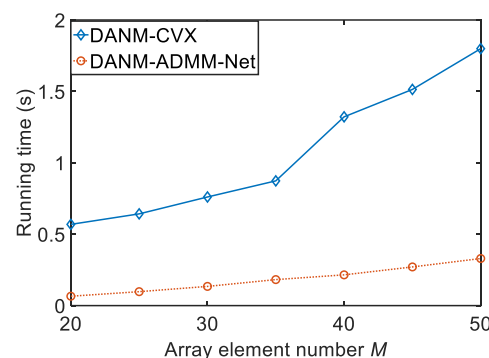


Figure 11. Running time of different methods versus array element number M .

From the paper [43], it is known that the computational complexity of the solver SDPT3-based CVX method is $q_1^2 q_2^2$, where q_1 denotes the number of variables and q_2 denotes the dimensionality of the SDP matrix. Therefore, the computational complexity of the CVX method for solving the ANM model is $O\{Q(M+L)^2(ML+M+L)^2\}$ and $O\{Q(M+N)^2(MN+M+N)^2\}$ for solving the DANM model. In contrast, while that of the ADMM method is $O\{Q(M+L)^3\}$ and $O\{Q(M+N)^3\}$, respectively, where Q denotes the number of iterations, which indicates that when the matrix size is large, the computational complexity of the ANM-ADMM and DANM-ADMM methods to solve the ANM model and DANM model, respectively, is much smaller than that of the ANM-CVX and DANM-CVX methods, respectively.

It is important to emphasize that the computational complexity analysis of ANM/DANM-ADMM-Net in this paper does not include the computational costs required for network training, since it is possible to train offline and apply online. Moreover, after training to obtain the optimal parameters, the computational complexity of networks and its ADMM algorithms are identical and differ in the iterative parameters. Therefore, networks and ANM/DANM-ADMM algorithms will have the same computational complexity when applied with the same network layers (number of iterations). However, considering the reduced iterations requirements of networks, the computational complexity is further decreased compared to algorithms.

The results in Figure 10 show that the growth rate of the proposed method's running time is much lower than that of the ANM-CVX and dual ANM methods, as the number of array elements or snapshots increases. In conclusion, ANM-ADMM-Net can be an alternative in the presence of larger-scale arrays or snapshots and high real-time requirements, and it is consistent with the above theoretical analysis. The same conclusion also can be drawn for DANM-CVX and DANM-ADMM-Net in Figure 11.

5. Conclusions

In this paper, a DU-based gridless DOA method is proposed. ANM-ADMM and DANM-ADMM algorithms are examined in the continuous domain, and the deep neural networks ANM-ADMM-Net and DANM-ADMM-Net are constructed for their problems. Their network structure, data set construction method, network initialization, and training method are introduced. Then, their performance is validated by simulation experiments. The results show that: compared with the existing methods, model-plus-data-driven ANM-ADMM-Net and DANM-ADMM-Net can learn the optimal iteration parameters from the data and quickly obtain more accurate 1D and 2D DOA estimation. The required computational complexity is reduced by 50~60 times and 20 times, respectively.

Author Contributions: Conceptualization, H.Z., W.F. and C.F.; methodology, H.Z., C.F. and W.F.; software, H.Z., B.Z. and T.M.; validation, H.Z., B.Z. and T.M.; formal analysis, W.F., C.F.; investigation, H.Z., W.F.; resources, W.F., H.Z.; data curation, H.Z., B.Z.; writing—original draft preparation, H.Z.; writing—review and editing, W.F., C.F.; visualization, H.Z., B.Z. and T.M.; supervision, T.M., H.Z.; project administration, W.F.; funding acquisition, W.F. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by: ① National Natural Science Foundation of China, No. 62001507; ② Young Talent fund of University Association for Science and Technology in Shaanxi, China, No. 20210106.

