

Article

Two-Dimensional DOA Estimation Based on Dual-Branch CNN

Fangyu Liu ^{1,2}, Guimei Zheng ^{1,*}, Yuwei Song ¹, Yujie Bai ^{1,2} and He Zheng ¹

¹ Air and Missile Defense College, Air Force Engineering University, Xi'an 710051, China; fangyu020513@163.com (F.L.); songyuwei2015@163.com (Y.S.); baiyujie19971210@163.com (Y.B.); 14160111033@stu.xidian.edu.cn (H.Z.)

² Graduate School, Air Force Engineering University, Xi'an 710051, China

* Correspondence: zheng-gm@163.com; Tel.: +86-17791976171

Abstract

Direction-of-arrival (DOA) estimation is a core research topic in array signal processing, and two-dimensional (2-D) DOA estimation can jointly acquire the azimuth and elevation of incoming signals, which bears great practical value. Traditional subspace and sparse reconstruction algorithms are plagued by heavy computation and deteriorated accuracy under imperfect array manifolds, low signal-to-noise ratios (SNRs) and insufficient snapshots. To enhance estimation robustness and inference speed simultaneously, this paper presents a dual-branch convolutional neural network (CNN) for 2-D DOA estimation based on uniform rectangular arrays. The network takes the sample covariance matrix of array received data as input. A shared feature encoder with residual blocks and channel-attention modules extracts common spatial features, followed by two prediction heads with independent parameters for elevation and azimuth estimation. Because each branch has a 61-dimensional output while two sources may be simultaneously present, the angle estimation is formulated as multi-label classification using sigmoid outputs and weighted binary cross-entropy. Simulations covering diverse SNRs, snapshot counts, angular intervals and off-grid cases verify that the proposed network obtains smaller root mean square errors than methods with multiple signal classification (MUSIC), estimation of signal parameters via rotational invariance techniques (ESPRIT) and ordinary CNN methods, with millisecond-level inference latency. This framework offers an efficient, high-precision real-time 2-D DOA estimation scheme for complicated electromagnetic scenes.

Keywords: direction-of-arrival estimation; convolutional neural networks; array signal processing; robust estimation



Academic Editor: Costas Psychalinos

Received: 5 July 2026

Revised: 4 August 2026

Accepted: 4 August 2026

Published: 6 August 2026

Copyright: © 2026 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 \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Direction-of-arrival (DOA) estimation is a fundamental task in array signal processing and provides spatial information for radar detection and tracking, sonar and underwater acoustic localization, wireless positioning, adaptive beamforming, cognitive radio, and unmanned-platform perception [1–4]. Two-dimensional (2-D) DOA estimation is particularly important because it jointly recovers the elevation and azimuth of each incident source. Unlike one-dimensional estimation, however, the unknown parameter is not merely two independent angle sets but a collection of paired angle tuples, (θ_k, ϕ_k) , associated with individual sources. A practical 2-D estimator must therefore extract the two angular components while preserving their source-wise correspondence. This requirement makes search dimensionality, inter-angle coupling, and multi-source pairing central issues in planar-, coprime-planar-, and metasurface-array applications [5,6]. Real-time angle-of-arrival (AoA)

estimates can also serve as feedback variables for adaptive beam steering. For example, a direct-digital-synthesis/phase-locked-loop (DDS-PLL) architecture has been used to convert real-time AoA estimates into local-oscillator phase shifts for closed-loop beam steering [7]. Hardware-aware angle estimation provides a complementary route to low latency: a recent field-programmable-gate-array (FPGA) system replaces a full-resolution analog-to-digital converter (ADC) with a 1-bit interface to reduce digital complexity and enable real-time AoA estimation [8]. These hardware-oriented studies are not direct algorithmic baselines for the present covariance-based 2-D estimator, but they motivate transparent reporting of online latency and implementation conditions.

Classical 2-D DOA estimators are mainly built on subspace principles. Two-dimensional MUSIC exploits the orthogonality between the signal and noise sub-spaces and can provide high resolution under accurate array calibration and sufficient snapshots [9–11]. Its practical cost, however, is dominated by eigen decomposition and a two-dimensional spectral search over the elevation–azimuth plane. ESPRIT and its multidimensional variants avoid exhaustive peak searching by exploiting rotational invariance [12–15], while propagator-based methods further reduce the cost of subspace decomposition [16]. These advantages depend on array structures that preserve the required invariance relations and on reliable subspace estimation. Mutual coupling, gain–phase errors, sensor-position perturbations, coherent sources, low signal-to-noise ratio (SNR), and limited snapshots can distort the array manifold or sample covariance matrix and substantially degrade the resulting 2-D estimates.

Sparse-reconstruction methods provide a second major route to high-resolution 2-D estimation. By representing the spatial field with an angular dictionary, compressive-sensing approaches can recover multiple sources from limited observations and can improve resolution in difficult angular configurations [17–20]. Nevertheless, a two-dimensional dictionary grows approximately with the product of the azimuth and elevation grid sizes, and iterative sparse recovery can become computationally expensive [21,22]. Moreover, grid-based sparse estimators are affected by dictionary mismatch when the true angles do not coincide with the predefined grid. Off-grid sparse Bayesian inference [23] and recent gridless planar-array methods alleviate this limitation through continuous-parameter or structured-covariance formulations, but generally require additional iterative optimization. Thus, grid quantization is a general limitation of grid-based estimators; in 2-D problems it is amplified by the quadratic growth of the angular search space [24,25].

These limitations have motivated data-driven DOA estimators that learn a mapping from array observations or sample covariance matrices to source angles. With the development of deep learning [26,27], Early support-vector and deep-neural-network approaches demonstrated that online spectral searching can be replaced by learned inference and that training over array perturbations can improve robustness to model mismatch [28–30]. Subsequent studies introduced DeepMUSIC, complex-valued CNNs, graph neural networks, Transformer architectures, and hybrid model-driven/data-driven estimators [31–35]. Most of these methods, however, address one-dimensional DOA estimation or learn an intermediate spatial spectrum. The literature devoted specifically to multi-source 2-D estimation is comparatively limited. Representative designs include a Khatri–Rao-product and sparse-prior CNN estimator [36], an L-shaped-array framework that estimates marginal angles and then performs pairing with a separate residual network [37], a high-resolution multi-source CNN developed for imperfect arrays [38], and a tensorized neural network that preserves the multidimensional structure of uniform rectangular array (URA) statistics while reducing the number of trainable parameters [39].

The above studies reveal two 2-D-specific design questions. First, elevation and azimuth are generated by the same received field and therefore share informative covariance

features, yet their decision boundaries and error characteristics are not identical. A fully joint prediction head may introduce optimization interference between the two angular tasks, whereas two completely separate networks duplicate feature extraction and may underuse their common spatial structure. Second, once the marginal elevation and azimuth sets are estimated separately, their source-wise association is no longer guaranteed, and an explicit pairing mechanism is required. These two issues constitute the main focus of this work. By contrast, the fixed-grid quantization problem is treated as a limitation of the adopted classification formulation rather than as a problem solved by the proposed architecture. Accordingly, the off-grid experiment in this paper is used only to characterize sensitivity to grid mismatch and not to claim continuous off-grid estimation.

To address the shared-versus-specific representation trade-off and the pairing requirement, this paper develops a dual-branch CNN for 2-D DOA estimation with a URA. The real and imaginary components of the sample covariance matrix are used as network inputs. A shared encoder composed of residual blocks and channel-attention modules extracts common spatial features, after which two prediction heads with independent parameters learn elevation-specific and azimuth-specific representations. In addition to the angle-classification terms, a contrastive pairing term is introduced to encourage features belonging to the elevation and azimuth of the same source to be more consistent than features belonging to different sources. The proposed design therefore separates the two angular decisions without discarding the common information contained in the array covariance matrix.

