

## Article

# Direction-of-Arrival Estimation for a Random Sparse Linear Array Based on a Graph Neural Network

Yiye Yang <sup>1</sup>, Miao Zhang <sup>1,\*</sup>, Shihua Peng <sup>1</sup>, Mingkun Ye <sup>2</sup> and Yixiong Zhang <sup>1</sup>

<sup>1</sup> School of Electronic Science and Technology, Xiamen University, Xiamen 361005, China; 36620211150369@stu.xmu.edu.cn (Y.Y.); 36620221150427@stu.xmu.edu.cn (S.P.); zyx@xmu.edu.cn (Y.Z.)

<sup>2</sup> Tung Thih Electron Research Center, Xiamen 361021, China; yemkun0720@gmail.com

\* Correspondence: miao@xmu.edu.cn; Tel.: +86-592-2182456

**Abstract:** This article proposes a direction-of-arrival (DOA) estimation algorithm for a random sparse linear array based on a novel graph neural network (GNN). Unlike convolutional layers and fully connected layers, which do not interact well with information between different antennas, the GNN model can adapt to the goniometry problem of non-uniform random sparse linear arrays without any prior information by applying neighbor nodes' aggregation and update operations. This helps the model in learning signal features under complex environmental conditions. We train the model in an end-to-end way to reduce the complexity of the network. Experiments are conducted on the uniform and sparse linear arrays for various signal-to-noise ratio (SNR) and numbers of snapshots for comparison. We prove that the GNN model has superior angle estimation performance on arrays with large sparsity that cannot be used by traditional algorithms and surpasses existing deep learning models based on convolutional or fully connected structures. The proposed algorithm shows excellent DOA estimation performance under the complex conditions of limited snapshots, low signal-to-noise ratio, and large array sparsity as well. In addition, the algorithm has a low time calculation cost and is suitable for scenarios that require low latency.

**Keywords:** direction-of-arrival (DOA) estimation; sparse linear array; graph neural network; single snapshot; array signal processing



**Citation:** Yang, Y.; Zhang, M.; Peng, S.; Ye, M.; Zhang, Y. Direction-of-Arrival Estimation for a Random Sparse Linear Array Based on a Graph Neural Network. *Sensors* **2024**, *24*, 91. <https://doi.org/10.3390/s24010091>

Academic Editor: Xiaofei Zhang

Received: 16 October 2023

Revised: 6 December 2023

Accepted: 21 December 2023

Published: 23 December 2023



**Copyright:** © 2023 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

Direction of arrival (DOA), which realizes the spatial localization of signal sources by analyzing the signals received from array antennas, has always been a key technology in array signal processing. It has been widely applied in many scenarios, including airborne radar [1,2], underwater surveying [3,4], medical diagnostics [5], and autonomous driving [6,7]. The research challenges of DOA estimation are mainly focused on high-precision estimation under the conditions of limited snapshots, low signal-to-noise ratio (SNR), and sparse arrays.

Traditional DOA estimation algorithms can be divided into three categories. The first one comprises beamforming algorithms, such as digital beamforming (DBF) [8], minimum variance without distortion response (MVDR) [9], etc. The second one includes super-resolution goniometry algorithms based on subspace, such as estimating signal parameters via rotational invariance techniques (ESPRIT) [10], multi-signal classification (MUSIC) [11], and some of its derivatives [12–14]. The third one comprises sparse reconstruction algorithms, such as atomic norm minimization [15], sparse Bayesian learning (SBL) [16], compressed sensing (CS) [17,18] algorithms, etc. These algorithms rely on traditional array structures to achieve accurate DOA estimation with high SNR and many snapshots. However, non-ideal conditions such as limited snapshots and array defects lead to the degradation of traditional algorithm performance, and calculations such as feature value decomposition and spectral peak search are difficult to adapt to the real-time requirements of the algorithm in engineering.

In order to solve this problem, many scholars have introduced neural networks into the field of DOA estimation [19–23] and have achieved angle estimation by learning the nonlinear relationship between the output of the array and the position of the spatial signal source. Deep learning has been adopted for DOA estimation in large-scale MIMO arrays [24]. The literature shows that the uniform array goniometry algorithm based on a convolutional neural network has a significantly better DOA estimation accuracy than the traditional super-resolution estimation algorithm under the conditions of low SNR and a small number of snapshots. Furthermore, a DOA estimation algorithm based on a convolutional recurrent neural network (CRNN) [25] is proposed and can produce high-precision DOA estimation results under different SNR. The denoising autoencoder (DAE) neural network structure [26] is added to the literature, and the DOA detection accuracy is higher than that of the basic autoencoder by recovering the feature information. Recently, the literature has combined the sum-difference array with deep neural networks [21], demonstrating that neural networks can better adapt to array defects.

These studies show that DOA estimators based on neural networks have a faster computation speed and more robust estimation effects than traditional algorithms and can adapt to more complex environments and interference situations. However, most of them focus on uniform linear arrays with more than 200 snapshots and high SNR of 10 dB or more, which are barely implemented in reality. In addition, fewer DOA estimation studies on sparse arrays with random missing elements have been developed.

The CNN and MLP structure cannot make good information communication between antennas, and it is difficult to adapt to situations in which an array has large sparseness. However, a graph neural network (GNN) [27] is a neural network algorithm with nodes and adjacency matrices, which can well update the graph structure with information between neighbor nodes.

In this work, we propose a novel DOA estimation algorithm based on a graph neural network (GNN) within the GraphSAGE [28] structure to achieve high-precision DOA estimation in complex environments, such as those with low-SNR states, low numbers of snapshots, defective arrays or a combination of these situations. Since non-uniform random sparse arrays have missing elements, GNN can update the feature information of each element by aggregating the received signals between neighbor elements and can restore, to some extent, element information that is randomly missing in the non-uniform array. Hence, the proposed DOA estimation algorithm based on a GNN can achieve strong robustness.

The rest of this article is organized as follows: Section 2 describes the model of array signal and the formation and feature generation process for random sparse arrays. Furthermore, the same section proposes a DOA estimation framework based on a GNN and clarifies how the input and output meet the DOA estimation requirements, and it also introduces the training strategy of the neural network. Section 3 sets the simulation parameters and conducts experimental simulation to verify that the proposed algorithm is superior to previous algorithms based on neural networks and traditional algorithms adopted in random sparse arrays under the conditions of a small number of snapshots or even a single snapshot and low SNR. Section 4 concludes this paper.

