remote sensing

MDPI

Technical Note

Gridless Underdetermined DOA Estimation for Mobile Agents with Limited Snapshots Based on Deep Convolutional Generative Adversarial Network

Yue Cui [1], Feiyu Yang [1], Mingzhang Zhou [2,3], Lianxiu Hao [4], Junfeng Wang [2,5,*], Haixin Sun [2,3], Aokun Kong [1] and Jiajie Qi [1]

[1] College of Computer and Information Engineering, Tianjin Normal University, Tianjin 300387, China; cuiyue@tjnu.edu.cn (Y.C.); 2111090052@stu.tjnu.edu.cn (F.Y.); 2211090049@stu.tjnu.edu.cn (A.K.); 2111090053@stu.tjnu.edu.cn (J.Q.)
[2] Key Laboratory of Southeast Coast Marine Information Intelligent Perception and Application, Ministry of Natural Resources, Zhangzhou 363000, China; mzzhou@xmu.edu.cn (M.Z.); hxsun@xmu.edu.cn (H.S.)
[3] School of Informatics, Xiamen University, Xiamen 361005, China
[4] School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, China; tjhlx@163.com
[5] School of Integrated Circuit Science and Engineering, Tianjin University of Technology, Tianjin 300384, China
* Correspondence: jfwang@tjut.edu.cn

Abstract: Deep learning techniques have made certain breakthroughs in direction-of-arrival (DOA) estimation in recent years. However, most of the current deep-learning-based DOA estimation methods view the direction finding problem as a grid-based multi-label classification task and require multiple samplings with a uniform linear array (ULA), which leads to grid mismatch issues and difficulty in ensuring accurate DOA estimation with insufficient sampling and in underdetermined scenarios. In order to solve these challenges, we propose a new DOA estimation method based on a deep convolutional generative adversarial network (DCGAN) with a coprime array. By employing virtual interpolation, the difference co-array derived from the coprime array is extended to a virtual ULA with more degrees of freedom (DOFs). Then, combining with the Hermitian and Toeplitz prior knowledge, the covariance matrix is retrieved by the DCGAN. A backtracking method is employed to ensure that the reconstructed covariance matrix has a low-rank characteristic. We performed DOA estimation using the MUSIC algorithm. Simulation results demonstrate that the proposed method can not only distinguish more sources than the number of physical sensors but can also quickly and accurately solve DOA, especially with limited snapshots, which is suitable for fast estimation in mobile agent localization.

Keywords: direction-of-arrival (DOA) estimation; deep learning; matrix recovery

check for updates

Citation: Cui, Y.; Yang, F.; Zhou, M.; Hao, L.; Wang, J.; Sun, H.; Kong, A.; Qi, J. Gridless Underdetermined DOA Estimation for Mobile Agents with Limited Snapshots Based on Deep Convolutional Generative Adversarial Network. *Remote Sens.* **2024**, *16*, 626. https://doi.org/10.3390/rs16040626

Academic Editor: Andrzej Stateczny

Received: 19 December 2023
Revised: 19 January 2024
Accepted: 22 January 2024
Published: 8 February 2024

Copyright: © 2024 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

In the past few decades, direction-of-arrival (DOA) estimation has emerged a critical issue across various domains, including radar, sonar, mobile communication and localization. To perform DOA estimation in actual environments, researchers have conducted in-depth studies and developed two main types of methods: physical model-driven methods [1–5] and data-driven methods [6–8]. The DOA estimation method based on phase interferometry is proposed in [1] for real-time localization. This method can compute the DOA in real time with lightweight architecture and full-digital dedicated hardware. However, it has implications for phase ambiguity and phase error, and could only distinguish a low number of receivers, with no ability to accurately estimate more DOAs at the same time. The high-resolution DOA estimation methods, such as the multiple signal classification (MUSIC) algorithm in [2] and the estimation of signal parameters via rotational invariance techniques (ESPRIT) algorithm in [3], could estimate more DOAs of signals and achieve more accurate

performance. Nevertheless, the number of DOAs they can distinguish is still limited by the number of physical sensors, and the computational complexity remains very high. To cope with a multipath environment, the forward/backward spatial smoothing (FBSS) algorithm is proposed in [4] to decorrelate the coherent signals, but its degrees of freedom (DOFs) are reduced, and the required SNR is slightly higher. In [5], the DOA estimation algorithm for coherent GPS signals not only employs Toeplitz decorrelation but also oblique projection to suppress noise at low SNR. The aforementioned physical model-driven methods usually require a number of snapshots and have high computational complexity and lengthy solution time. Moreover, those methods are based on a rigorous physical model; once in non-ideal conditions, such as a limited number of snapshots or model mismatch, their estimation performance would be degraded obviously. Among data-driven methods, deep neural network models have shown excellent performance and lower computational complexity. The literature [8] introduces a deep neural network (DNN) model, which exhibits robustness in the presence of defective arrays. In [9], the authors used a convolutional neural network (CNN) for DOA estimation in low SNR conditions. In [10], it is proved that the columns of the covariance matrix can be expressed as linear measurements of undersampling noise in the spatial spectrum, and a deep convolutional neural network (DCNN) was built using sparse priors. In response to the significant reduction in estimation accuracy of existing methods for a multipath environment, reference [11] proposes a phase enhancement model based on a CNN for coherent DOA estimation which improves DOA estimation accuracy by enhancing phase and reducing phase distortion. In addition, the authors evaluate the importance of the phase feature for DOA estimation accuracy and demonstrated that the amplitude feature is redundant for DOA estimation. In [12], residual neural networks (ResNet) were used to achieve DOA estimation in a single snapshot. In [13], deep learning was applied to DOA estimation in underwater acoustic arrays. In [14], the authors present a novel DOA estimation framework that utilizes a complex-valued deep learning technique. In [15], researchers used the upper triangular region data of the received signal covariance matrix for training, effectively reducing training complexity and accelerating training speed. In [16], an angle separation deep learning method is proposed to achieve near-real-time DOA estimation for coherent signal sources. Furthermore, the lightweight DNN DOA estimation method for array imperfection correction has lower computational complexity and faster running speed, making it suitable for real-time signal processing application [17]. In [18], deep residual learning was used to achieve wideband DOA estimation. In addition, the DOA estimation method based on unsupervised learning with sparse array employs ResNet, which can effectively cope with low SNR and few snapshots scenarios [19]. However, the aforementioned methods did not consider the underdetermined scenario.