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

Appendix A

The following steps are elucidated to prove the iterative process of Algorithm 1. The five minimization subproblems of (16) can be given as

$$\left\{ \begin{array}{l} (1) : \underset{\mathbf{X}}{\operatorname{argmin}} \frac{1}{2} \|\mathbf{Y} - \mathbf{X}\|_F^2 + \Delta \\ (2) : \underset{\mathbf{W}}{\operatorname{argmin}} \frac{\tau}{2} T(\mathbf{W}) + \Delta \\ (3) : \underset{\mathbf{u}}{\operatorname{argmin}} \frac{\tau}{2} T(\mathbf{u}) + \Delta \\ (4) : \underset{\Theta \geq 0}{\operatorname{argmin}} \Delta \\ (5) : \underset{\Lambda}{\operatorname{argmin}} \left\langle \Lambda, \Theta - \begin{bmatrix} T(\mathbf{u}^{(k+1)}) & \mathbf{X}^{(k+1)} \\ (\mathbf{X}^{(k+1)})^H & \mathbf{W}^{(k+1)} \end{bmatrix} \right\rangle \end{array} \right. \quad (\text{A1})$$

where Θ and Λ are Hermitian matrices, and $\Theta_{\mathbf{W}}, \Lambda_{\mathbf{W}} \in \mathbb{C}^{L \times L}$, $\Theta_{T(\mathbf{u})} \in \mathbb{C}^{M \times M}$, $\Lambda_{T(\mathbf{u})} \in \mathbb{C}^{M \times M}$, and $\Theta = \begin{bmatrix} \Theta_{T(\mathbf{u})} & \Theta_{\mathbf{X}} \\ (\Theta_{\mathbf{X}})^H & \Theta_{\mathbf{W}} \end{bmatrix}$, $\Lambda = \begin{bmatrix} \Lambda_{T(\mathbf{u})} & \Lambda_{\mathbf{X}} \\ (\Lambda_{\mathbf{X}})^H & \Lambda_{\mathbf{W}} \end{bmatrix} \in \mathbb{C}^{(M+L) \times (M+L)}$.

Let the gradient of the objective function of the above five subproblems be equal to zero, then the $(k+1)$ -th iteration of the ADMM yields

$$\begin{cases} \mathbf{A}^{(k+1)} : \mathbf{X}^{(k+1)} = 1/(1+2\rho)(\mathbf{Y} + 2\mathbf{\Lambda}_X^{(k)} + 2\rho\mathbf{\Theta}_X^{(k)}) \\ \mathbf{B}^{(k+1)} : \mathbf{W}^{(k+1)} = \rho^{-1}\mathbf{\Lambda}_W^{(k)} + \mathbf{\Theta}_W^{(k)} - \tau/(2\rho)\mathbf{I}_L \\ \mathbf{C}^{(k+1)} : \mathbf{u}^{(k+1)} = \mathbf{\Gamma}(T^*(\rho^{-1}\mathbf{\Lambda}_{T(u)}^{(k)} + \mathbf{\Theta}_{T(u)}^{(k)}) - \tau/(2\rho)M\mathbf{e}_1) \\ \mathbf{D}^{(k+1)} : \mathbf{\Theta}^{(k+1)} = \mathbf{G}\text{diag}(\{\delta_g\}_+) \mathbf{G}^{-1} \\ \mathbf{E}^{(k+1)} : \mathbf{\Lambda}^{(k+1)} = \mathbf{\Lambda}^{(k)} + \rho \left(\mathbf{\Theta}^{(k+1)} - \begin{bmatrix} T(\mathbf{u}^{(k+1)}) & \mathbf{X}^{(k+1)} \\ (\mathbf{X}^{(k+1)})^H & \mathbf{W}^{(k+1)} \end{bmatrix} \right) \end{cases} \quad (\text{A2})$$

where $g = 1, 2, \dots, M+N$, $\{\delta_g\}_+$ denotes that all elements smaller than zero are set to zero, and $\mathbf{G}$ is an orthogonal matrix satisfying $\mathbf{G}\mathbf{G}^T = \mathbf{I}_{M+L}$.

For subproblem $\mathbf{C}^{(k+1)}$, its objective function can be derived from

$$(T(\mathbf{u}))^+ = \rho^{-1}\mathbf{\Lambda}_{T(u)}^{(k)} + \mathbf{\Theta}_{T(u)}^{(k)} - \tau/(2\rho)\mathbf{I}_M \quad (\text{A3})$$