The main contributions of this paper are summarized as follows.

- (1) A shared-encoder/dual-head architecture is formulated for URA-based 2-D DOA estimation. It reuses common spatial features while allowing the elevation and azimuth branches to learn task-specific decision representations. This architecture is motivated by the trade-off between fully joint task-specific parameters, which may introduce cross-task optimization interference, and two completely independent networks, which duplicate feature extraction.
- (2) The 61-dimensional outputs are reformulated consistently as multi-label angle-occupancy predictions using independent sigmoid activations and weighted binary cross-entropy. The resulting unordered elevation and azimuth candidates are associated using a model-based covariance-fitting criterion, so that multi-source pairing does not rely on angle sorting, confidence ordering, or an unreported source-index convention.
- (3) The proposed method is evaluated in terms of estimation accuracy and online computational cost under different SNRs, snapshot numbers, and angular configurations. Grid-mismatched samples are additionally used to quantify the classification model's sensitivity to angular quantization; they are not presented as evidence that the method solves the continuous off-grid estimation problem.

The remainder of this paper is organized as follows. Section 2 introduces the URA signal model, statistical assumptions, SNR definition, and covariance-matrix representation. Section 3 describes the shared encoder, dual prediction heads, multi-label loss, and model-based angle-pairing procedure. Section 4 presents the simulation settings and performance evaluation. Section 5 concludes the paper and discusses current limitations and future work.

2. Signal Model

The performance of 2-D DOA estimation is closely related to the structure of the receiving array. The Uniform Rectangular Array (URA) has become the preferred array structure in 2-D angle estimation scenarios due to its advantages of balanced spatial

resolution, strong decoupling ability between azimuth and elevation angles, and regular array element layout [26]. Therefore, this paper adopts URA as the receiving array. The planar array has $M \times N$ array elements uniformly distributed on the x-y plane, with an interval of d between adjacent array elements. To avoid grating lobes, in this article $d = \lambda/2$. Assume there are K far-field narrowband signal sources in space, then the 2-D DOA of the k -th signal source is (θ_k, ϕ_k) , $k = 1, 2, \dots, K$, where θ_k represents the elevation angle of the k -th signal source, and ϕ_k represents the azimuth angle of the k -th signal source. Consistent with the steering-vector expressions used below, θ_k is defined as the elevation-related polar angle measured from the positive z-axis, and ϕ_k is the azimuth measured counterclockwise from the positive x-axis in the x-y plane. The specific situation is shown in Figure 1.

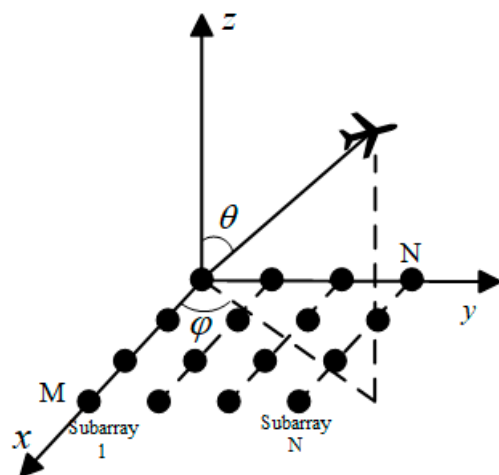


Figure 1. Uniform rectangular array.

The steering vector of the signal source on the x-axis can be expressed as follows:

$$\mathbf{a}_x(\theta_k, \phi_k) = \begin{bmatrix} 1 \\ e^{j2\pi d \sin \theta_k \cos \phi_k / \lambda} \\ \vdots \\ e^{j2\pi (M-1) d \sin \theta_k \cos \phi_k / \lambda} \end{bmatrix} \quad (1)$$

Thus, the direction matrix corresponding to the M array elements on the x-axis is $\mathbf{A}_x = [\mathbf{a}_x(\theta_1, \phi_1), \dots, \mathbf{a}_x(\theta_K, \phi_K)]$.

Similarly, the steering vector of the signal source on the y-axis can be expressed as follows:

$$\mathbf{a}_y(\theta_k, \phi_k) = \begin{bmatrix} 1 \\ e^{j2\pi d \sin \theta_k \sin \phi_k / \lambda} \\ \vdots \\ e^{j2\pi (N-1) d \sin \theta_k \sin \phi_k / \lambda} \end{bmatrix} \quad (2)$$

The direction matrix corresponding to the N array elements on the y-axis is $\mathbf{A}_y = [\mathbf{a}_y(\theta_1, \phi_1), \dots, \mathbf{a}_y(\theta_K, \phi_K)]$.

The signal received by Subarray 1 in the rectangular array shown in Figure 1 can be expressed as follows:

$$\mathbf{x}_1(t) = \mathbf{A}_x \mathbf{s}(t) + \mathbf{n}_1(t) \quad (3)$$

where $\mathbf{A}_x$ represents the direction matrix in Subarray 1, $\mathbf{s}(t)$ is the signal source vector, and $\mathbf{n}_1(t)$ represents the additive white Gaussian noise in Subarray 1.

The signal received by the m -th subarray in the planar array can also be expressed through the direction matrix corresponding to Subarray 1 as follows:

$$\mathbf{x}_n(t) = \mathbf{A}_x \Phi_y^{n-1} \mathbf{s}(t) + \mathbf{n}_n(t) \quad (4)$$

where $\Phi_y = \text{diag}\left(e^{j2\pi d \sin \theta_1 \sin \phi_1 / \lambda}, \dots, e^{j2\pi d \sin \theta_k \sin \phi_k / \lambda}\right)$, and $\mathbf{n}_n(t)$ represents the additive white Gaussian noise of the m -th subarray. Thus, the signal received by the entire uniform rectangular array can be obtained as follows:

$$\mathbf{x}(t) = \begin{bmatrix} \mathbf{x}_1(t) \\ \mathbf{x}_2(t) \\ \vdots \\ \mathbf{x}_N(t) \end{bmatrix} = \begin{bmatrix} \mathbf{A}_x \\ \mathbf{A}_x \Phi_y \\ \vdots \\ \mathbf{A}_x \Phi_y^{N-1} \end{bmatrix} \mathbf{s}(t) + \begin{bmatrix} \mathbf{n}_1(t) \\ \mathbf{n}_2(t) \\ \vdots \\ \mathbf{n}_N(t) \end{bmatrix} \quad (5)$$