With the continuous development of the Internet of Things (IoT) and Internet of Vehicles (IoV), the number of intelligent mobile agents is growing constantly. In the process of localization and communication, the number of estimated targets is often greater than the number of array sensors, which results in the frequent occurrence of underdetermined situations. Moreover, the mobile agents require fast calculation speed with limited snapshots, which places higher requirements on the running speed of the DOA estimation algorithm. However, most of the current deep-learning-based DOA estimation methods use CNN models [10,11,20], treating the direction finding problem as a multi-label classification task and requiring multiple samplings with a uniform linear array (ULA). The output of the network in these methods is the probability associated with each corresponding label. These methods not only suffer from grid mismatch problems but are also unable to distinguish all targets in underdetermined situations, which would decrease the estimation accuracy dramatically. In [20], the sparse array was adopted, and its covariance matrix was recovered from the first row using a CNN-based regression method. Then, the DOA was obtained with the Root-MUSIC algorithm from the recovered covariance matrix. This approach has the ability to cope with underdetermined situations but cannot guarantee the

low-rank characteristic of the recovered covariance matrix, so its DOA estimation accuracy is constrained, especially with limited snapshots.

Therefore, in order to address the aforementioned challenges, a virtual ULA was constructed in this study by filling the virtual sensors into the difference co-array derived from the coprime array, which can obtain more DOFs and improve DOA estimation resolution. The deep convolutional generative adversarial network (DCGAN) was adopted to recover the data associated with the virtual sensors and rebuild the covariance matrix of the virtual ULA using the Hermitian and Toeplitz prior knowledge. In order to ensure the low-rank characteristic of the covariance matrix, the output data of the DCGAN were further processed using the low-rank matrix optimization algorithm. Finally, DOA estimation was performed using the MUSIC algorithm. The proposed method not only has the ability to cope with underdetermined scenarios but can also improve the accuracy and estimation speed with limited snapshots.

The remaining sections of this paper are organized as follows. Section 2 introduces the signal model. Section 3 elaborates on the structure and processing details of the proposed method. Section 4 describes the loss function used by the network and some important parameters. Section 5 provides experimental results. The last section summarizes the entire paper.

2. Signal Model

It is presumed that K far-field narrow-band source signals impinge onto an M-element array antenna (K>M), and the received signal at the array is given by:

$$\boldsymbol{X}(t) = \boldsymbol{A}(\boldsymbol{\theta})\mathbf{s}(t) + \mathbf{n}(t), t = 1, 2, \ldots, T, \tag{1}$$

where $\boldsymbol{\theta}$, $\boldsymbol{A}$, and T represent the source direction vector, array manifold matrix, and snapshot number, respectively. $\mathbf{s}(t)$ and $\mathbf{n}(t)$ denote the spatial signal vector and additive Gaussian white noise vector at time t, respectively. The k-th column of the array manifold matrix $\boldsymbol{A}$ can be represented as $\mathbf{a}(\theta) = \left[e^{\frac{-j*2\pi u_1 d\sin(\theta)}{\lambda}}, e^{\frac{-j*2\pi u_2 d\sin(\theta)}{\lambda}}, \ldots, e^{\frac{-j*2\pi u_M d\sin(\theta)}{\lambda}}\right]^T$, where $u_i, (i = 1, 2, \ldots M)$ represents the i-th element position.

A coprime array is constructed with two sparse uniform linear sub-arrays with $I + J - 1$ sensors, the first sub-array being $[0, Id, 2Id, \ldots, (J-1)Id]$ and the second sub-array being $[0, Jd, 2Jd, \ldots, (I-1)Jd]$, where I and J are coprime integers. The two sub-arrays do not overlap except for position 0. The structure of the coprime array is depicted in Figure 1a. The covariance matrix of the received signal $\boldsymbol{X}(t)$ with the coprime array can be expressed as

$$\boldsymbol{R}_X = E\left[\boldsymbol{X}(t)\boldsymbol{X}^H(t)\right] = \sum_{k=1}^{K} p_k \mathbf{a}(\theta_k)\mathbf{a}^H(\theta_k) + \sigma^2\boldsymbol{I}, \tag{2}$$

where p_k denotes the power of the k-th source signal, and $\boldsymbol{I}$ denotes the identity matrix. Afterward, by vectorizing the covariance matrix $\boldsymbol{R}_X$ and taking the distinct elements, the equivalent virtual signal of the difference co-array can be obtained as

$$\mathbf{y}_d = \boldsymbol{A}_d\mathbf{p} + \sigma^2\mathbf{i}, \tag{3}$$

where $\boldsymbol{A}_d = [\mathbf{a}^*(\theta_1) \otimes \mathbf{a}(\theta_1), \mathbf{a}^*(\theta_2) \otimes \mathbf{a}(\theta_2), \ldots, \mathbf{a}^*(\theta_K) \otimes \mathbf{a}(\theta_K)] \in \mathbb{C}^{[2I(J-1)+1]\times K}$, $\otimes$ denotes the Kronecker product, $\mathbf{p} = [p_1, p_2, \ldots p_K]^T$ and $\mathbf{i} = vec(\boldsymbol{I})$. The difference co-array contains a few missing elements that are called holes. The array structure of the difference co-array is shown in Figure 1b.

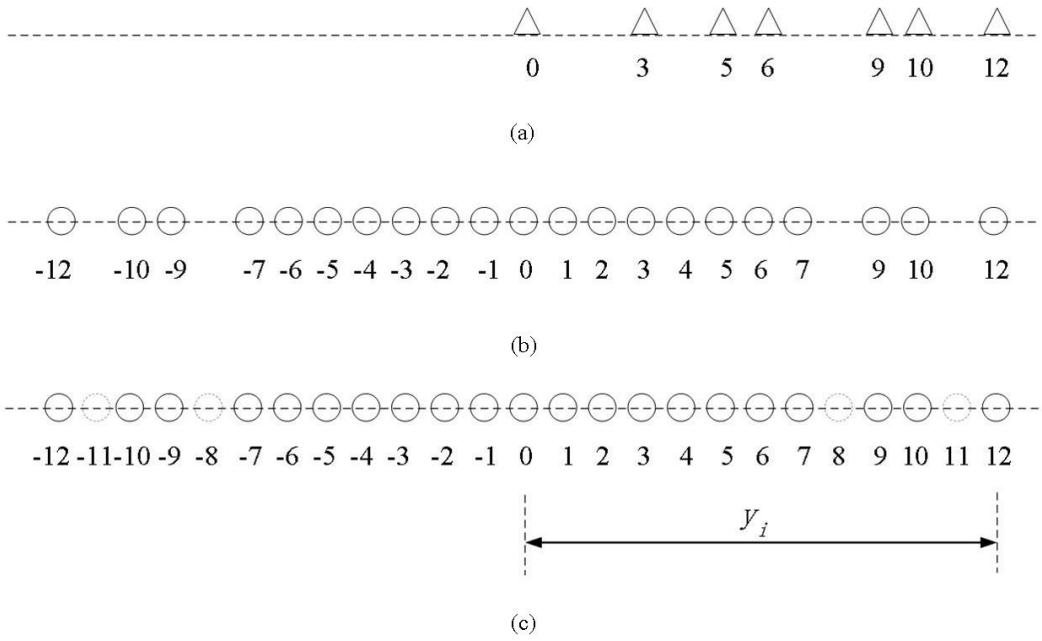


Figure 1. Array structures. (**a**) The coprime array for I = 3, J = 5. (**b**) The difference co-array derived from coprime array. (**c**) The virtual ULA when the number of sensors is 13.