In truth, the resulting $(T(\mathbf{u}))^+$ is a non-Toeplitz matrix at the beginning of the algorithm. Therefore, according to the Toeplitz property of $T(\mathbf{u})$, first, transform $(T(\mathbf{u}))^+$ into a vector $\mathbf{u}^{(k+1)}$, and then reassemble it to reach the accurate Toeplitz matrix $T(\mathbf{u}^{(k+1)})$ by the mapping $T(\cdot)$ in (5) for solving the latter function, i.e.,

$$\mathbf{u}^{(k+1)} = \mathbf{\Gamma}(T^*(\rho^{-1}\mathbf{\Lambda}_{T(u)}^{(k)} + \mathbf{\Theta}_{T(u)}^{(k)}) - \tau/(2\rho)M\mathbf{e}_1) \quad (\text{A4})$$

where $\mathbf{a} = T^*(\mathbf{A})$ denotes a mapping from a matrix $\mathbf{A} \in \mathbb{C}^{M \times M}$ to a vector $\mathbf{a} \in \mathbb{C}^{M \times 1}$ and satisfying $a_i = \text{sum}(\mathbf{A}_{p,q} | q - p + 1 = i)$, $\mathbf{\Gamma} = \text{diag}(\frac{1}{M}, \frac{1}{M-1}, \dots, \frac{1}{M-(M-1)})$. $\mathbf{e}_1 \in \mathbb{C}^{M \times 1}$ is the first column of the identity matrix.

For subproblem $\mathbf{D}^{(k+1)}$, its original objective function can be merged as

$$\underset{\mathbf{\Theta} \geq 0}{\text{argmin}} \frac{\rho}{2} \left\| \mathbf{\Theta}^{(k+1)} - \begin{bmatrix} T(\mathbf{u}^{(k+1)}) & \mathbf{X}^{(k+1)} \\ (\mathbf{X}^{(k+1)})^H & \mathbf{W}^{(k+1)} \end{bmatrix} + \rho^{-1}\mathbf{\Lambda}^{(k)} \right\|_F^2 + \text{const} \quad (\text{A5})$$

where $\text{const} = \frac{\rho}{2} \|\mathbf{\Lambda}^{(k)}\|_2^2$, $T(\mathbf{u}^{(k+1)})$ is a Toeplitz matrix obtained from subproblem $\mathbf{C}^{(k+1)}$.

Then, the derivative of (41) is obtained by

$$\tilde{\mathbf{\Theta}}^{(k+1)} = \begin{bmatrix} T(\mathbf{u}^{(k+1)}) & \mathbf{X}^{(k+1)} \\ (\mathbf{X}^{(k+1)})^H & \mathbf{W}^{(k+1)} \end{bmatrix} + \rho^{-1}\mathbf{\Lambda}^{(k)} \quad (\text{A6})$$

Let $\tilde{\mathbf{\Theta}}^{(k+1)} = \mathbf{G}\mathbf{\Sigma}\mathbf{G}^{-1}$, where the diagonal matrix $\mathbf{\Sigma} = \text{diag}(\{\delta_1, \delta_2, \dots, \delta_{M+L}\})$. Consider $\mathbf{\Theta} \geq 0$, thus there exists

$$\mathbf{\Theta}^{(k+1)} = \mathbf{G}\text{diag}(\{\delta_g\}_+) \mathbf{G}^{-1} \quad (\text{A7})$$