*Notations:*  $(\cdot)^*$ ,  $(\cdot)^T$  and  $(\cdot)^H$  represent the conjugate operator, transpose operator and conjugate transpose operator of the matrix, respectively.  $E(\cdot)$  indicates the expected operator and  $\|\cdot\|_2$  represents the  $L_2$  norm operator of a matrix.  $Re(\cdot)$  and  $Im(\cdot)$  represent the real and imaginary parts of a complex value, and  $j = \sqrt{-1}$  represents the imaginary unit.  $\mathbb{C}$  and  $\mathbb{R}$  represent the fields of real numbers and complex numbers.

## 2. Materials and Methods

### 2.1. Signal Model

#### 2.1.1. Uniform Linear Array Signal

As an assumption,  $K$  independent far-field narrowband signals are incident on an  $M$ -element uniform linear array. As summarized in Figure 1, the uniform element spacing



where  $\mathbf{R}_{ss}$  represents the covariance matrix of signal  $S(t)$ ,  $\sigma^2$  is the noise power and  $\mathbf{I} \in \mathbb{R}^{M \times M}$  is the identity matrix.

### 2.1.2. Random Sparse Linear Array Signal

Random sparse linear arrays with different degrees of sparsity are also illustrated in Figure 1 for comparison. The aperture sizes of RSLAs are kept unchanged to match that of a ULA. Only one randomly generated sparse array for each sparsity rate is provided. The first and last elements of all arrays are reserved. A certain number of missing elements are randomly generated inside the array according to the specified sparsity rate. Hence, the received signals for the missing elements in the RSLA are zero.

As an assumption, the received signal matrix of the RSLA is denoted by  $\tilde{\mathbf{X}}(t)$ . The manifold matrix  $\tilde{\mathbf{A}}(\theta)$  of the RSLA can be generated from  $\mathbf{A}(\theta)$ , only by setting  $a(\theta_k)$  in  $\mathbf{A}(\theta)$  at 0 for the missing elements. Thus,  $\tilde{\mathbf{X}}(t)$  is expressed as follows:

$$\tilde{\mathbf{X}}(t) = \tilde{\mathbf{A}}(\theta)\mathbf{S}(t) + \mathbf{N}(t), \quad (8)$$

where  $\tilde{\mathbf{A}}(\theta) = [\tilde{a}(\theta_1), \tilde{a}(\theta_2), \dots, \tilde{a}(\theta_K)]^T$ , and  $\tilde{\mathbf{A}}(\theta)$  is a sparse matrix.

The covariance matrix  $\tilde{\mathbf{R}}_{xx} \in \mathbb{C}^{M \times M}$  of the random sparse array receiving signal  $\tilde{\mathbf{X}}(t)$  is as follows:

$$\tilde{\mathbf{R}}_{xx} = E\left\{\tilde{\mathbf{X}}(t)\tilde{\mathbf{X}}^H(t)\right\} = \tilde{\mathbf{A}}(\theta)\mathbf{R}_{ss}\tilde{\mathbf{A}}^H(\theta) + \sigma^2\mathbf{I}, \quad (9)$$

where  $\mathbf{R}_{ss}$  is the covariance matrix of signal  $S(t)$ ,  $\sigma^2$  is the noise power and  $\mathbf{I} \in \mathbb{R}^{M \times M}$  is the identity matrix.

### 2.1.3. Preprocessing of the Array Signal

The solution for the covariance matrix  $\mathbf{R}_{xx}$  and  $\tilde{\mathbf{R}}_{xx}$  is obtained under ideal conditions, but in a real system, the original covariance matrix is replaced by a sampling covariance matrix  $\mathbf{R} \in \mathbb{C}^{M \times M}$ . In order to facilitate the construction of mathematical models, the received signal  $\mathbf{X}(t)$  of a uniform linear array and the received signal  $\tilde{\mathbf{X}}(t)$  of a random sparse array are uniformly defined as  $\hat{\mathbf{X}}(t)$ , so it can be universally applied to the case of uniform arrays and sparse arrays. For the case where the received signal has multiple snapshots, the sampling covariance matrix expression of the array is:

$$\mathbf{R} = \frac{1}{T} \sum_{n=1}^T \hat{\mathbf{X}}(t_n)\hat{\mathbf{X}}^H(t_n), \quad (10)$$

where  $T$  is the number of snapshots, and  $\hat{\mathbf{X}}(t) = [\hat{x}_0(t), \hat{x}_1(t), \dots, \hat{x}_{M-1}(t)]^T$ .

Since the covariance matrix  $\mathbf{R}$  is a complex matrix, complex values cannot be directly used as input features of graph convolutional neural networks, so this work processes the covariance matrix as follows:

$$\bar{\mathbf{R}} = \begin{bmatrix} \text{Re} \left( \begin{pmatrix} R_{1,1} & R_{1,2} & \dots & R_{1,M} \\ R_{2,1} & R_{2,2} & \dots & R_{2,M} \\ \vdots & \vdots & \ddots & \vdots \\ R_{M,1} & R_{M,2} & \dots & R_{M,M} \end{pmatrix} \right), \text{Im} \left( \begin{pmatrix} R_{1,1} & R_{1,2} & \dots & R_{1,M} \\ R_{2,1} & R_{2,2} & \dots & R_{2,M} \\ \vdots & \vdots & \ddots & \vdots \\ R_{M,1} & R_{M,2} & \dots & R_{M,M} \end{pmatrix} \right) \end{bmatrix}, \quad (11)$$

where  $R_{i,j}$  ( $i = 1, 2, 3, \dots, M; j = 1, 2, 3, \dots, M$ ) represents the row- $i$  and column- $j$  element of the covariance matrix  $\mathbf{R}$ .

Normalizing the matrix  $\bar{\mathbf{R}}$  according to Equation (12) the matrix  $\hat{\mathbf{R}}$  can finally be used as the input feature of the neural network.