So as to fully utilize the available information and increase DOFs, by filling the interpolated virtual sensors, the model can be extended further as a virtual ULA with $N = \max\{(I-1)J, (J-1)I\} + 1$ sensors, as shown in Figure 1c. The virtual ULA corresponds to a binary vector $\boldsymbol{v}$ of 0s and 1s, in which 0 represents the interpolated virtual element and 1 stands for the others. Correspondingly, the received signal $\mathbf{y}_d$ is extended to the N dimension vector $\mathbf{y}_i$, which has some zero elements corresponding to the virtual received signal of interpolated virtual sensors. As demonstrated in [21], the covariance matrix $\boldsymbol{R}_v$ of the received signal with the virtual ULA is equal to the Toeplitz matrix $\mathcal{T}(\mathbf{y}_i)$ with vector $\mathbf{y}_i$ as its first row, which can be represented as

$$\mathcal{T}(\mathbf{y}_i) = \boldsymbol{R}_v. \tag{4}$$

In actual application, because the received signals of the interpolated virtual sensors in virtual ULA default to 0, some elements in covariance matrix $\boldsymbol{R}_v$ are also set to 0. Compared with the covariance matrix $\boldsymbol{R}$ of the actual ULA with N physical sensors, the covariance matrix of the virtual ULA and actual ULA has the following relationship

$$\mathcal{T}(\mathbf{y}_i) = \boldsymbol{R}_v = \boldsymbol{R} \odot \boldsymbol{L}, \tag{5}$$

where $\odot$ denotes the Hadamard product, $\boldsymbol{L} = \boldsymbol{v} * \boldsymbol{v}^T$ is a binary matrix to imply the zero and non-zero elements in $\boldsymbol{R}_v$ and $\boldsymbol{R}$ is the covariance matrix associated with the actual ULA with N elements. Our focus is to rebuild the covariance matrix $\boldsymbol{R}$ of the virtual ULA from $\mathcal{T}(\mathbf{y}_i)$ with some missing elements.

As a priori knowledge, a covariance matrix should be a Hermitian matrix with a Toeplitz structure and has a low-rank characteristic in theory. Therefore, in order to reconstruct the covariance matrix accurately and quickly, we adopted some measures to ensure that the recovered $\boldsymbol{R}_{res}$ has the above characteristics. Here, we took the average of the values in the conjugate symmetric part of the generated matrix so as to limit the changes in the non-missing part to the minimum range. Finally, the backtracking method further ensures the positive definiteness of the covariance matrix.

3. The Proposed Method

Depicted in Figure 2, the proposed model framework consists of three components: preprocessing, the DCGAN structure and post-processing.

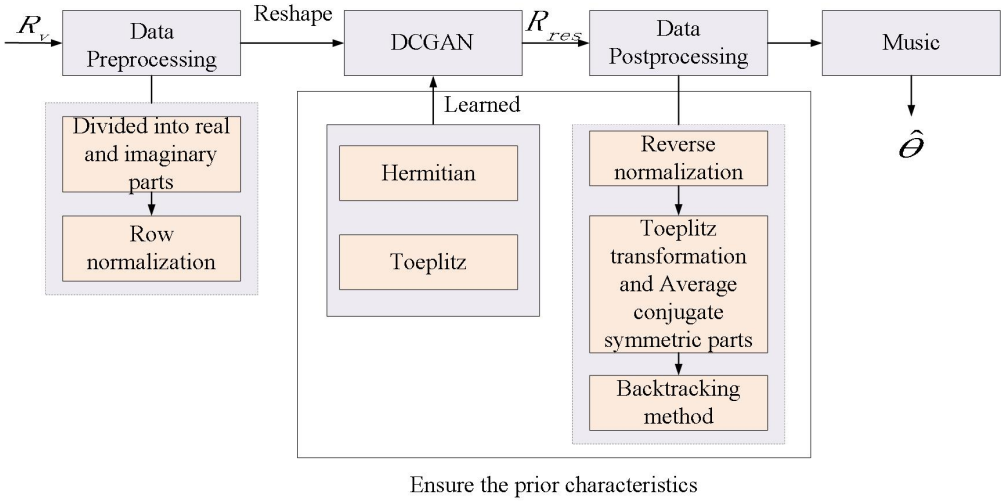


Figure 2. Framework of proposed model.

Firstly, we assume that the signal $\boldsymbol{X}$ is collected by T snapshots with a coprime array. The preprocessing part calculates the covariance matrix through the raw data, which is then normalized to the range of $[-1, 1]$. This is to reduce the range of values for different features to the same range in order to accelerate training speed and improve model stability. Subsequently, the covariance matrix $\boldsymbol{R}_v$ is transformed into a two-channel tensor. The DCGAN structure is responsible for reconstructing the covariance matrix with a virtual ULA from noise signals. The generator produces a result that is similar to the real covariance matrix. Finally, the post-processing part ensures the low-rank characteristic of the recovered covariance matrix and solves the DOA using the generated output.

3.1. Data Preprocessing

In order to adapt to the input requirements of the DCGAN, we used $\boldsymbol{R}_v$ and $\boldsymbol{R}^{'}$ as two inputs for the DCGAN, both of which are real tensors. In the experiment, since the covariance matrix $\boldsymbol{R}$ is a theoretical value and unknown, its sampled value $\boldsymbol{R}^{'}$ was used with N-elements ULA. The first dimension represents the real part matrix $\boldsymbol{R}_v[1, :, :] = Real(\boldsymbol{R}_v)$, and the second dimension represents the imaginary part matrix $\boldsymbol{R}_v[2, :, :] = Imag(\boldsymbol{R}_v)$. According to the structure of the DCGAN generator, the generator restricts the output data to the range of $[-1, 1]$. In order to speed up the training process, we performed row-wise normalization of the real and imaginary parts. It is also helpful to create different features with the same scale, which leads to easier optimization. Moreover, normalizing the input data can effectively prevent gradient explosion and mode collapse, which can better balance the generator and discriminator and improve the stability and robustness of the model.

3.2. DCGAN Structure

For the DCGAN, the proposed design is illustrated in Figure 3.