References

1. Mao, Z.; Liu, S.; Zhang, Y.D.; Han, L.; Huang, Y. Joint DoA-Range Estimation Using Space-Frequency Virtual Difference Coarray. *IEEE Trans. Actions Signal Process.* **2022**, *70*, 2576–2592. [\[CrossRef\]](#)
2. Liu, D.; Zhao, Y.; Zhang, T. Sparsity-Based Two-Dimensional DOA Estimation for Co-Prime Planar Array via Enhanced Matrix Completion. *Remote Sens.* **2022**, *14*, 4690. [\[CrossRef\]](#)
3. Pascal, V.; Mestre, X.; Loubaton, P. Performance analysis of an improved MUSIC DOA estimator. *IEEE Trans. Signal Process.* **2015**, *63*, 6407–6422.
4. Yu, J.; Li, J.; Sun, B.; Jiang, Y.; Xu, L. Multiple RFI sources location method combining two-dimensional ESPRIT DOA estimation and particle swarm optimization for spaceborne SAR. *Remote Sens.* **2021**, *13*, 1207. [\[CrossRef\]](#)
5. Haardt, M.; Zoltowski, M.D.; Mathews, C.P.; Nosske, J. 2D unitary ESPRIT for efficient 2D parameter estimation. In Proceedings of the 1995 IEEE International Conference on Acoustics, Speech, and Signal Processing, Detroit, MI, USA, 9–12 May 1995; Volume 3, pp. 2096–2099.
6. Xu, T.; Wang, X.; Huang, M.; Lan, X.; Sun, L. Tensor-based reduced-dimension music method for parameter estimation in monostatic fda-mimo radar. *Remote Sens.* **2021**, *13*, 3772. [\[CrossRef\]](#)
7. Tropp, J.A. Greed is Good: Algorithmic Results for Sparse Approximation. *IEEE Trans. Inf. Theory* **2004**, *50*, 2231–2242. [\[CrossRef\]](#)
8. Cao, Z.; Zhou, L.; Dai, J. Sparse Bayesian Approach for DOD and DOA Estimation with Bistatic MIMO Radar. *IEEE Access* **2019**, *7*, 155335–155346. [\[CrossRef\]](#)
9. Ramasamy, D.; Venkateswaran, S.; Madhoo, U. Compressive Parameter Estimation in AWGN. *IEEE Trans. Signal Process.* **2014**, *62*, 2012–2027. [\[CrossRef\]](#)
10. Bhaskar, B.N.; Tang, G.; Recht, B. Atomic norm denoising with applications to line spectral estimation. *IEEE Trans. Signal Process.* **2013**, *61*, 5987–5999. [\[CrossRef\]](#)
11. Chen, X.; Zhang, X.; Yang, J.; Tan, A. How to overcome basis mismatch: From atomic norm to gridless compressive sensing. *Acta Autom. Sin.* **2016**, *42*, 335–346.
12. Chandrasekaran, V.; Recht, B.; Parrilo, P.A.; Willsky, A.S. The convex geometry of linear inverse problems. *Found. Comput.-Tional Math.* **2012**, *12*, 805–849. [\[CrossRef\]](#)
13. Candes, E.J.; Romberg, J.K.; Tao, T. Stable signal recovery from incomplete and inaccurate measurements. *Commun. Pure Appl. Math. A J. Issued Courant Inst. Math. Sci.* **2006**, *59*, 1207–1223. [\[CrossRef\]](#)
14. Liu, S.; Mao, Z.; Zhang, Y.D.; Huang, Y. Rank minimization-based Toeplitz reconstruction for DoA estimation using coprime array. *IEEE Commun. Lett.* **2021**, *25*, 2265–2269. [\[CrossRef\]](#)
15. Chi, Y.; Chen, Y. Compressive two-dimensional harmonic retrieval via atomic norm minimization. *IEEE Trans. Signal Process.* **2014**, *63*, 1030–1042. [\[CrossRef\]](#)
16. Tang, W.G.; Jiang, H.; Pang, S.X. Grid-free DOD and DOA estimation for MIMO radar via duality-based 2D atomic norm minimization. *IEEE Access* **2019**, *7*, 60827–60836. [\[CrossRef\]](#)
17. Tian, Z.; Zhang, Z.; Wang, Y. Low-complexity optimization for two-dimensional direction-of-arrival estimation via decoupled atomic norm minimization. In Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, USA, 5–9 March 2017; pp. 3071–3075.
18. Pan, J.; Tang, J.; Niu, Y. Fast two-dimensional atomic norm minimization in spectrum estimation and denoising. *arXiv* **2018**, arXiv:1807.08606.
19. Zhang, Z.; Wang, Y.; Tian, Z. Efficient two-dimensional line spectrum estimation based on decoupled atomic norm minimization. *Signal Process.* **2019**, *163*, 95–106. [\[CrossRef\]](#)
20. Tang, W.G.; Jiang, H.; Zhang, Q. ADMM for gridless DOD and DOA estimation in bistatic MIMO radar based on decoupled atomic norm minimization with one snapshot. In Proceedings of the 2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP), Ottawa, ON, Canada, 11–14 November 2019; pp. 1–5.
21. Tang, W.G.; Jiang, H.; Pang, S.X. Gridless angle and range estimation for FDA-MIMO radar based on decoupled atomic norm minimization. In Proceedings of the ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 12–17 May 2019; pp. 4305–4309.
22. Gregor, K.; Lecun, Y. Learning fast approximations of sparse coding. In Proceedings of the 27th International Conference on International Conference on Machine Learning, Haifa, Israel, 21–24 June 2010; pp. 399–406.
23. Liu, J.; Chen, X. ALISTA: Analytic weights are as good as learned weights in LISTA. In Proceedings of the International Conference on Learning Representations (ICLR), New Orleans, LA, USA, 6–9 May 2019.
24. Borgerding, M.; Schniter, P.; Rangan, S. AMP-inspired deep networks for sparse linear inverse problems. *IEEE Trans. Signal Process.* **2017**, *65*, 4293–4308. [\[CrossRef\]](#)
25. Yang, C.; Gu, Y.; Chen, B.; Rangan, S. Learning proximal operator methods for nonconvex sparse recovery with theoretical guarantee. *IEEE Trans. Signal Process.* **2020**, *68*, 5244–5259. [\[CrossRef\]](#)
26. 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, 18–23 June 2018; pp. 1828–1837.
27. Chen, X.; Liu, J.; Wang, Z.; Pang, S. Theoretical linear convergence of unfolded ISTA and its practical weights and thresholds. *Adv. Neural Inf. Process. Syst.* **2018**, *31*. [\[CrossRef\]](#)