The signal corresponding to (3) can also be expressed as follows:

$$\mathbf{x}(t) = [\mathbf{A}_y \odot \mathbf{A}_x] \mathbf{s}(t) + \mathbf{n}(t) \quad (6)$$

where $\odot$ denotes the Khatri–Rao product, and $\mathbf{A}_y$ is the direction matrix corresponding to the N array on the y -axis. According to the definition of the Khatri–Rao product, the received signal can be expressed as follows:

$$\mathbf{x}(t) = [\mathbf{a}_y(\theta_1, \phi_1) \otimes \mathbf{a}_x(\theta_1, \phi_1), \dots, \mathbf{a}_y(\theta_k, \phi_k) \otimes \mathbf{a}_x(\theta_k, \phi_k)] \mathbf{s}(t) + \mathbf{n}(t) \quad (7)$$

where $\otimes$ denotes the Kronecker product. Assume the signal is sampled L times, the steering vectors on the x -axis and y -axis are fixed. Define $\mathbf{X} = [\mathbf{x}(1), \mathbf{x}(2), \dots, \mathbf{x}(L)]$, then the received signal after L samplings can be expressed as follows:

$$\mathbf{X} = [\mathbf{A}_y \odot \mathbf{A}_x] \mathbf{S}^T + \mathbf{N} = \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \\ \vdots \\ \mathbf{X}_N \end{bmatrix} = \begin{bmatrix} \mathbf{A}_x \mathbf{D}_1(\mathbf{A}_y) \\ \mathbf{A}_x \mathbf{D}_2(\mathbf{A}_y) \\ \vdots \\ \mathbf{A}_x \mathbf{D}_N(\mathbf{A}_y) \end{bmatrix} \mathbf{S}^T + \begin{bmatrix} \mathbf{N}_1 \\ \mathbf{N}_2 \\ \vdots \\ \mathbf{N}_N \end{bmatrix} \quad (8)$$

where $\mathbf{S} = [\mathbf{s}(1), \mathbf{s}(2), \dots, \mathbf{s}(L)]^T$ represents the signal source vector after L samplings, $\mathbf{S} \in \mathbb{C}^{L \times K}$. $\mathbf{D}_m(\cdot)$ represents a diagonal matrix formed from the m -th row of the matrix. $\mathbf{N} = [\mathbf{n}(1), \mathbf{n}(2), \dots, \mathbf{n}(L)]$ is the additive white Gaussian noise matrix in the received signal. Each received-data and noise vector has MN complex entries, each source vector has K complex entries, and the snapshot matrices $\mathbf{X}$, $\mathbf{N}$, and $\mathbf{S}$ contain L columns. The number of sources K is assumed known and fixed; $K = 2$ is used in the current experiments. Unless a robustness experiment states otherwise, the sources are zero-mean, circularly symmetric complex Gaussian, mutually independent, temporally independent, noncoherent, and equal in power.

In classical DOA estimation algorithms, the array covariance matrix is mainly used for estimation. The covariance matrix can be obtained from (7) as follows:

$$\hat{\mathbf{R}} = \mathbf{X} \mathbf{X}^H / L \quad (9)$$

In the deep learning network proposed in this paper, the array covariance matrix is also used as the input of the network.

The SNR is defined as the average received signal power per sensor divided by the noise variance. In decibels, the convention used in this manuscript is

$$\text{SNR} = 10 \log_{10} \left(\frac{\text{tr}(\mathbf{A}\mathbf{R}_s\mathbf{A}^H)}{MN\sigma_n^2} \right) \quad (10)$$

For the equal-power source model, this definition is applied consistently when generating the training, validation, and test data.

3. Network Model

As a multi-layer nonlinear neural network, the convolutional neural network (CNN) has achieved great success in the field of computer vision due to its powerful feature extraction capability [40,41]. In the field of DOA estimation, classification-based convolutional neural networks are generally adopted. The network proposed in this paper adopts an encoder–dual prediction head design, as shown in Figure 2. Taking the spatial covariance matrix as input, the network extracts high-level feature representations through a shared feature encoder, and then estimates the elevation angles and azimuth angles of two targets respectively via two prediction heads with the same structure but independent parameters. This structure not only ensures the sufficiency of feature extraction but also guarantees the decoupling of the two angles. For the fixed two-source setting, each branch outputs 61 logits corresponding to the 1° grid over 0° – 60° . Independent sigmoid activations are used because more than one angular class can be active simultaneously. The marginal angle candidates are finally associated by the covariance-fitting procedure described in Section 3.3.

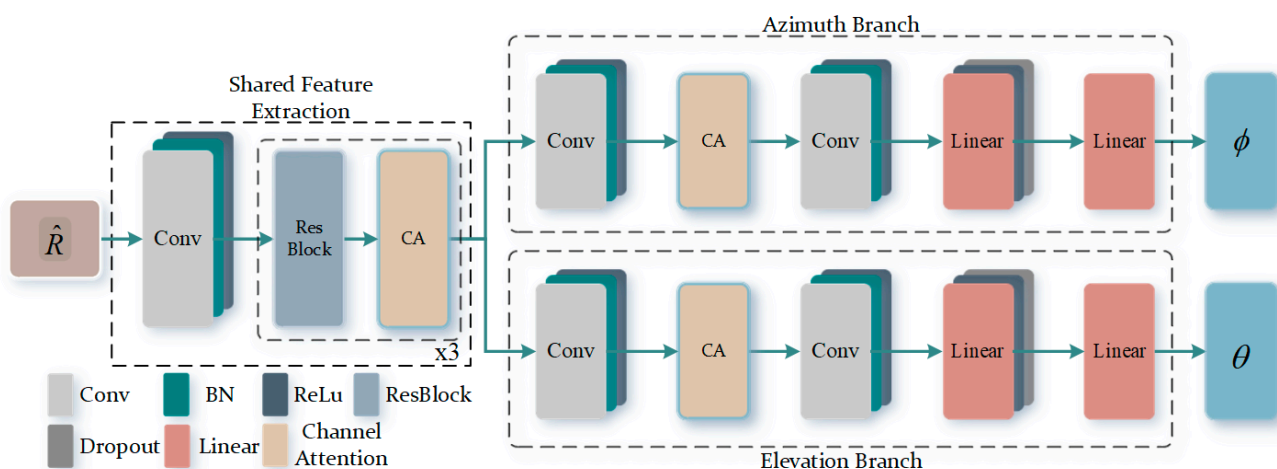


Figure 2. Network structure diagram.

3.1. Shared Feature Encoder

The first module shown in Figure 2 is the shared feature encoder, which is mainly used to extract the common features of elevation and azimuth angles in the array covariance matrix. This module contains an initial convolutional layer followed by three residual blocks with output channels 64, 128, and 256, respectively. A channel attention module (CAM) is applied after each residual block.

When the number of layers in a convolutional neural network increases, gradient vanishing and gradient explosion may occur [26], which have a significant impact on the training effect of the network. ResNet solves the problem of gradient vanishing through direct mapping [27]. The structure of an ordinary network is just a simple stack, while

a residual block has a cross-layer connection structure, which has certain advantages in backpropagation [42]:

$$\frac{\partial Loss}{\partial x_l} = \frac{\partial Loss}{\partial x_L} \times \left(\mathbf{I} + \frac{\partial}{\partial x_L} \sum_{i=1}^{L-1} F(x_i, W_i) \right) \quad (11)$$

where $\frac{\partial Loss}{\partial x_L}$ denotes the gradient propagated from a deeper layer. The identity term provides a direct gradient path in addition to the residual-transformation term. Accordingly, residual blocks can alleviate gradient degradation in practice, but they do not theoretically guarantee non-vanishing gradients under all network states [43].

The structure of the residual block used in this paper is shown in Figure 3, which includes two convolutional layers, two batch normalization layers, and corresponding activation functions. The activation function adopts the Rectified Linear Unit (ReLU), which can be expressed as follows [44]:

$$f(x) = \max(0, x) \quad (12)$$

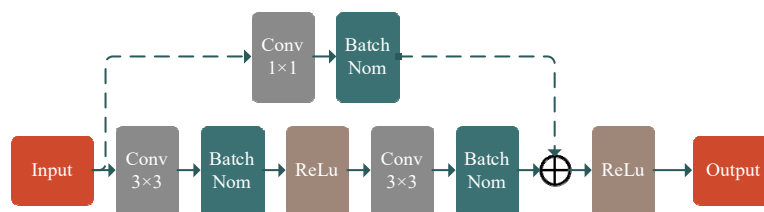


Figure 3. Residual block structure.