We approach the covariance matrix reconstruction as a restoration task aiming to compute the mapping correlation between $\boldsymbol{R}_v$ and $\boldsymbol{R}^{'}$, so that the generated $\boldsymbol{R}_{\mathbf{res}}$ is as close as possible to $\boldsymbol{R}^{'}$. The DCGAN consists of a generator with a transposed convolutional structure and a discriminator with a convolutional structure. The transposed convolutional structure in the generator allows for a more suitable upsampling method based on the dataset. Following each transposed convolutional layer in the generator, a ReLU activation

function and a batch normalization layer are applied. The final layer of the generator network utilizes a Tanh activation function. A convolutional structure was adopted for the discriminator. Following each convolutional layer, a LeakyReLU activation function and a batch normalization layer are utilized. The final layer uses a Sigmoid activation function. The LeakyReLU activation function retains a small gradient for the negative part, facilitating higher quality recovery by the generator. Additionally, a dropout layer is incorporated into the discriminator to balance training.

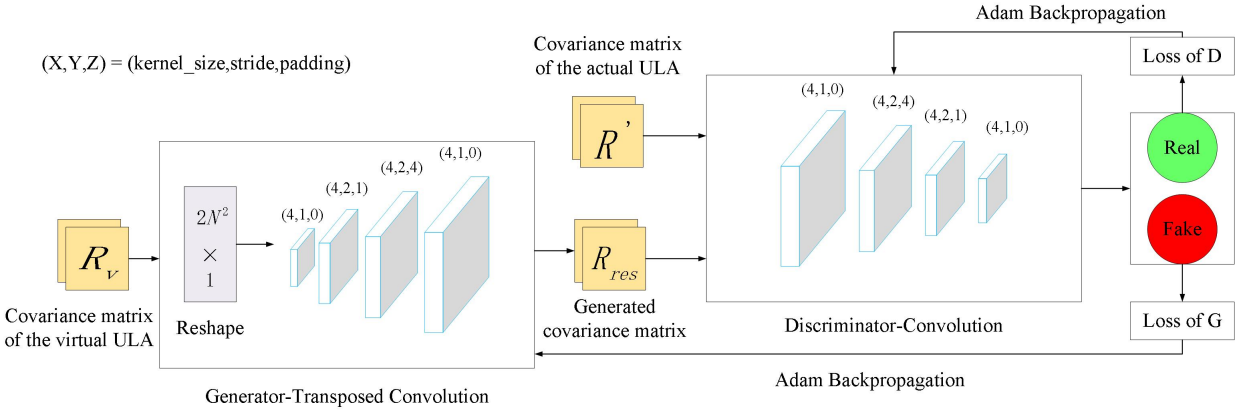


Figure 3. DCGAN structure.

The proposed model widely uses batch normalization layers due to their ability to prevent overfitting and accelerate the training and convergence process. However, it is important to note that batch normalization layers are not used in the input layer of the generator or the output layer of the discriminator, since this may cause sample oscillation and model instability. The DCGAN structure does not have pooling layers or fully connected layers because pooling operations may lose some important information, and the use of fully connected layers is prone to overfitting. The final output shape of the generator is $2 \times N \times N$.

3.3. Data Post-Processing

Finally, for the data post-processing part, it should be noted that the last layer of the generator uses the Tanh activation function. Therefore, we used the saved parameters to reverse-normalize the network output back to its original values. In addition, although the generated data roughly conform to the distribution of $\boldsymbol{R}'$, the data do not strictly satisfy the conjugate symmetry. Therefore, the average of the conjugate symmetric parts of the real part matrix was directly calculated. The diagonal data of the imaginary matrix were set to 0, the absolute values of the conjugate symmetric parts of other data were taken, the average was calculated and positive and negative signs were assigned. Strict adherence to this property was ensured. Furthermore, since the training strategy involves real and imaginary dual channels, the two-channel real value data for each recovered covariance matrix were combined into a complex-valued matrix for DOA estimation. Finally, in order to ensure positive definiteness of the complex-valued matrix, we utilized the low-rank matrix optimization algorithm to regularize this matrix.

4. Training Approach

4.1. Loss Function

For small-scale tasks, cross-entropy loss is sufficient for network training. However, during the experimental process, it was found that a single cross-entropy loss led to difficulty in limiting the recovery direction of the covariance matrix. Therefore, in this study, a combined approach of generator loss, discriminator loss, context loss, perceptual loss and nuclear norm loss was adopted for training. Both generator loss and discriminator

loss are cross-entropy losses, and the input of the cross-entropy loss is a pair of outputs from the generator or discriminator and the corresponding size label. The labels of the real data have been smoothed using 0.9 to maintain balance at both ends. The perceptual loss is generated by the DCGAN itself and can be represented as

$$L_{perceptual} = \log(1 - \mathbf{D}(\mathbf{G}(\boldsymbol{R}_v))), \tag{6}$$

where $\mathbf{D}(\cdot)$ represents the discriminator, and $\mathbf{G}(\cdot)$ stands for the generator.

The context loss constrains the consistency of the non-missing parts of the covariance matrix and aims to minimize changes in non-missing parts during the recovery process. The L_2 norm is employed to calculate the loss, and the inputs of the context loss are $\boldsymbol{R}^{'}$,$\mathbf{G}(\boldsymbol{R}_v)$, and $\boldsymbol{L}$, which can be represented as

$$L_{contextual} = \left\| \boldsymbol{L} \odot \mathbf{G}(\boldsymbol{R}_v) - \boldsymbol{L} \odot \boldsymbol{R}^{'} \right\|_2. \tag{7}$$

The nuclear norm loss serves as a regularization constraint to reduce the rank of the restored covariance matrix. It can reduce the number of unknown values that need to be restored. Additionally, the nuclear norm loss helps control the complexity of the matrix to avoid overfitting. The nuclear norm loss can be represented as

$$\|\boldsymbol{R}_{res}\|_* = \sum_{i=1}^{N} \sigma_i(\boldsymbol{R}_{res}), \tag{8}$$

where $\boldsymbol{R}_{res}$ is the covariance matrix generated by the generator, N represents the number of rows and columns in the covariance matrix and $\sigma_i(\boldsymbol{R}_{res})$ represents the i-th singular value of $\boldsymbol{R}_{res}$.

Cross-entropy loss provides backpropagation gradients and other parameters. The entire restoration task's loss function can be represented as

$$L_{total} = L_{contextual} + \lambda_1 L_{perceptual} + \lambda_2 L_{norm}, \tag{9}$$

where λ_1 and λ_2 are hyperparameters used to adjust the importance of the two losses.