28. Metzler, C.A.; Mousavi, A.; Baraniuk, R.G. Learned D-AMP: Principled neural network based compressive image recovery. *arXiv* **2017**, arXiv:1704.06625.
29. Yang, Y.; Sun, J.; Li, H.; Rangan, S. 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. Dong, W.; Wang, P.; Yin, W.; Shi, G.; Wu, F.; Lu, X. Denoising prior driven deep neural network for image restoration. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *41*, 2305–2318. [[CrossRef](#)] [[PubMed](#)]
31. Zhu, H.G.; Feng, W.K.; Feng, C.Q.; Liu, W. Deep unfolding based space-time adaptive processing method for airborne radar. *J. Radars* **2022**, *11*, 676–691. [[CrossRef](#)]
32. Wu, L.; Liu, Z.; Liao, J. DOA Estimation Using an Unfolded Deep Network in the Presence of Array Imperfections. In Proceedings of the 2022 7th International Conference on Signal and Image Processing (ICSIP), Suzhou, China, 20–22 July 2022; pp. 182–187.
33. Guo, Y.; Jin, J.; Wang, Q.; Chen, H.; Liu, W. Position-enabled complex Toeplitz LISTA for DOA estimation with unknown mutual coupling. *Signal Process.* **2022**, *194*, 108422. [[CrossRef](#)]
34. Xiao, P.; Liao, B.; Deligiannis, N. Deepfpc: A deep unfolded network for sparse signal recovery from 1-bit measurements with application to doa estimation. *Signal Process.* **2020**, *176*, 107699. [[CrossRef](#)]
35. Wan, L.; Sun, Y.; Sun, L.; Ning, Z.; Rodrigues, J.J.P.C. Deep learning based autonomous vehicle super resolution DOA estimation for safety driving. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 4301–4315. [[CrossRef](#)]
36. Zhu, H.G.; Feng, C.Q.; Feng, W.K.; Chen, H. A deep learning approach for sparse snapshot doa estimation. *J. Signal Process.* **2022**, *38*, 2114–2123.
37. Li, Y.; Chi, Y. Off-the-Grid Line Spectrum Denoising and Estimation with Multiple Measurement Vectors. *IEEE Trans. Signal Process.* **2015**, *64*, 1257–1269. [[CrossRef](#)]
38. Toh, K.C.; Todd, M.J.; Tütüncü, R.H. On the implementation and usage of SDPT3—a Matlab software package for semidefinite-quadratic-linear programming, version 4.0. In *Handbook on Semidefinite, Conic and Polynomial Optimization*; Springer: Boston, MA, USA, 2012; pp. 715–754.
39. Yang, L.; Xie, L.; Stoica, P. Vandermonde Decomposition of Multilevel Toeplitz Matrices with Application to Multidimensional Super-Resolution. *IEEE Trans. Inf. Theory* **2016**, *62*, 3685–3701. [[CrossRef](#)]
40. Tang, W.G.; Jiang, H.; Zhang, Q. Gridless DOD and DOA estimation in bistatic MIMO radar using 2D-ANM and its low complexity algorithms. *Digit. Signal Process.* **2021**, *108*, 102900. [[CrossRef](#)]
41. Boyd, S.; Parikh, N.; Chu, E. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.* **2011**, *3*, 1–122. [[CrossRef](#)]
42. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning representations by back-propagating errors. *Nature* **1986**, *323*, 533–536. [[CrossRef](#)]
43. Krishnan, K.; Terlaky, T. Interior point and semidefinite approaches in combinatorial optimization. In *Graph Theory and Combinatorial Optimization*; Springer: Boston, MA, USA, 2005; pp. 101–157.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.