ReLU is computationally simple and avoids saturation on its positive branch, which is beneficial for optimization. Nevertheless, neither ReLU nor the residual connection alone guarantees that gradient-related optimization problems are completely eliminated.

The shared encoder also employs channel attention. For a feature tensor with C channels and spatial size $H \times W$, global average pooling first produces one descriptor per channel. Two fully connected transformations with reduction ratio r generate channel weights, after which each feature channel is explicitly recalibrated by its corresponding weight.

$$w = \sigma(W_2 \delta(W_1 g)) \\ \tilde{Z}_c = w_c Z_c, \quad c = 1, \dots, C. \quad (13)$$

where $g = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W Z(i, j)$ is the global average pooling feature, $W_1 \in \mathbb{R}^{C/r \times C}$ and $W_2 \in \mathbb{R}^{C \times C/r}$ are learnable parameters, δ is the ReLU activation function, and σ is the Sigmoid function. The last line makes the channel-wise feature recalibration explicit. With $r = 8$, the CAM bottleneck dimensions are 8, 16, and 32 for encoder outputs with 64, 128, and 256 channels, respectively.

3.2. Dual-Angle Prediction Heads

Elevation and azimuth are generated by the same received field and therefore share informative covariance features, but their prediction mappings are not identical. A fully joint head must optimize both angular tasks through the same task-specific parameters, which may introduce optimization interference, whereas two completely separate networks would duplicate the feature extractor. The proposed architecture therefore shares the encoder while assigning independent prediction-head parameters to the elevation and azimuth tasks. This choice is intended to reduce task interference; it is not presented as a theoretical guarantee of complete feature disentanglement.

Each prediction head follows the same feature-refinement, spatial-aggregation, and angle-classification pipeline, but the two heads do not share parameters. Batch normalization is applied after convolutional layers to stabilize training, and dropout is used before the final classifier as a regularizer against overfitting.

In the second stage, the spatial dimension is compressed through a Global Average Pooling (GAP) layer to convert the feature map into a fixed-length feature vector.

$$\mathbf{h}^{(k)} = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W |\mathbf{H}^{(k)}(i, j)| \quad (14)$$

where $H \times W$ is the spatial dimension of the feature map. The global average pooling strategy can reduce the number of parameters while retaining the statistical characteristics of spatial information, thereby enhancing the robustness of the model to input changes.

After global average pooling and the fully connected feature layer, each prediction head outputs $C = 61$ logits. The elevation and azimuth logits and their corresponding probabilities are defined by the following:

$$\begin{aligned} z_{\theta} &= \mathbf{w}_{\theta}^{(k)\top} \mathbf{h}^{(k)} + b_{\theta}^{(k)}, p_{\theta} = \sigma(z_{\theta}), \\ z_{\phi} &= \mathbf{w}_{\phi}^{(k)\top} \mathbf{h}^{(k)} + b_{\phi}^{(k)}, p_{\phi} = \sigma(z_{\phi}), C = 61. \end{aligned} \quad (15)$$

The two logit vectors and two probability vectors each have length C . Independent sigmoid activation is required because, with two sources, more than one grid location in a marginal angular dimension may be active simultaneously. The branch output is therefore an unordered set of occupied angular grid cells rather than two source-indexed probability rows.

$$y_{\theta,c} = \mathbb{I}(g_c \in \Theta), \quad y_{\phi,c} = \mathbb{I}(g_c \in \Phi), \quad c = 1, \dots, C \quad (16)$$

The supervision for each branch is a multi-hot occupancy vector. For each angular grid cell, the target value is one if that direction is occupied by at least one source and zero otherwise. The elevation and azimuth labels are defined in Equation (14).

Because the branch outputs represent occupied grid directions rather than preordered sources, no rule such as sorting by elevation, azimuth, received power, confidence, or data-generation order is imposed during training. This avoids introducing artificial source indices into the training labels. If two sources share the same marginal angle, that grid class is activated once; the known source count is handled later by the pairing stage, which may reuse the shared marginal direction when constructing the two 2-D DOA hypotheses.

3.3. Loss Function Design

The loss function is used to measure the difference between the model's predicted values and the true values, which is crucial for model performance. In the 2-D DOA estimation task based on deep learning, the model needs to simultaneously optimize the angle classification accuracy and the correlation of multi-target angle pairing. To address this problem, this paper proposes a hybrid loss function fusing Binary Cross-Entropy Loss and Contrastive Loss. Binary Cross-Entropy Loss is a commonly used loss function for label classification problems, which can ensure the accuracy of angle classification. Contrastive Loss can reflect the correlation between elevation and azimuth angles in 2-D DOA estimation, thereby solving the angle pairing problem in multi-target scenarios.

Because each branch produces one 61-dimensional vector while multiple angle classes can be active simultaneously, the two angular tasks are formulated as multi-label classification rather than mutually exclusive categorical classification. Training therefore uses

binary cross-entropy for marginal-angle occupancy, while elevation–azimuth association is performed explicitly during inference using the physical covariance model.

For $q \in \{\theta, \phi\}$, weighted binary cross-entropy is used for the C independent occupancy labels:

$$L_{BCE} = -\frac{1}{C} \sum_{c=1}^C [\omega + y_{q,c} \log p_{q,c} + (1 - y_{q,c}) \log(1 - p_{q,c})], q = \theta \text{ or } \phi \quad (17)$$

The weighted binary cross-entropy is evaluated independently over the C grid cells and then averaged. An occupied class contributes $-(\omega + \log p)$, whereas an unoccupied class contributes $-\log(1 - p)$. The positive-class weight compensates for the imbalance between the small number of occupied cells and the many unoccupied cells. For two sources and 61 grid classes, an initial value of 29.5 follows from the negative-to-positive class ratio and can be tuned on the validation set.

The total training objective is the weighted sum of the two branch losses:

At inference, up to K dominant elevation and azimuth candidates are extracted from the two occupancy vectors. These candidates are unordered. For $K = 2$ with two distinct candidates in each marginal dimension, there are only two possible elevation–azimuth permutations. For a candidate permutation π , define the steering matrix $A_\pi = [a(\hat{\theta}_1, \hat{\phi}_{\{\pi(1)\}}), a(\hat{\theta}_2, \hat{\phi}_{\{\pi(2)\}})]$.

$$L_{cls} = \lambda_\theta L_{BCE}^{(\theta)} + \lambda_\phi L_{BCE}^{(\phi)} \quad (18)$$

Under the independent-source covariance model, the signal powers and noise variance for each permutation are estimated by nonnegative least squares:

$$B_\pi = [\text{vec}(a_{\pi,1} a_{\pi,1}^H), \text{vec}(a_{\pi,2} a_{\pi,2}^H), \text{vec}(I)], \quad \hat{\eta}_\pi = \arg \min_{\eta \geq 0} \|\text{vec}(\hat{R}) - B_\pi \eta\|_2^2 \quad (19)$$

The selected association is the permutation that gives the minimum covariance-fitting residual:

$$\pi^* = \arg \min_{\pi \in \mathcal{S}_2} \|\text{vec}(\hat{R}) - B_\pi \hat{\eta}_\pi\|_2^2 \quad (20)$$

The final 2-D estimates are $\left\{ \left(\hat{\theta}_1, \hat{\phi}_{\{\pi^*(1)\}} \right), \left(\hat{\theta}_2, \hat{\phi}_{\{\pi^*(2)\}} \right) \right\}$. If the two sources share one marginal angle, the single dominant candidate in that dimension may be reused when the two 2-D hypotheses are formed. Thus, the association does not depend on angle sorting, peak-confidence ordering, or data-generation order.