Therefore, our goal is to ensure the stability of the non-missing parts while guiding the generator to produce globally consistent results with the real covariance matrix. This can further improve the accuracy of subsequent DOA estimation.

4.2. DCGAN Training

To construct the dataset, we randomly selected two angles within the range of $[-60^\circ, 60^\circ]$. Each data point was then generated based on the signal model. The dataset has SNR values ranging from -5 dB to 10 dB, with a size of 1,000,000. The training set consists of 80% of the data, and the remaining 20% are used for validation. The model employs the Adam optimizer to update the weights, and hyperparameters λ_1 and λ_2 of the total loss of the recovery task were all set to 0.1.

The model initializes its weights from a normal distribution $N\left(0, 0.02^2\right)$. After initialization, the model immediately applies these weights. Unlike generative tasks, the generator's input $\boldsymbol{R}_v$ is reshaped into a vector of shape $(N \times N \times 2, 1)$ rather than random noise, which can utilize the prior characteristics of the covariance matrix.

5. Simulation Results

We conducted several experiments to demonstrate the performance of the proposed method. Based on individual experiment results and quantitative experimental results, we compared this approach with some other methods. In this study, all experiments were conducted on a desktop computer equipped with an Intel Core i7-12700F processor running at 3.5 GHz, with 16 GB of RAM and an NVIDIA GeForce RTX 4060Ti GPU (Galax, Hong Kong, China). The operating system used is Windows 10. The software environment uses

Python 3.6.5 as the programming language and uses the PyTorch framework for training and testing deep learning models.

5.1. Single Experiment Results

The proposed method was tested with a physical array consisting of seven sensors. We conducted the experiments with a fixed snapshot count of 256 and SNR at 10 dB. Two scenarios were considered: one with five signal sources (less than seven) and another with eight signal sources (greater than seven). In both scenarios, we employed the following comparison algorithms: the MUSIC algorithm, the sparse representation with l_p-norm algorithm (MAP) [22], the sparse-recovery-based method (SR-D) [23] and the CNN-based DOA estimation method (CNN-D) [20].

As shown in Figure 4, when the number of signal sources is five (less than seven), we assume that five uncorrelated signals originate from $[-43°, -29°, 10°, 32°, 54°]$. It is visible that all of the aforementioned methods can achieve good performance and provide accurate DOA estimation.

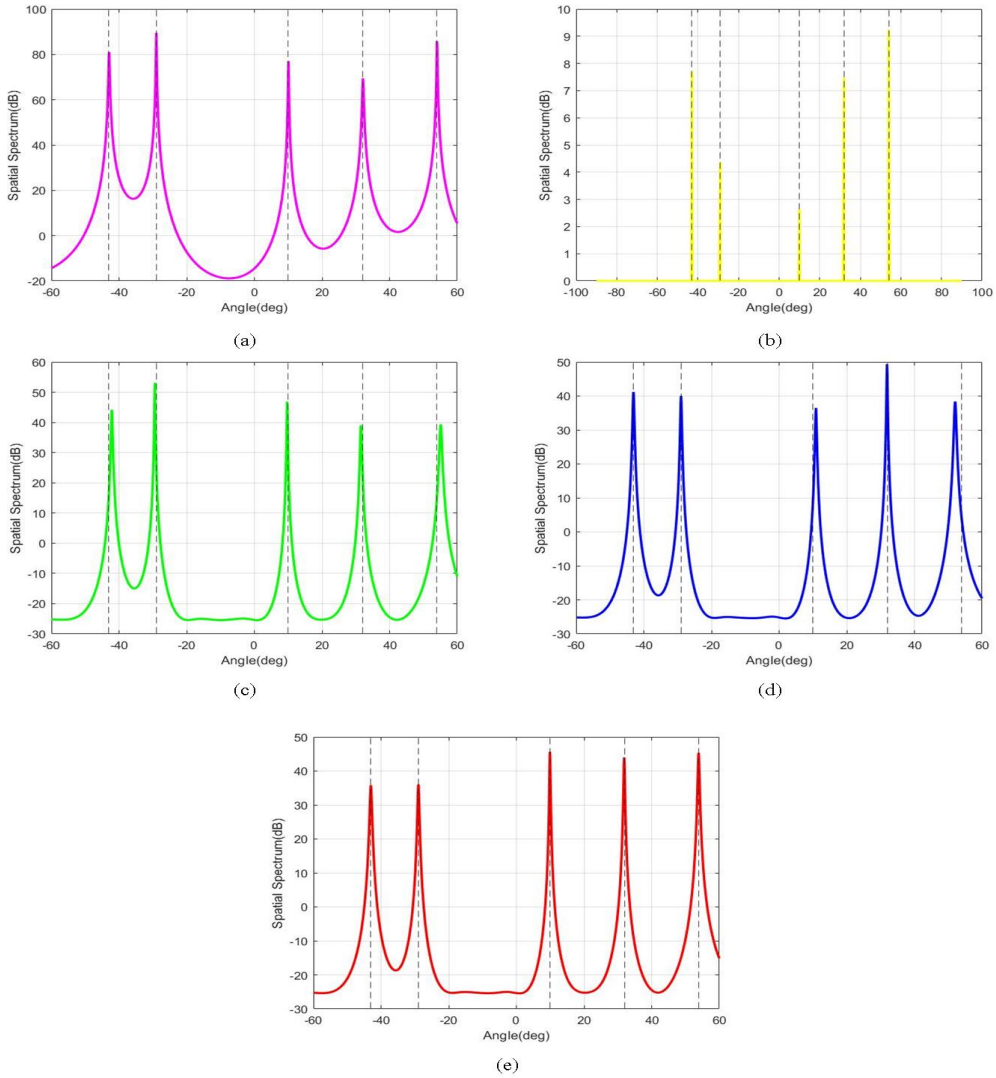


Figure 4. Spectrum of DOA estimation methods when the number of signal sources is five. (**a**) MUSIC. (**b**) MAP. (**c**) SR−D. (**d**) CNN−D. (**e**) Proposed method.

However, as depicted in Figure 5, when the number of signal sources increases to eight (greater than seven), these signal sources arrive from $[-43°, -29°, -16°, 0°, 10°, 21°, 32°, 54°]$. The spatial spectrum of the MUSIC algorithm becomes flattened, and some spectral peaks

merge together. The MAP algorithm can only accurately estimate partial angles of arrival. Both of these two algorithms fail to distinguish more sources than the number of physical sensors. Although the CNN-D method and SR-D method can obtain eight spectral peaks, their peaks exhibit some bias, which leads to a decrease in the accuracy of these DOA estimation algorithms. In contrast, the proposed method still forms eight sharp peaks at the actual DOAs, which is more than the number of physical sensors (seven). It can achieve more accurate DOA estimation in underdetermined scenarios, which is because the proposed method extends the DOFs of the virtual ULA to 12 and reconstructs its covariance matrix accurately using the DCGAN with the prior knowledge.