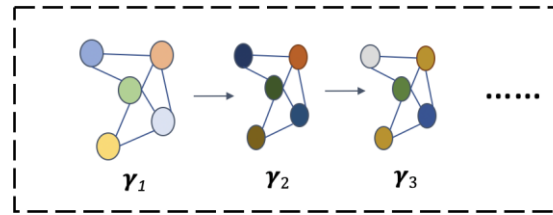
$$\hat{\mathbf{R}} = \bar{\mathbf{R}} / \|\bar{\mathbf{R}}\|_2, \quad (12)$$

where  $\|\cdot\|_2$  represents the  $L_2$  norm operator.

## 2.2. DOA Estimation Framework Based on GNN

### 2.2.1. Graph Convolutional Network

Graph convolutional neural networks have nodes and adjacency matrices. In this study, we introduce a residual graph convolutional neural network based on GraphSAGE to process sparse array signals. The process of working with GraphSAGE consists of two steps: aggregation and update. It is conducted to perform sampling and aggregation operations on the neighbor nodes of each node and obtain its own characteristic information by adding the current self-information and then updating it. We assume an undirected graph  $Z = (V, E)$  with  $N$  nodes, as shown in Figure 2. The features of different nodes, as well as their updated features, are distinguished by different colors.



**Figure 2.** Graph structure and its update process.

Here,  $V \in \mathbb{R}^{N \times F}$  is the node matrix in the graph.  $N$  represents the number of nodes, and  $F$  represents the dimension of each node feature in the graph.  $E \in \mathbb{R}^{N \times N}$  is the adjacency matrix and represents the connection relationship between nodes. If there is an edge between two nodes denoted by  $p$  and  $q$ ,  $E_{p,q} = 1$ ; otherwise, it is equal to 0. The set of nodes around a node with edges connected to it is called the set of neighbor nodes. When  $v$  is a node,  $\mathcal{N}(v)$  represents the set of neighbor nodes. The update process of GraphSAGE can be defined as follows [28]:

$$h_{\mathcal{N}(v)}^l = \text{AGGREGATE}_l \left( \left\{ h_u^{l-1}, \forall u \in \mathcal{N}(v) \right\} \right), \quad (13)$$

$$h_v^l = g \left( W^l \text{CONCAT} \left( h_v^{l-1}, h_{\mathcal{N}(v)}^l \right) \right), \quad (14)$$

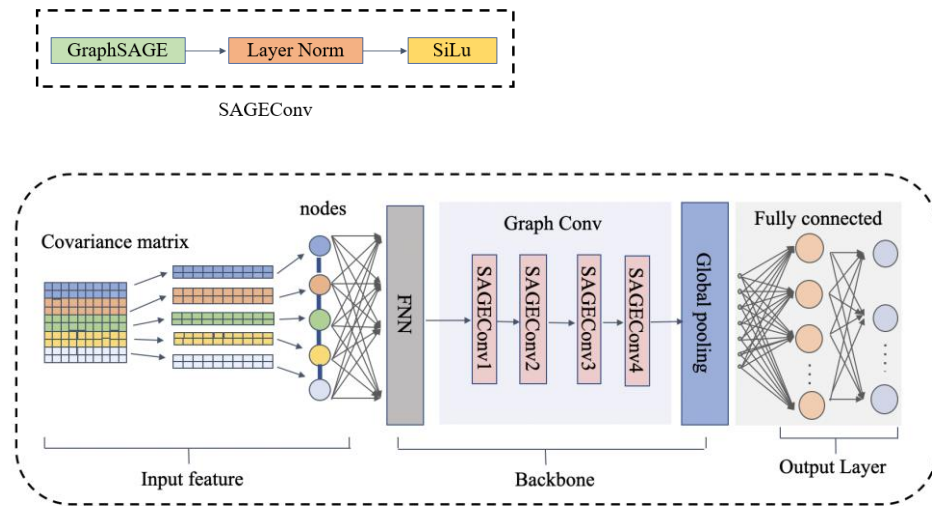
where  $h_{\mathcal{N}(v)}^l$  denotes the characteristics of neighbor nodes after aggregation.  $h_u^{l-1}$  and  $h_v^l$  represent the node feature information of layer  $l - 1$  and updated layer  $l$ , respectively. AGGREGATE is the aggregator using the averaging method for aggregation operations. CONCAT is the feature splicing operation, and  $g(\cdot)$  is an activation function.

For the non-uniform sparse array goniometry problem, the GraphSAGE convolutional layer updates the received signal characteristics of each group of elements by aggregating the information between different array groups. The addition of self-loop information can effectively fill the feature information at the vacant array elements and can update the features by aggregating the neighbor nodes and its own information. Therefore, the GNN has a certain degree of predictability for unknown information and can better adapt to the angle measurement situation under the conditions of random sparse arrays, low SNR, and small numbers of snapshots. Meanwhile, GNNs usually have strong generalization ability and can cope with radar goniometry tasks in different environments without the need for complex feature engineering in advance.

### 2.2.2. GNN-DOA Model for Random Sparse Arrays

The covariance matrix contains the angular information of the target and can be adopted as a feature of the input neural network. Since each covariance matrix corresponds to a set of angle estimation results, this study constructs the DOA estimation as a graph-level classification problem.

The DOA estimation model with a graph neural network (GNN) is shown in Figure 3. Each input covariance matrix is a graph and the covariance matrix is divided into a subset composed of multiple elements according to rows. Since the number of subsets equals the number of input nodes of the graph convolutional network, it is necessary to combine the received array signals in each subset into a set of vectors as the input features of the nodes. Finally, the adjacency matrix  $E$  is constructed according to the distance between each subset.



**Figure 3.** DOA estimation model with GNN.

The backbone network consists of a feedforward neural network (FNN), a Graph Conv module with four SAGEConv layers and a global pooling layer. The graph convolution structure used here is GraphSAGE. It adds a self-loop to the adjacency matrix, can effectively aggregate the information between nodes and updates the embedded expression of its own nodes.

When assuming  $z$  as the feature of input nodes, through two fully connected layers in FNN, the network output features are  $z_1$  and  $z_2$ . By introducing the weights of  $w_1$  and  $w_2$  and the biases of  $b_1$  and  $b_2$ , the update process can be expressed as follows:

$$z_1 = \text{SiLU}(w_1^T z + b_1), \quad (15)$$

$$z_2 = \text{SiLU}(w_2^T z_1 + b_2), \quad (16)$$

where

$$\text{SiLU}(a) = a\vartheta(a), \quad (17)$$

$$\vartheta(b) = \frac{1}{1 + e^{-b}}, \quad (18)$$

The activation function  $\text{SiLU}(\cdot)$  [32] nonlinearizes the output of the network.