By introducing Contrastive Loss, the similarity of angle pairs of the same target can be forced to be higher than the threshold, and the similarity of angle pairs of different targets can be lower than the threshold, explicitly constraining the angle pairing relationship, and avoiding incorrect pairing caused by Cross-Entropy Loss ignoring angle correlation.

3.4. Optimization Strategy Design

In CNN, the optimization process is crucial to ensure model convergence. Aiming at the characteristics of the DOA estimation task, this paper designs a corresponding optimization strategy, including optimizer selection, learning rate scheduling, and gradient management, to ensure that the model has good generalization ability while converging stably.

In terms of optimizer selection, this paper adopts AdamW instead of the Adam optimizer [45]. This optimizer decouples weight decay from gradient update, making weight decay play a regularization role, thereby solving the conflict between weight decay and adaptive learning rate in the Adam optimizer.

The learning rate scheduling strategy is also crucial for model convergence. This paper adopts the OneCycle learning rate strategy [46]. Under this strategy, the learning rate first increases and then decreases, which not only ensures the convergence speed of the model in the early stage but also ensures the stability of training in the later stage. The two stages of the OneCycle strategy can be divided into a linear rise stage and a cosine decay stage. In the rise stage, the learning rate linearly increases from the initial value to the maximum value, and the duration of the rise stage accounts for 30% of the total training steps. In the decay stage, the learning rate decays from the maximum value to the initial value according to the cosine function.

4. Simulation Experiments

In the simulation experiments proposed in this paper, a 4×4 URA is adopted with an array element interval of half a wavelength. The proposed 2-D DOA estimation network is implemented using the PyTorch 2.5.1 framework. The computer for neural network training is configured with an Intel Core i7-11800 CPU, an NVIDIA RTX 3070 GPU, and 64 GB of memory. Unless otherwise stated, the simulation scenario settings in this paper default to $k = 2$ signal sources and $L = 200$ snapshots.

4.1. Simulation Parameter Settings

4.1.1. Data Generation and Preprocessing

In the simulation experiments adopted in this paper, the azimuth range of the signal sources to be estimated is $[0^\circ, 60^\circ]$, and the elevation range is $[0^\circ, 60^\circ]$. The resolution of discrete angles is set to 1° in this paper. The labels of azimuth angles are $\varphi = \{0^\circ, 1^\circ, 2^\circ, \dots, 60^\circ\}$ and the labels of elevation angles are $\theta = \{0^\circ, 1^\circ, 2^\circ, \dots, 60^\circ\}$, with 61 angle labels each. The training data is generated through Matlab R2021b simulation and stored in HDF5 files, with the input feature being the array covariance matrix. The training set and validation set are divided in a ratio of 8:2, and the input features are standardized by channel to accelerate model convergence.

4.1.2. Network Configuration and Training Hyperparameters

The structure of the model is shown in Figure 2. The input $2 \times 16 \times 16$ covariance tensor first passes through a 3×3 convolutional layer that maps the channel number from 2 to 64 with stride 1 and padding 1, followed by batch normalization and ReLU. The shared backbone then contains three residual blocks with output channels 64, 128, and 256. The first two blocks use stride 2, reducing the feature-map size from 16×16 to 8×8 and then to 4×4 ; the third uses stride 1. Each residual block contains two 3×3 convolutions, and a 1×1 shortcut convolution is used when spatial size or channel number changes. A CAM follows each residual block. With reduction ratio $r = 8$, the CAM bottleneck dimensions are 8, 16, and 32 for 64-, 128-, and 256-channel inputs, respectively.

The $256 \times 4 \times 4$ feature map output by the common feature extraction backbone network is fed into the azimuth and elevation prediction branches with the same structure respectively. Each branch first performs feature refinement through 3 stacked “convolution-BN-ReLU” layers. The first convolutional layer reduces the 256 channels to 128 channels, the second convolutional layer reduces the 128 channels to 64 channels, and the third convolutional layer remains unchanged. All convolutional layers use 3×3 convolution kernels with a stride of 1 and padding of 1 to maintain the feature map size. After the first convolutional layer, a CAM with a compression ratio of 8 is also introduced to further enhance the perception ability of specific angle information. After feature refinement, the $64 \times 4 \times 4$ feature map is compressed into a 64-dimensional feature vector through Global Average Pooling (GAP). This vector is then fed into a fully connected layer with

256 neurons (ReLU activation), and a Dropout layer with a dropout rate of 0.4 is introduced to suppress overfitting. Finally, a fully connected layer with an output dimension of 61 is used to obtain the labels of angle categories.

In the training hyperparameter settings, the batch size is 128, and the total number of training epochs is set to 200. The initial learning rate is set to 1×10^{-4} , and a cosine annealing scheduler is used for dynamic adjustment, with the minimum learning rate being 1×10^{-6} , to ensure the convergence speed of the model and the ability to jump out of local optima. The loss function adopts the hybrid loss function proposed in this paper, with the weight of Cross-Entropy Loss being 0.9 and the weight of Contrastive Loss being 0.1.

4.1.3. Baseline Definition and Fairness Controls

The conventional CNN takes the real and imaginary parts of the sample covariance matrix as a $2 \times 16 \times 16$ input tensor. It consists of three convolutional layers and two fully connected layers, without residual connections, channel attention, dual prediction heads, or contrastive learning. The final layer outputs 122 logits, which are divided into two 61-dimensional vectors for elevation and azimuth estimation. The baseline contains approximately 0.158 million trainable parameters. The comparison of this method against our proposed method is presented in Table 1.

Table 1. Training configurations of the baseline CNN and the proposed method.

Item	CNN Baseline	Proposed Method
Input	$2 \times 16 \times 16$ covariance tensor	$2 \times 16 \times 16$ covariance tensor
Convolutional layers	3 layers with 32, 64, and 128 channels	Shared encoder: one $2 \rightarrow 64$ stem convolution and three residual blocks with 64, 128, and 256 channels, with a CAM after each block; each prediction branch: three convolutional layers with 128, 64, and 64 channels, with an additional CAM after the first branch convolution
Classifier	GAP and two fully connected layers; dropout = 0.4	Separate elevation and azimuth classifiers; each uses GAP, a $64 \rightarrow 256$ fully connected layer, dropout = 0.4, and a $256 \rightarrow 61$ output layer
Output	61 elevation classes + 61 azimuth classes	61 elevation probabilities + 61 azimuth probabilities; the two highest peaks are selected from each branch
Network depth	3 convolutional layers and 2 fully connected layers	13 main convolutional layers and 4 classifier fully connected layers; additionally, three 1×1 projection convolutions and five two-layer CAMs
Trainable parameters	Approximately 0.158 M	Approximately 2.14 M
Loss function	Sum of elevation and azimuth cross-entropy losses	Sum of weighted elevation and azimuth binary cross-entropy losses; source-wise elevation–azimuth pairing is determined by covariance-fitting residual minimization during inference
Training settings	AdamW, batch size 128, 200 epochs, initial learning rate 10^{-4}	AdamW; batch size 128; 200 epochs; initial learning rate 10^{-4} ; cosine-annealing learning-rate scheduler

For MUSIC, the number of sources is fixed to $K = 2$ and a 1° two-dimensional search grid identical to the classification grid is used. ESPRIT is supplied with the same known source number and operates on the same sample covariance matrices. All timing measurements exclude one-time neural-network training and include only per-sample online inference or estimation. A separate single-head ablation that retains the proposed encoder and uses one shared task representation should be reported in the revised experiments to isolate the dual-branch design itself.