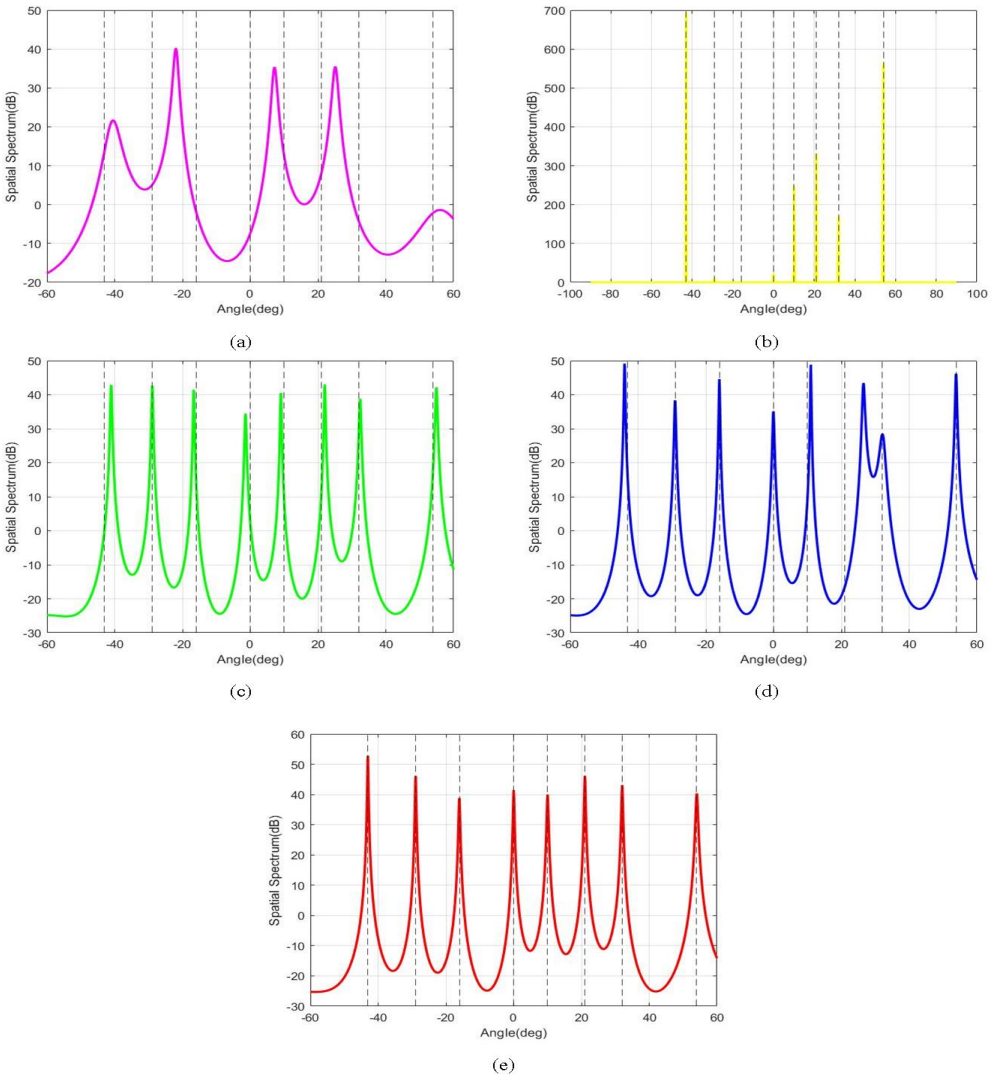


Figure 5. Spectrum of DOA estimation methods when the number of signal sources is eight. (**a**) MUSIC. (**b**) MAP. (**c**) SR−D. (**d**) CNN−D. (**e**) Proposed method.

5.2. Quantitative Experimental Results

To evaluate the performance of the proposed DCGAN method, we compared it with two existing methods: CNN-D and SR-D. The evaluation is based on the root mean square error (RMSE) metric. We constructed a coprime array using the coprime pairs of 3 and 5, with the element positions being $\{0, 3, 5, 6, 9, 10, 12\}d$. Furthermore, experiments were conducted at SNR values of $[-5, 0, 5, 10]$ dB.

As presented in Figure 6, when the quantity of snapshots is held constant at 256, the performance of the proposed method improves consistently as the SNR increases and surpasses the other methods, especially in low SNR conditions.

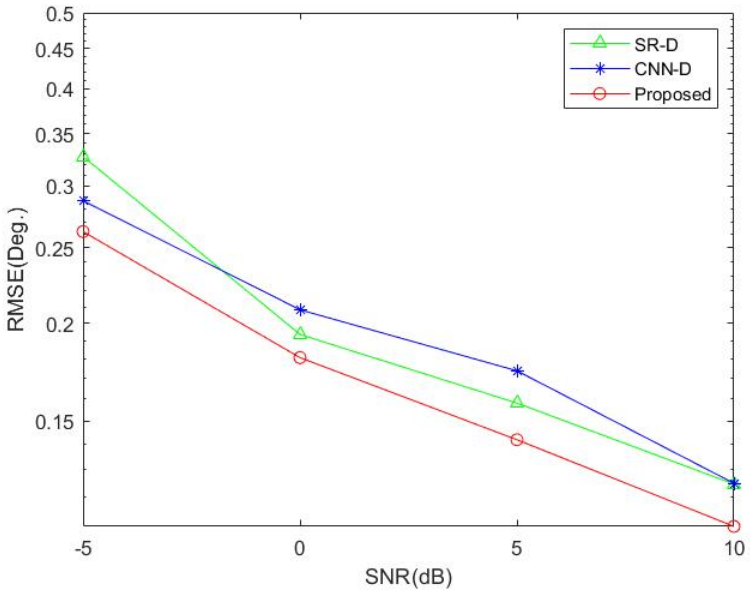


Figure 6. RMSE versus SNR.

Then, as illustrated in Figure 7, the SNR was fixed at 10 dB, and we compared the performance of the above methods with different numbers of snapshots. The proposed method does not experience a significant performance degradation as snapshots decrease and outperforms other methods. This is because the covariance matrix could be accurately rebuilt although with the limited snapshots, which preserves low-rank characteristics and more DOFs.