Features enter the graph convolutional neural network for nodes' feature updates. As shown in Figure 3, the SAGEConv layer consists of the GraphSAGE, the Layer Norm (LN) and an activation layer. The LN calculates the mean and variance of a layer of inputs and normalizes the data, which can stabilize the forward input distribution and accelerate convergence on mini-batch inputs.

The update process of GraphSAGE is based on Equations (13) and (14). For simplicity, we define the Graph Conv module as a function  $G(\cdot)$ . Therefore, the update process for the Graph Conv module is:

$$z_3 = G(z_2), \quad (19)$$

After the node feature update is completed, the node features on each graph are mapped into a full graph representation through a global average pooling layer. The graph representation of the obtained feature  $z_3$  is  $\mathbf{H} = \{h_1, h_2, h_3, \dots, h_N\}$ , and the feature representations of each node at each layer are aggregated using the common summation method of global pooling to obtain the graph-level representation of the feature.

$$\mathcal{R}(\mathbf{H}) = \frac{1}{T} \sum_{i=1}^N h_i, \quad (20)$$

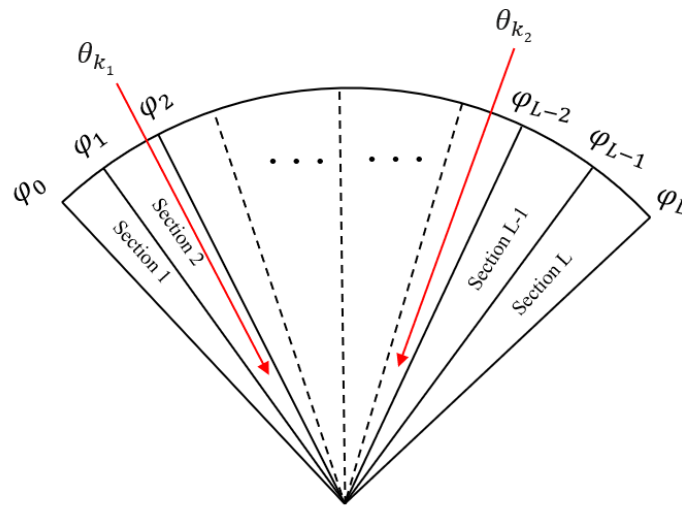
where  $\mathcal{R}(\cdot)$  is a commonly used readout function.

Finally, the graph-level features are classified through the fully connected layer. The network outputs the probability value of the presence of a target for each airspace interval, ranging from 0 to 1, and the closer the result is to 1, the greater the probability of the presence of a target in that direction.

### 2.2.3. Outputs of the GNN Model

Generally, since the number of measured target signals is limited and sparse in airspace, the target airspace range can be evenly divided into  $L$  intervals, as shown in Figure 4. We can obtain  $\varphi_0 < \varphi_1 < \dots < \varphi_L$ . If the target  $\theta_k$  ( $k = 1, 2, \dots, K$ ) exists in a sub-interval, the value of the interval section takes the value 1; otherwise, it takes a value of 0. The above concept can be described as the following formula:

$$f = \begin{cases} 1, & \varphi_{l-1} < \varphi_k < \varphi_l \\ 0, & \text{otherwise} \end{cases}, \forall k \in \{1, 2, \dots, K\}; l = 1, 2, \dots, L. \quad (21)$$



**Figure 4.** The relationship between target airspace and sections of GNN labels.

Therefore, the DOA estimation constructs a sparse mapping relationship from the covariance matrix to the target airspace.

The specific algorithm steps are summarized as follows: The first step is to preprocess the received signal. The second step is to build graphs and nodes to build a graph convolutional neural network model. The third step is to test various arrays with different structures under different conditions. The fourth step is to optimize the network structure and achieve the best DOA estimation algorithm.

### 3. Simulation and Discussion

Table 1 lists the parameters of the training dataset. A random sparse linear array of a 24-element aperture has a sparsity of 0.5. That means 12 elements are randomly missing

inside the array. Every six rows of the normalized covariance matrix are selected as a set of node features, and the number of nodes becomes four. To construct the adjacency matrix, we connect an edge between adjacent nodes with a distance of less than 2. In addition, the number of snapshots per sample is 50, and the SNR is randomly generated within the range of 0~10 dB. The discrete target space of  $-60^\circ \sim +60^\circ$  has a resolution of  $1^\circ$ , and 121 angle labels are obtained.

**Table 1.** Training parameters and values.

Training Parameters	Values
Number of array sensors	12
Sparsity of the array	0.5
Number of incoherent sources	2
Angular range	$[-60^\circ - 60^\circ]$
Number of sections	121
Angular resolution	$1^\circ$
SNR	0–10 dB
Number of snapshots	50

According to the above settings, the training dataset of two incoherent signals with 50,000 samples is generated. The training and testing datasets in the simulation are generated by MATLAB R2022a. The simulated samples are generated according to Equations (1)–(12), and the corresponding labels are generated according to Equation (21).

The batch size and the learning rate used for training are 32 and 0.001. All neural network-based models were trained for 500 epochs. The optimizer is Adam, and the activation function is SiLU. We used ZLPR Loss [33] as the loss function for adjusting and updating parameters during training.

ZLPR Loss compares all non-target class scores with the target class scores, in order to achieve the effect that the target class score is greater than the score of each non-target class. The specific formula is provided as follows:

$$Loss = \log\left(1 + \sum_{i \in \Omega_{neg}} e^{s_i}\right) + \log\left(1 + \sum_{i \in \Omega_{pos}} e^{-s_i}\right), \quad (22)$$

where  $\Omega_{neg}$  is the non-target class set,  $\Omega_{pos}$  is the target class set and  $S_i$  is the score.