A tensor-decomposition-based estimator, denoted as Tensor-CPD-NLS, was included as a representative multidimensional DOA estimation baseline. The received snapshots of the uniform rectangular array were rearranged into a third-order complex-valued tensor, whose first two modes corresponded to the two spatial dimensions of the array and whose third mode corresponded to the snapshot dimension. A rank- K complex canonical polyadic decomposition was first performed to separate the components associated with different sources. Initial elevation and azimuth estimates were then extracted from the phase progressions of the two spatial factor matrices. Finally, the initial estimates were jointly refined by nonlinear least-squares optimization under the URA steering-vector model, with the tensor reconstruction error used as the objective function. Because the two spatial factors belonging to the same rank-one component share an identical source index, the estimated elevation and azimuth angles were naturally paired.

4.2. Simulation Result Analysis

To verify the estimation effect of the model, this paper analyzes the DOA estimation effect under four different scenarios. The results in Figure 4a are the angle estimation results under general scenarios, with the angles set to elevation angle $\theta_1 = 30^\circ$, $\theta_2 = 15^\circ$ and azimuth angle $\varphi_1 = 20^\circ$, $\varphi_2 = 25^\circ$. Under this scenario, both traditional algorithms and the method proposed in this paper have good estimation effects; Figure 4b uses $\theta_1 = \theta_2 = 30^\circ$ and $\varphi_1 = 20^\circ$, $\varphi_2 = 25^\circ$; this is not a small-separation case in both dimensions, but a shared-elevation case with a 5° azimuth separation. In the revised inference rule, the elevation branch may retain the single dominant 30° direction and reuse it when pairing the two azimuth candidates; in Figure 4c, the SNR is -10 dB. Under this scenario, the estimation effect of traditional methods is poor with large errors, while the result estimated by the network proposed in this paper has a small deviation from the true value; in Figure 4d, the elevation angles of the targets are $\theta_1 = 10.5^\circ$, $\theta_2 = 18.3^\circ$ and the azimuth angles are $\varphi_1 = 26.2^\circ$, $\varphi_2 = 45.7^\circ$, and the angle settings are not on the label grid of the network. Under this scenario, since the estimation results of the MUSIC algorithm and the network proposed in this paper are both on the grid, there is grid error, but the error is small. Although the ESPRIT algorithm is not affected by grid error, there is still a certain estimation error.

The operation times of the MUSIC algorithm, ESPRIT algorithm, and the network proposed in this paper are shown in Table 2. Since the MUSIC algorithm requires eigenvalue decomposition and spectral peak search, its computational complexity is on the order of $O(M^3 + KM^2)$, where K is the number of spectral peak search points, so it requires more operation time and has poor real-time performance. Although the ESPRIT algorithm can avoid the computational complexity caused by spectral peak search, it still needs to perform eigenvalue decomposition, with a computational complexity of $O(M^3)$. Although the computation amount is reduced compared with the MUSIC algorithm, the improvement in real-time performance is not significant. With the method proposed in this paper, due to the neural network replacing complex subspace extraction, spectral peak search, and optimization processes with efficient forward propagation calculations, the computational speed of the proposed network is also significantly improved.

Table 2. Operation time of different methods.

Method	Implementation	Execution Device	Mean Online Time
MUSIC	MATLAB R2021b	Intel i7-11800 CPU	1.6365 s
ESPRIT	MATLAB R2021b	Intel i7-11800 CPU	0.8459 s
Conventional CNN	PyTorch 2.5.1	NVIDIA RTX 3070 GPU	0.0219 s
Proposed method	PyTorch 2.5.1	NVIDIA RTX 3070 GPU	0.0243 s

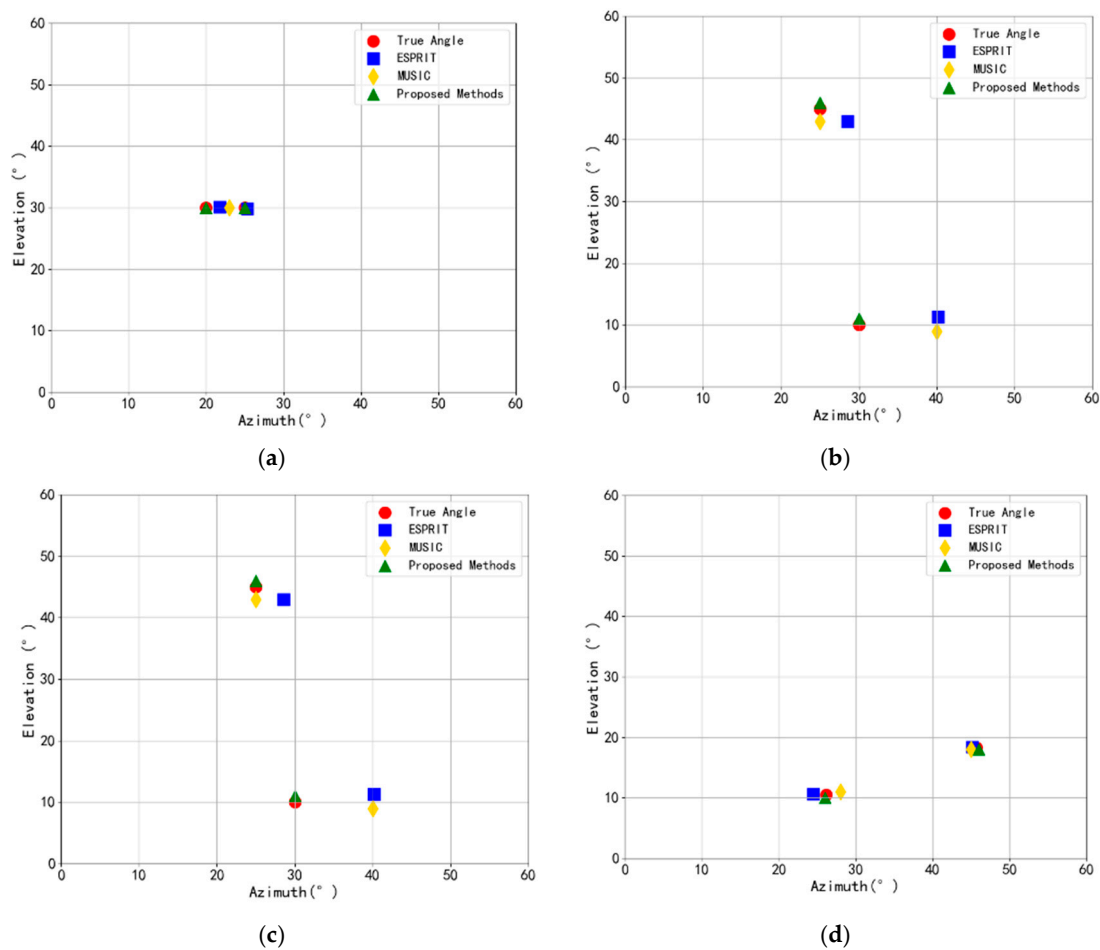


Figure 4. Estimation results: (a) general scenario; (b) small-angle interval scenario; (c) low SNR scenario; (d) off-grid angle scenario.