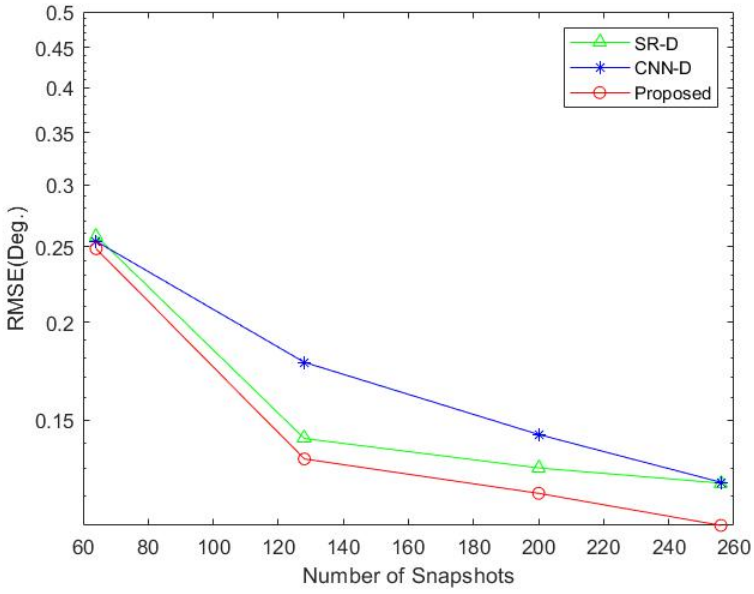


Figure 7. RMSE versus snapshots.

Next, as shown in Figure 8, we demonstrate the RMSE of these methods at different angle separation degrees. It is apparent that the proposed method exhibits robust performance at different resolutions, without significant fluctuations, and exhibits considerable robustness. The CNN-D method only uses the first-row elements to recover the covariance matrix and is incapable of guaranteeing the positive definiteness of the resulting covariance matrix.

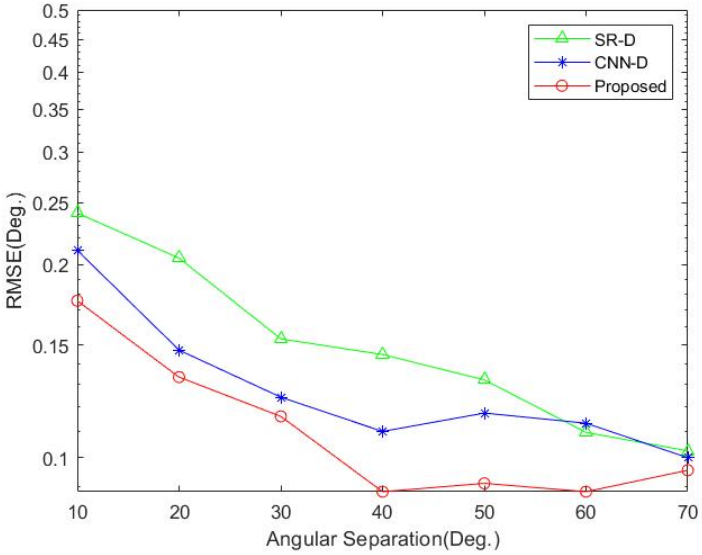


Figure 8. RMSE versus angular separation.

In the subsequent analysis, we investigated the influence of different snapshots and SNR levels on the performance of the proposed method. As shown in Figure 9, the performance of the proposed method improves with an increase in the number of snapshots. It can be observed from the figure that when the snapshot count is greater than 256, the performance of the proposed method stabilizes. Even with a relatively limited number of snapshots, the proposed method can achieve accurate DOA estimation without excessive performance loss. Furthermore, the performance of the proposed method continuously improves with an increase in SNR and stabilizes at a level of 10 dB.

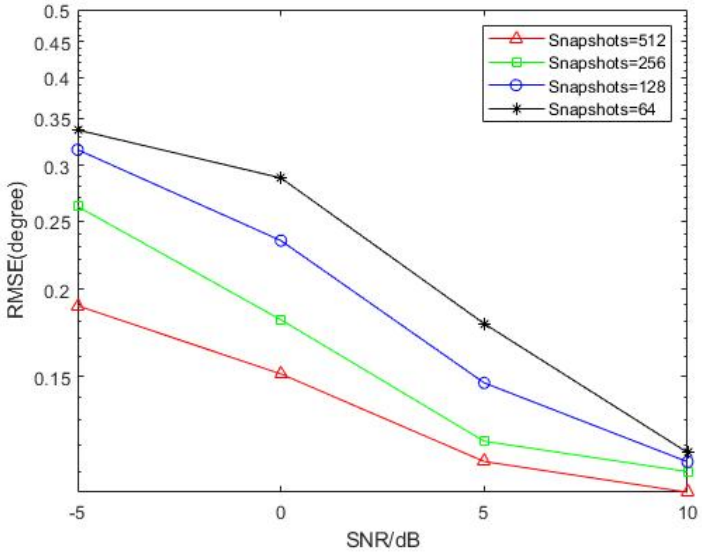


Figure 9. RMSE of the proposed method with different SNRs and snapshots.

Finally, we performed 10,000 Monte Carlo simulations and recorded the estimated total time results in Table 1. It should be noted that, to ensure the accuracy of the model, the deep learning methods mentioned above require a long training period, so well-trained models were used for testing. From the results, it can be seen that compared to the traditional physics-based model SR-D, the proposed method can achieve faster estimation time, with a decrease of about 30 times. Compared to the CNN-D method, especially at lower SNR from −5 dB to 5 dB, our estimation time is about 10–30 s faster, although, at

10 dB SNR, the proposed method also has a slight improvement. Combining Figure 7 and Table 1, it becomes evident that the proposed method is capable of achieving fast estimation even with limited snapshots. Therefore, it is suitable for fast DOA estimation in mobile agent localization scenarios.

Table 1. DOA estimation times.

Method	−5 dB	0 dB	5 dB	10 dB
SR-D	5890 s	5290 s	5720 s	5810 s
CNN-D	216.5672 s	209.0678 s	184.01 s	171.723 s
Proposed	192.8996 s	179.845 s	172.8436 s	171.667 s

6. Conclusions

This paper proposes a DOA estimation framework based on the DCGAN in underdetermined scenarios. Compared with most of the current DL-based methods, our proposed method transforms DOA estimation to a recovery task of a covariance matrix with more DOFs. Our method uses the DCGAN model and takes measures to preserve the Hermitian, Toeplitz and low-rank prior characteristics of the recovered covariance matrix. In underdetermined scenarios, the proposed method exhibits notable advantages in the fields of both accuracy and estimation speed, especially with limited snapshots. It is suitable for mobile agent localization.