When the result is greater than the threshold of 0.8, it is considered that there is a target in this direction, and the output result is set to 1; otherwise, it is set to 0. The accuracy calculation of the performance evaluation index is defined as follows: The total number of sample sets used for testing is recorded as *TOTAL*. If a sample is considered to be positive, the values of predictions and labels must be identical in all 121 dimensions. Meanwhile, the remaining samples are negative. The number of positive samples is recorded as *TRUE*, and the accuracy is calculated as follows:

$$Accuracy = \frac{TRUE}{TOTAL}, \quad (23)$$

In addition, mean absolute error (MAE) and root mean square error (RMSE) are two indicators used to evaluate DOA estimation error. MAE reflects the true error of DOA, while RMSE is more sensitive to outliers and can well reflect the robustness of DOA estimators. They are evaluated as follows:

$$MAE = \frac{1}{m} \sum_{i=1}^m |y_{true} - y_{pred}|, \quad (24)$$

$$RMSE = \sqrt{\frac{1}{m} \sum_{i=1}^m (y_{true} - y_{pred})^2}, \quad (25)$$

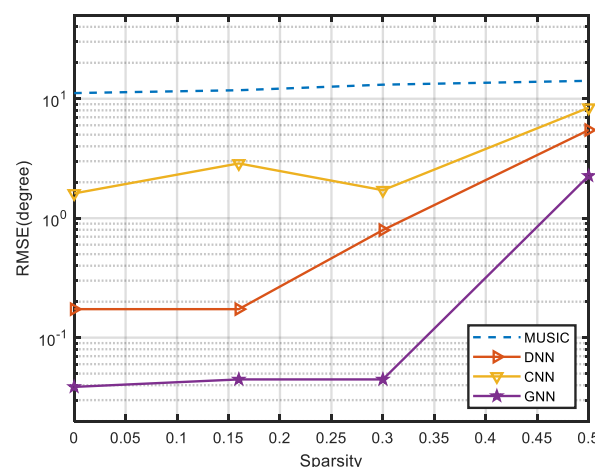
where  $m$  is the number of test samples.  $y_{true}$  and  $y_{pred}$  represent the true angle and pred angle, respectively.

We now consider three experiments for analysis. Firstly, the proposed GNN algorithm is tested to estimate the performance from the angles of different sparsity arrays and compared with other algorithms. Secondly, the goniometric performance of GNN and traditional algorithms under the condition of a single snapshot is tested. Thirdly, the performance of GNNs for sparse arrays is examined. Finally, we compare the goniometric times of different algorithms, applying an Intel® Core (TM) i7-9750Hv CPU @ 2.6 GHz hardware platform, based on CUDA 11.8 under the Pytorch framework.

### 3.1. Effect of the Array Sparsity on the Performance of DOA Estimation

In this experiment, to explore the performance of different angle estimation algorithms under different sparsity conditions, we select the SNR as 10 dB and the number of snapshots as 200. The traditional super-resolution algorithm MUSIC is chosen as a standard for comparison, while the existing neural network-based algorithm CNN and the DNN with MLP structure are compared.

First of all, the RMSE values of different algorithms under different sparsity conditions are tested. As shown in Figure 5, the proposed GNN algorithm has excellent performance, even when the array loses one-third of its elements. Its RMSE remains as low as  $0.04^\circ$ , which is much lower than  $13.12^\circ$  for MUSIC and  $0.79^\circ$  for DNN. Since MLP cannot exchange information between different array elements, under the condition of sparse arrays, a large amount of information can indeed lead to algorithm performance degradation. On the one hand, the CNN uses local perception to extract feature information, and the connection between global features is not tight. On the other hand, the graph convolution module transmits updates layer by layer through the connection relationship between nodes. It effectively captures and models the spatial relationship between sensors. More specifically, the GNN algorithm uses the signals received by the adjacent antennas to update the characteristics of the nodes. It obtains global information through the aggregation and update of each layer. So, the GNN algorithm can build a complex nonlinear mapping relationship even if there are a large number of elements lost.



**Figure 5.** RMSE of different algorithms under different conditions of sparsity.

Meanwhile, the accuracies of different algorithms under different sparsity conditions are tested. It can be seen from Table 2 that when the array loses one-half of its elements, the neural network-based algorithms CNN and DNN are significantly better than the traditional goniometric algorithm MUSIC. This is because neural network models are data-driven and can learn more robust nonlinear representations through backpropagation mechanisms to combat noise and fluctuations in complex environments. The proposed al-

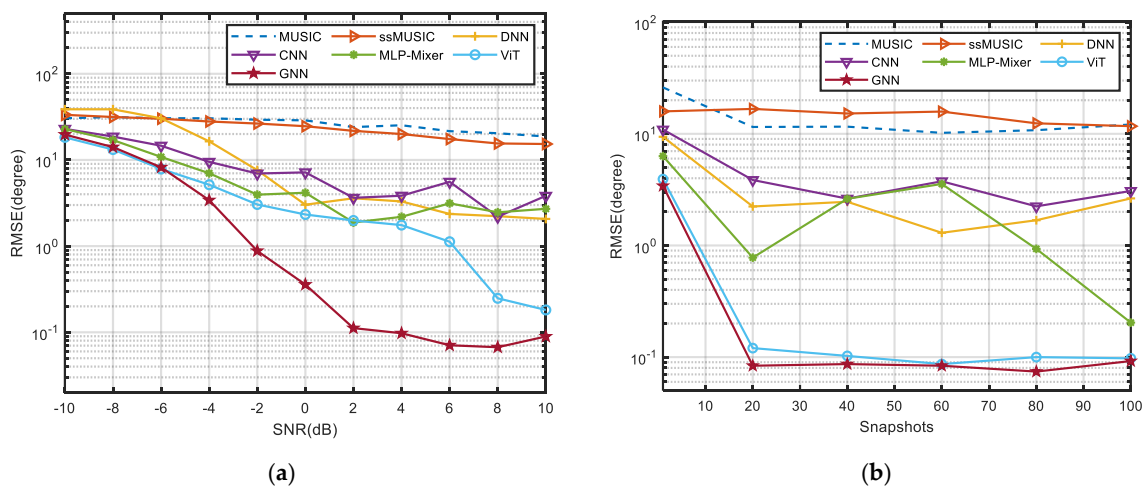
gorithm guarantees an accuracy of 95.3% in this case, which proves that the GNN algorithm can effectively cope with the situation of losing a large number of array elements.