4.3. Comparison of DOA Estimation Accuracy

In terms of estimation accuracy, the proposed network is compared with MUSIC, ESPRIT, Tensor-CPD-NLS, and the conventional CNN baseline. The Cramér–Rao lower bound (CRLB) is additionally included as a theoretical performance reference rather than as an estimator. Unless otherwise stated, the angular separation between the two sources is 10° . Root-mean-square error (RMSE) is used for elevation, azimuth, and the combined two-dimensional error:

$$RMSE = \sqrt{\frac{1}{2KT_{mc}} \sum_{k=1}^K \sum_{t_{mc}=1}^{T_{mc}} \left(\vartheta_k^{(t)} - \hat{\vartheta}_k^{(t)} \right)^2} \quad (21)$$

where T_{mc} represents the number of Monte Carlo experiments, K represents the number of signal sources, $\hat{\vartheta}_k^{(t)}$ and $\vartheta_k^{(t)}$ respectively represent the angle estimation value and true value of the k -th signal source in the t -th Monte Carlo experiment.

4.3.1. Estimation Accuracy Under Different SNRs

This paper compares different methods under the scenario where the SNR ranges from -10 dB to 10 dB with an interval of 5 dB. At each SNR level, we conduct 1000 Monte Carlo experiments, and the variation of RMSE with SNR is shown in Figure 5.

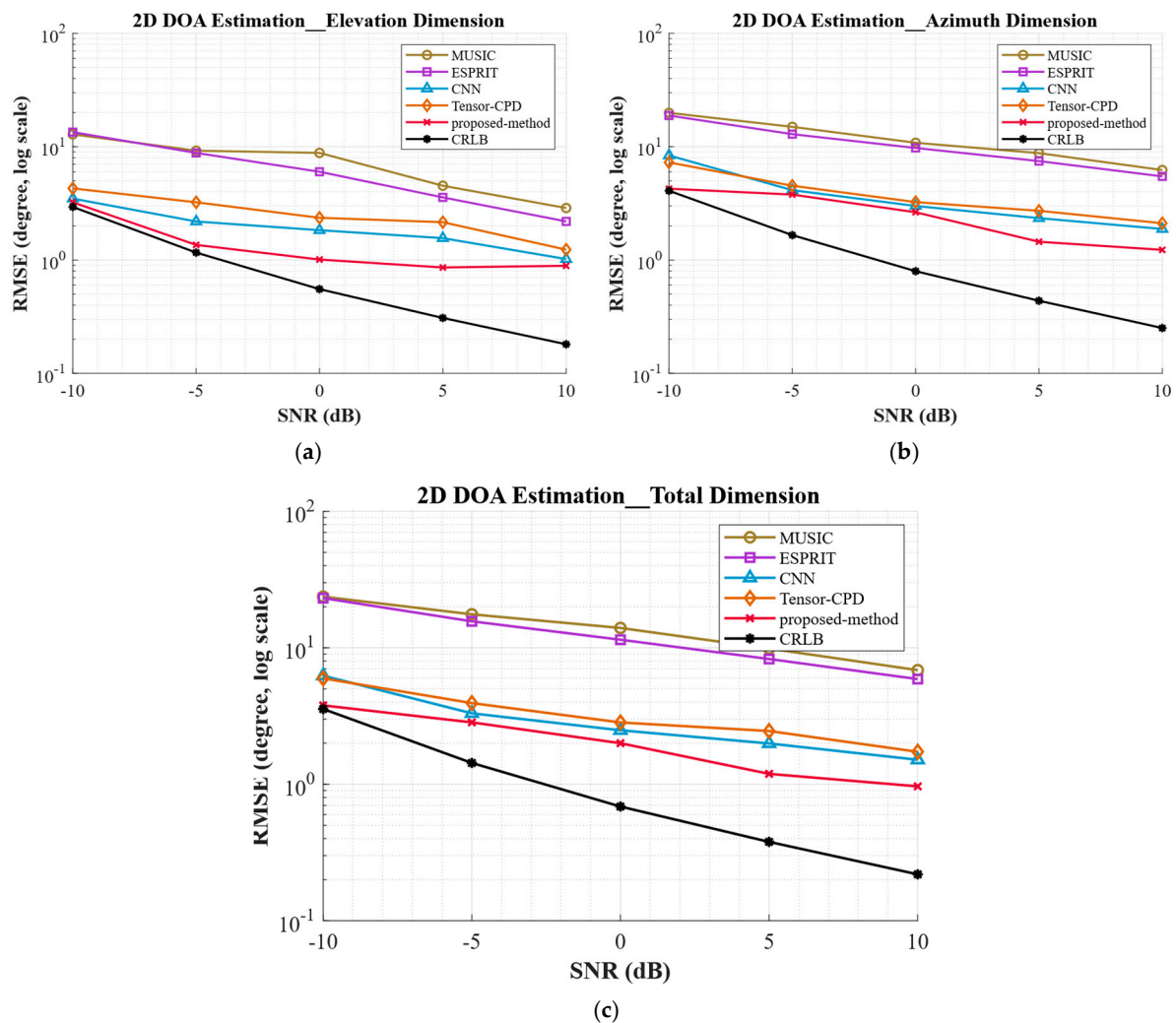


Figure 5. RMSE under different SNRs: (a) elevation dimension; (b) azimuth dimension; (c) total dimension.

From the result graphs, it can be seen that under various SNR conditions, the data-driven 2-D DOA estimation method has higher estimation accuracy and smaller errors compared with the model-driven estimation method. Compared with the conventional CNN network, the network structure proposed in this paper significantly improves the estimation accuracy of elevation angles and also improves the estimation accuracy of azimuth angles to a certain extent because the branch networks estimate the azimuth and elevation angles respectively. As the SNR decreases, the model-driven DOA estimation methods are greatly affected by noise. Although the network-based DOA estimation methods are also affected by noise, the impact on accuracy is small. To provide quantitative comparisons at representative operating points, at SNR = 0 dB, the proposed method achieved elevation, azimuth, and combined 2-D RMSEs of 1.010° , 2.640° , and 1.999° , respectively. The combined RMSEs of MUSIC, ESPRIT, the conventional CNN, and Tensor-CPD-NLS were 13.971° , 11.457° , 2.486° , and 2.837° , respectively, corresponding to relative RMSE reductions of 85.69%, 82.55%, 19.59%, and 29.54%. At SNR = 10 dB, the proposed method obtained a combined RMSE of 0.965° , representing reductions of 85.93%, 83.64%, 36.18%, and 44.22% relative to MUSIC (6.858°), ESPRIT (5.898°), the conventional CNN (1.512°), and Tensor-CPD-NLS (1.730°), respectively.

4.3.2. Estimation Accuracy Under Different Snapshots

To verify the impact of different numbers of snapshots on the 2-D DOA estimation accuracy of different methods, this paper conducts Monte Carlo experiments under the scenario where SNR = 10 dB and number of snapshots is {50, 100, 150, 200, 250}. The results are shown in Figure 6.