Author Contributions: Conceptualization, Y.C. and F.Y.; methodology, Y.C. and F.Y.; software, F.Y. and A.K.; supervision, Y.C.; validation, Y.C., F.Y. and A.K.; formal analysis, J.W.; funding acquisition, Y.C., J.W. and H.S.; investigation, Y.C., F.Y., M.Z., J.W., H.S., A.K. and J.Q.; data curation, F.Y.; writing—original draft preparation, Y.C. and F.Y.; writing—review and editing, Y.C., F.Y., M.Z., L.H., J.W. and H.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported by National Natural Science Foundation of China (Grant No. 62201385), Technology Innovation Guidance Special Fund of Tianjin Science and Technology Plan Project (Grant No. 22YDTPJC00610), the Stable Supporting Fund of National Key Laboratory of Underwater Acoustic Technology (Grant No. JCKYS2023604SSJS008), the Key Laboratory of Southeast Coast Marine Information Intelligent Perception and Application, MNR, (Grant No. 220202) and the National Natural Science Foundation of China (Grant No. 62271426).

Data Availability Statement: Data are contained within the article.

Acknowledgments: We would like to express our gratitude to all members of the signal processing laboratory who have contributed to the experiment.

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

References

1. Florio, A.; Avitabile, G.; Talarico, C.; Coviello, G. A Reconfigurable Full-Digital Architecture for Angle of Arrival Estimation. *IEEE Trans. Circuits Syst. Regul. Pap.* 2023, *in press*. [CrossRef]
2. Schmidt, R. Multiple emitter location and signal parameter estimation. *IEEE Trans. Antennas Propag.* **1986**, *34*, 276–280. [CrossRef]
3. Roy, R.; Kailath, T. ESPRIT-estimation of signal parameters via rotational invariance techniques. *IEEE Trans. Acoust. Speech Signal Process.* **1989**, *37*, 984–995. [CrossRef]
4. Ye, Z.; Xu, X. DOA Estimation by Exploiting the Symmetric Configuration of Uniform Linear Array. *IEEE Trans. Antennas Propag.* **2007**, *55*, 3716–3720. [CrossRef]
5. Cui, Y.; Liu, K.; Wang, J. Direction-of-arrival estimation for coherent GPS signals based on oblique projection. *Signal Process.* **2012**, *92*, 294–299. [CrossRef]
6. El Zooghby, A.H.; Christodoulou, C.G.; Georgiopoulos, M. Performance of radial-basis function networks for direction of arrival estimation with antenna arrays. *IEEE Trans. Antennas Propag.* **1997**, *45*, 1611–1617. [CrossRef]
7. Shieh, C.; Lin, C. Direction of arrival estimation based on phase differences using neural fuzzy network. *IEEE Trans. Antennas Propag.* **2000**, *48*, 1115–1124. [CrossRef]
8. Liu, Z.-M.; Zhang, C.; Yu, P.S. Direction-of-Arrival Estimation Based on Deep Neural Networks with Robustness to Array Imperfections. *IEEE Trans. Antennas Propag.* **2018**, *66*, 7315–7327. [CrossRef]

9. Papageorgiou, G.K.; Sellathurai, M.; Eldar, Y.C. Deep Networks for Direction-of-Arrival Estimation in Low SNR. *IEEE Trans. Signal Process.* **2021**, *69*, 3714–3729. [CrossRef]
10. Wu, L.; Liu, Z.-M.; Huang, Z.-T. Deep Convolution Network for Direction of Arrival Estimation with Sparse Prior. *IEEE Signal Process. Lett.* **2019**, *26*, 1688–1692. [CrossRef]
11. Xiang, H.; Chen, B.; Yang, T.; Liu, D. Phase enhancement model based on supervised convolutional neural network for coherent DOA estimation. *Appl. Intell.* **2020**, *50*, 2411–2422. [CrossRef]
12. Lima de Oliveira, M.L.; Bekooij, M.J.G. ResNet Applied for a Single-Snapshot DOA Estimation. In Proceedings of the 2022 IEEE Radar Conference (RadarConf22), New York, NY, USA, 21–25 March 2022; pp. 1–6.
13. Peng, J.; Nie, W.; Li, T.; Xu, J. An end-to-end DOA estimation method based on deep learning for underwater acoustic array. In Proceedings of the OCEANS 2022, Hampton Roads, VA, USA, 17–20 October 2022; pp. 1–6.
14. Cao, Y.; Lv, T.; Lin, Z.; Huang, P.; Lin, F. Complex ResNet Aided DoA Estimation for Near-Field MIMO Systems. *IEEE Trans. Veh. Technol.* **2020**, *69*, 11139–11151. [CrossRef]
15. Zhao, F.; Hu, G.; Zhan, C.; Zhang, Y. DOA Estimation Method Based on Improved Deep Convolutional Neural Network. *Sensors* **2022**, *22*, 1305. [CrossRef]
16. Xiang, H.; Chen, B.; Yang, M.; Xu, S. Angle Separation Learning for Coherent DOA Estimation with Deep Sparse Prior. *IEEE Commun. Lett.* **2021**, *25*, 465–469. [CrossRef]
17. Fang, W.; Cao, Z.; Yu, D.; Wang, X.; Ma, Z.; Lan, B.; Xu, Z. A Lightweight Deep Learning-Based Algorithm for Array Imperfection Correction and DOA Estimation. *J. Commun. Inf. Netw.* **2022**, *7*, 296–308. [CrossRef]
18. Yao, Y.; Lei, H.; He, W. Wideband DOA Estimation Based on Deep Residual Learning with Lyapunov Stability Analysis. *IEEE Geosci. Remote. Sens. Lett.* **2022**, *19*, 8014505. [CrossRef]
19. Gao, S.; Ma, H.; Liu, H.; Yang, J.; Yang, Y. A Gridless DOA Estimation Method for Sparse Sensor Array. *Remote Sens.* **2023**, *15*, 5281. [CrossRef]
20. Wu, X.; Yang, X.; Jia, X.; Tian, F. A Gridless DOA Estimation Method Based on Convolutional Neural Network with Toeplitz Prior. *IEEE Signal Process. Lett.* **2022**, *29*, 1247–1251. [CrossRef]
21. Zhou, C.; Gu, Y.; Fan, X.; Shi, Z.; Mao, G.; Zhang, Y.D. Direction-of-Arrival Estimation for Coprime Array via Virtual Array Interpolation. *IEEE Trans. Signal Process.* **2018**, *66*, 5956–5971. [CrossRef]
22. Liu, L.; Rao, Z. An Adaptive Lp Norm Minimization Algorithm for Direction of Arrival Estimation. *Remote Sens.* **2022**, *14*, 766. [CrossRef]
23. Zhou, C.; Gu, Y.; Shi, Z.; Zhang, Y.D. Off-Grid Direction-of-Arrival Estimation Using Coprime Array Interpolation. *IEEE Signal Process. Lett.* **2018**, *25*, 1710–1714. [CrossRef]

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.