**Table 2.** Accuracy of different algorithms under different sparsity conditions.

Number of Antennas	12	16	20	24
MUSIC	58.5%	68.8%	79.4%	83.6%
DNN	81.3%	92.9%	97.6%	98.4%
CNN	87.1%	97.1%	99.1%	99.0%
GNN	95.3%	99.3%	99.8%	99.8%

### 3.2. Effect of SNR and the Number of Snapshots on the Performance of DOA Estimation

In this experiment, under the conditions of an array sparsity of 1/3, the dependences of different algorithms on different SNR and numbers of snapshots are explored. We compare the GNN algorithm with MUSIC, the improved front-and-back smooth MUSIC (ssMUSIC) [13], the neural network-based algorithms DNN and CNN and two excellent neural network generalization models ViT [34] and MLP-Mixer [35]. It is observed in Figure 6a that under the condition of 50 snapshots, all algorithms improved with an increase in SNR, among which the algorithm based on the neural network improved significantly, and the GNN model achieved the best performance above  $-6$  dB. As shown in Figure 6b, under the condition of a signal-to-noise ratio of 10 dB, the number of snapshots is between 20 and 100, and the algorithm is not highly sensitive to its changes. This means that excellent angular performance can be achieved with a small number of snapshots. Even though MUSIC's performance is poor in low-snapshot conditions, both GNN and ViT guarantee a very low RMSE with similarity. At the same time, the parameter size of the ViT model reproduced in the experiment is 1.18 M, while the proposed GNN model parameter is 690.93 KB, indicating that the GNN algorithm achieves the same high-precision performance as ViT with less computing resources. It is obvious that the proposed GNN model can reduce the computational burden without sacrificing performance and the Graph Conv structure used in the GNN algorithm can well adapt to the angle measurement conditions of sparse arrays with limited snapshots.

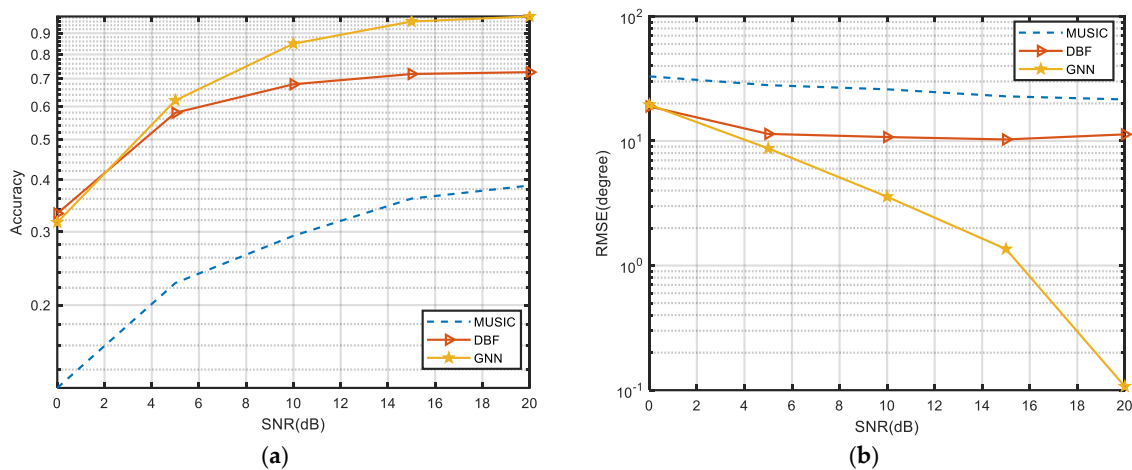


**Figure 6.** (a) RMSE of different algorithms with different SNR. (b) RMSE of different algorithms with different snapshots.

### 3.3. Effect of Single Snapshot and Large Array Sparseness on the Performance of DOA Estimation

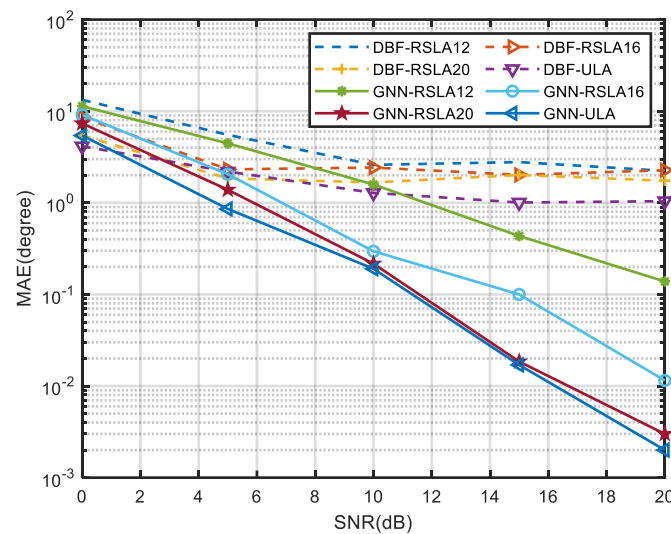
Single-snapshot angle estimation algorithms are commonly used in engineering, and their calculation speed can quickly meet the needs of low-latency scenarios such as automatic driving. In this experiment, with a sparsity of 1/3, a single snapshot is used for angle measurement to explore the performance of different algorithms. Since DBF [8] is

a very efficient single-snapshot angle estimation algorithm that can be implemented, the proposed algorithm is compared with it. As shown in Figure 7, MUSIC is difficult to cope with the situation of single-snapshot angle measurement when the array is sparse. With the improvement of SNR, MUSIC's performance is not significantly improved, because it is difficult to solve the feature vectors corresponding to the signal space and noise space from the covariance matrix of a single snapshot. Under the condition that the array has a large sparsity, the RMSE of the DBF algorithm remains high. However, The GNN algorithm proposed can adapt well to the situation of a single snapshot, and the RMSE is significantly reduced with the improvement of SNR. It should be noted that the GNN algorithm is very robust when the array is sparse.



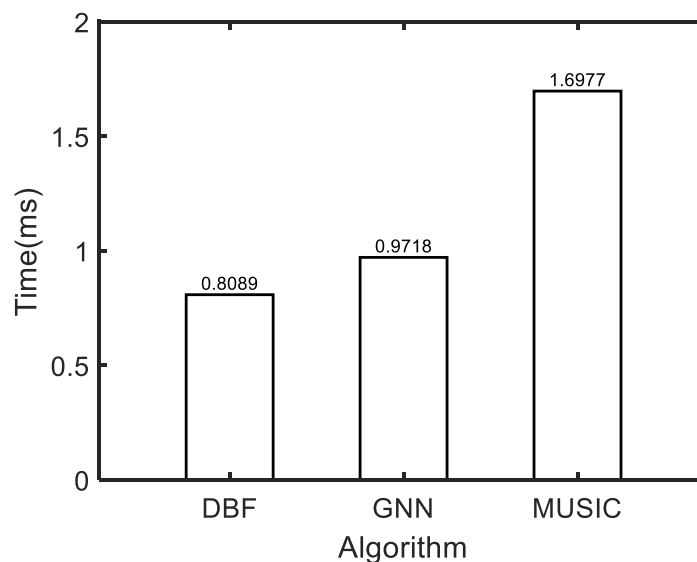
**Figure 7.** (a) RMSE of different algorithms under the single-snapshot condition. (b) Accuracy of different algorithms under the single-snapshot condition.

Furthermore, we tested the MAE of the single-snapshot algorithms on different sparsity arrays including RSLA12, RSLA16, RSLA20 and ULA. It can be seen from Figure 8 that when the SNR is higher than 5 dB, the GNN algorithm combined with RSLA16, RSLA20 and ULA achieves better angular measurement performance than the DBF algorithm applied to the ULA. Therefore, the proposed algorithm can adopt fewer antenna array elements occupying the same antenna aperture and obtain a level of performance only achieved by the traditional algorithm applied to the ULA. It also contributes to reducing the manufacturing cost of radar antennas. Especially in practical applications, a sparse array configuration, which can effectively reduce the coupling between physical antennas, has important engineering value.



**Figure 8.** MAE of DBF and GNN with different SNR on arrays of differing sparsity.

Finally, we compare the traditional DOA estimation algorithms with the proposed algorithm based on a graph neural network. The test experiment calculates the average of 1000 Monte Carlo experiments on an Intel® Core (TM) i7-9750Hv CPU @ 2.6GHz hardware platform. It can be found in Figure 9 that the time calculation cost of the GNN algorithm is nearly half that of MUSIC and is slightly higher than that of the DBF algorithm. It is obvious that the proposed algorithm can achieve better angular measurement performance while ensuring low computational cost and is more in line with the real-time requirements of engineering angle estimation algorithms.



**Figure 9.** Comparison of running time of different DOA estimation algorithms.

#### 4. Conclusions

This work proposes a GNN-based DOA estimation algorithm. The graph convolutional network structure is introduced into the field of DOA estimation for achieving robust angle estimation on non-uniform random sparse arrays. The GNN model can effectively fill in the information at the missing elements by aggregating and updating the characteristics of neighbor nodes in the convolutional structure and provides good robustness in the angle estimation problem of sparse arrays. In the experiments, the proposed algorithm achieves better goniometric performance, compared with the traditional algorithm and the

existing deep neural network algorithm based on multilayer perceptron and convolutional structure in a complex environment. We tested the proposed algorithm on an array with a sparsity of 1/3. When the SNR is  $-4$  dB and the number of snapshots is 50, the angle measurement performance is 88.81% higher than that of MUSIC. In the case of limited snapshots or even a single snapshot, the proposed GNN algorithm has no dependence on the specific array arrangement and is well adapted to arrays with a large number of random missing elements. In addition, the array sparseness can effectively reduce the manufacturing cost of antennas and reduce the coupling between elements. Meanwhile, the GNN algorithm has a low time calculation cost and can meet the requirements of scenarios requiring low-latency angle estimation. This proves the advanced and efficient introduction of graph convolutional networks into DOA estimation problems.

In the future, we will further investigate the DOA of sparse arrays for incoherent sources, study the DOA of virtual sparse arrays and improve the robustness of the model. In addition, experimental verification by measured data will be conducted soon.

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

**Funding:** This research was funded by the National Natural Science Foundation of China under Grant 61971364.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data presented in this study are available on request from the corresponding author. The training dataset used in this article is obtained through simulation and is not a public dataset.

**Acknowledgments:** The authors thank Liye Xiao and Qiqiang Zou for their contributions to this research.

**Conflicts of Interest:** The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

## References

1. Chen, Z.; Cao, Z.; He, X.; Jin, Y.; Li, J.; Chen, P. DoA and DoD Estimation and Hybrid Beamforming for Radar-Aided mmWave MIMO Vehicular Communication Systems. *Electronics* **2018**, *7*, 40. [\[CrossRef\]](#)
2. Ma, J.; Zhang, J.; Yang, Z.; Qiu, T. Off-Grid DOA Estimation Using Sparse Bayesian Learning for MIMO Radar under Impulsive Noise. *Sensors* **2022**, *22*, 6268. [\[CrossRef\]](#)
3. Niu, H.; Reeves, E.; Gerstoft, P. Source localization in an ocean waveguide using supervised machine learning. *J. Acoust. Soc. Am.* **2017**, *142*, 1176–1188. [\[CrossRef\]](#)
4. Ning, G.; Wang, Y.; Jing, G.; Zhao, X. A Velocity-independent DOA Estimator of Underwater Acoustic Signals via an Arbitrary Cross-linear Nested Array. *Circuits Syst. Signal Process.* **2022**, *42*, 996–1010. [\[CrossRef\]](#)
5. Shu Ting, G.; Reza Zekavat, S.A.; Pahlavan, K. DOA-Based Endoscopy Capsule Localization and Orientation Estimation via Unscented Kalman Filter. *IEEE Sens. J.* **2014**, *14*, 3819–3829. [\[CrossRef\]](#)
6. 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.* **2021**, *22*, 4301–4315. [\[CrossRef\]](#)
7. Xu, H.; Liu, W.; Jin, M.; Tian, Y. Positioning and Contour Extraction of Autonomous Vehicles Based on Enhanced DOA Estimation by Large-Scale Arrays. *IEEE Internet Things J.* **2023**, *10*, 11792–11803. [\[CrossRef\]](#)
8. Van Veen, B.D.; Buckley, K.M. Beamforming: A versatile approach to spatial filtering. *IEEE ASSP Mag.* **1988**, *5*, 4–24. [\[CrossRef\]](#)
9. Capon, J. High-Resolution Frequency-Wavenumber Spectrum Analysis. *Proc. IEEE* **1969**, *57*, 1408–1418. [\[CrossRef\]](#)
10. Roy, R.; Kailath, T. Estimation of Signal Parameters via Rotational Invariance Techniques—ESPRIT. In Proceedings of the MILCOM 1986—IEEE Military Communications Conference: Communications-Computers: Teamed for the 90's, Monterey, CA, USA, 5–9 October 1986; Volume 3, pp. 41.46.41–41.46.45.
11. Schmidt, R. Multiple emitter location and signal parameter estimation. *IEEE Trans. Antennas Propag.* **1986**, *34*, 276–280. [\[CrossRef\]](#)

12. Sen Li, B.L. On Spatial Smoothing for Direction-of-Arrival Estimation of Coherent Signals in Impulsive Noise. In Proceedings of the 2015 IEEE Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chongqing, China, 19–20 December 2015.
13. Pillai, S.U.; Kwon, B.H. Forward/backward spatial smoothing techniques for coherent signal identification. *IEEE Trans. Acoust. Speech Signal Process.* **1989**, *37*, 8–15. [[CrossRef](#)]
14. Choi, J.; Park, J.; Yeom, D. High Angular Resolution Estimation methods For Vehicle FMCW Radar. In Proceedings of the Proceedings of 2011 IEEE CIE International Conference on Radar, Chengdu, China, 24–27 October 2011.
15. Tropp, J.A.; Wright, S.J. Computational Methods for Sparse Solution of Linear Inverse Problems. *Proc. IEEE* **2010**, *98*, 948–958. [[CrossRef](#)]
16. Wang, H.; Wang, X.; Huang, M.; Wan, L. A novel variational SBL approach for off-grid DOA detection under nonuniform noise. *Digit. Signal Process.* **2022**, *128*, 103622. [[CrossRef](#)]
17. Yong, Z.; Li-Yi, Z.; Jian-Feng, H.; Zhe, B.; Yi, Y. A new DOA estimation algorithm based on compressed sensing. *Clust. Comput.* **2018**, *22*, 895–903. [[CrossRef](#)]
18. Mohimani, H.; Babaie-Zadeh, M.; Jutten, C. A Fast Approach for Overcomplete Sparse Decomposition Based on Smoothed  $\ell^0$  Norm. *IEEE Trans. Signal Process.* **2009**, *57*, 289–301. [[CrossRef](#)]
19. Zhao, F.; Hu, G.; Zhan, C.; Zhang, Y. DOA Estimation Method Based on Improved Deep Convolutional Neural Network. *Sensors* **2022**, *22*, 1305. [[CrossRef](#)] [[PubMed](#)]
20. Liu, W. Super resolution DOA estimation based on deep neural network. *Sci. Rep.* **2020**, *10*, 19859. [[CrossRef](#)] [[PubMed](#)]
21. Xu, S.; Chen, B.; Xiang, H. A low computational complexity DOA estimation using sum/difference pattern based on DNN. *Multidimens. Syst. Signal Process.* **2022**, *34*, 205–225. [[CrossRef](#)]
22. Zhang, M.; Wang, C.; Zhu, W.; Shen, Y. An Intelligent DOA Estimation Error Calibration Method Based on Transfer Learning. *Appl. Sci.* **2022**, *12*, 7636. [[CrossRef](#)]
23. Yang, T.; Wang, L.; Li, X. DOA Estimation based on Deconvolution Beamforming-based Unrolling Neural Network. In Proceedings of the 2022 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA), Dalian, China, 24–26 June 2022; pp. 305–310.
24. Huang, H.; Yang, J.; Huang, H.; Song, Y.; Gui, G. Deep Learning for Super-Resolution Channel Estimation and DOA Estimation Based Massive MIMO System. *IEEE Trans. Veh. Technol.* **2018**, *67*, 8549–8560. [[CrossRef](#)]
25. Li, X.; Chen, J.; Bai, J.; Ayub, M.S.; Zhang, D.; Wang, M.; Yan, Q. Deep learning-based DOA estimation using CRNN for underwater acoustic arrays. *Front. Mar. Sci.* **2022**, *9*, 1027830. [[CrossRef](#)]
26. Chen, D.; Shi, S.; Gu, X.; Shim, B. Robust DoA Estimation Using Denoising Autoencoder and Deep Neural Networks. *IEEE Access* **2022**, *10*, 52551–52564. [[CrossRef](#)]
27. Scarselli, F.; Gori, M.; Tsoi, A.C.; Hagenbuchner, M.; Monfardini, G. The graph neural network model. *IEEE Trans. Neural. Netw.* **2009**, *20*, 61–80. [[CrossRef](#)] [[PubMed](#)]
28. William, L.; Hamilton, R.Y. Jure Leskovec. Inductive Representation Learning on Large Graphs. *INPS* **2017**, arXiv:1706.02216.
29. Hamid Krlm, M.V. Two decades of array signal processing research the parametric approach. *IEEE Signal Process. Mag.* **1996**, *13*, 67–94.
30. Stoica, A.N.P. MUSIC, Maximum Likelihood, and Cramer-Rao Bound. *IEEE Trans. Acoust. Speech Signal Process* **1989**, *37*, 720–741. [[CrossRef](#)]
31. Stoica, A.N.P. Performance study of conditional and unconditional direction-of-arrival estimation. *IEEE Trans. Acoust. Speech Sig. Process* **1990**, *38*, 1783–1795. [[CrossRef](#)]
32. Elfving, S.; Uchibe, E.; Doya, K. Sigmoid-Weighted Linear Units for Neural Network Function Approximation in Reinforcement Learning. *Neural Netw. Off. J. Int. Neural Netw. Soc.* **2017**, *107*, 3–11. [[CrossRef](#)]
33. Su, J.; Zhu, M.; Murtadha, A.; Pan, S.; Wen, B.; Liu, Y. ZLPR: A Novel Loss for Multi-label Classification. *arXiv* **2022**, arXiv:2208.02955.
34. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv* **2020**, arXiv:2010.11929.
35. Tolstikhin, I.O.; Houlsby, N.; Kolesnikov, A.; Beyer, L.; Zhai, X.; Unterthiner, T.; Yung, J.; Steiner, A.; Keysers, D.; Uszkoreit, J.; et al. MLP-Mixer: An all-MLP Architecture for Vision. *Neural Inf. Process. Syst.* **2021**, *34*, 24261–24272.

**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.