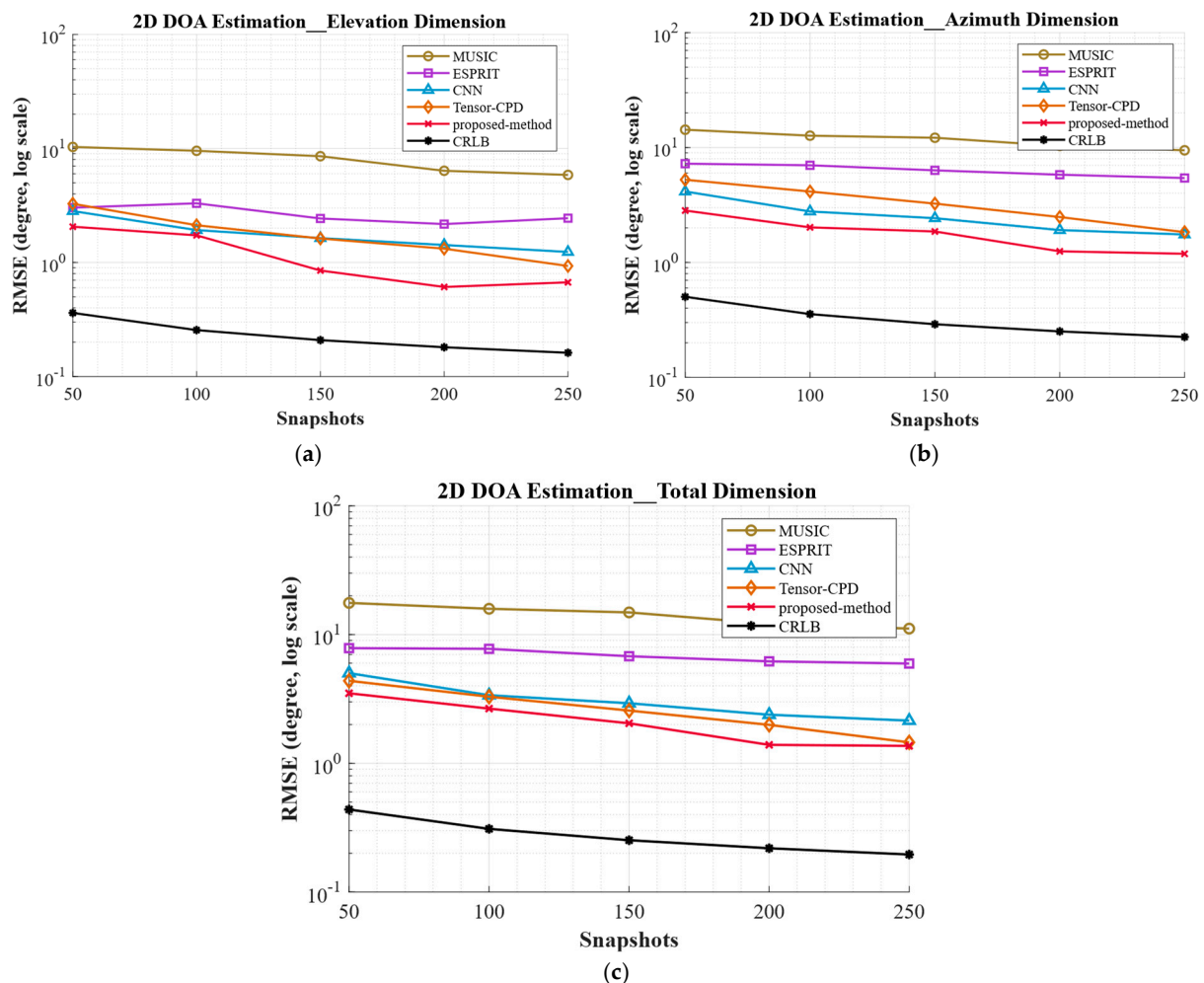


Figure 6. RMSE under different numbers of snapshots: (a) elevation dimension; (b) azimuth dimension; (c) total dimension.

From the results, it can be seen that the network proposed in this paper has high DOA estimation accuracy under various numbers of snapshots, which is significantly superior to the model-based 2-D DOA estimation methods. Moreover, compared with the conventional CNN network, the network proposed in this paper also has a certain improvement in accuracy when the number of snapshots is small. When the number of snapshots decreases, the angle estimation accuracy of the MUSIC method and ESPRIT method is greatly affected. At $L = 50$ snapshots, the proposed method achieved elevation, azimuth, and combined 2-D RMSEs of 2.06° , 2.83° , and 3.50° , respectively. The combined RMSEs of MUSIC, ESPRIT, the conventional CNN, and Tensor-CPD-NLS were 17.625° , 7.845° , 5.016° , and 4.374° , respectively; thus, the proposed method reduced the RMSE by 80.14%, 55.38%, 30.22%, and 19.98% relative to these baselines. At $L = 200$ snapshots, the proposed method achieved a combined RMSE of 1.391° , compared with 12.139° , 6.196° , 2.385° , and 1.990° for MUSIC, ESPRIT, the conventional CNN, and Tensor-CPD-NLS, respectively, corresponding to RMSE reductions of 88.54%, 77.55%, 41.68%, and 30.10%.

4.4. Ablation Study on the Prediction-Head Design

To evaluate the contribution of the dual-prediction-head architecture, an ablation experiment was conducted by comparing the proposed dual-head network with a single-head variant. Except for the prediction-head structure, the two models used the same input representation, shared feature encoder, training dataset, data partition, optimization strategy, and training hyperparameters. Unless otherwise specified, two signal sources and 200 snapshots were considered.

In the proposed dual-head model, the feature map produced by the shared encoder is separately fed into two prediction heads with identical architectures but independent parameters. The two heads are responsible for elevation and azimuth estimation, respectively, allowing each branch to learn task-specific angular representations. Each prediction head contains convolutional feature-refinement layers, global average pooling, and fully connected layers, and finally outputs a 61-dimensional angular probability vector.

For the single-head variant, the shared feature encoder was kept unchanged, while the two independent prediction heads were replaced by one common prediction head. The single prediction head used the same basic feature-refinement structure as one branch of the dual-head model, including convolutional layers, global average pooling, and a fully connected layer. Its final classification layer produced a 122-dimensional output vector, which was divided into two 61-dimensional subsets corresponding to the elevation and azimuth predictions. Therefore, the single-head model was required to generate both angular estimates from the same task-specific feature representation, whereas the dual-head model employed independent parameters to refine the features for the two angular dimensions.

The experimental results are summarized in Figure 7. These results show that estimating the two angular dimensions through independent prediction heads provides substantially better estimation accuracy than forcing a single prediction head to learn both tasks simultaneously.

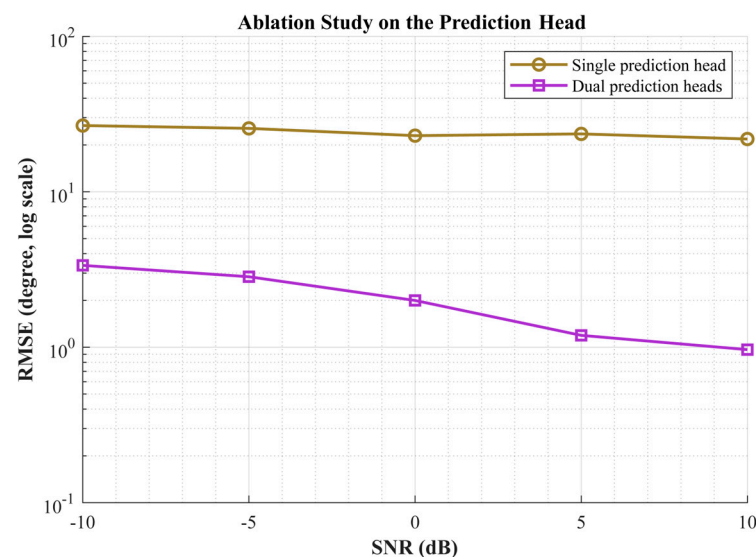


Figure 7. Ablation study on the prediction head.

Because the input data, shared encoder, training samples, and optimization settings were kept unchanged, the observed performance difference indicates that the prediction-head design has an important influence on two-dimensional DOA estimation. In the single-head model, elevation and azimuth estimation share the same task-specific transformation and output representation, which may cause competition between the two

learning objectives. By contrast, the dual-head architecture retains the common spatial information extracted by the shared encoder while allowing the two branches to adapt independently to the feature distributions associated with elevation and azimuth. The considerably lower RMSE obtained by the dual-head model therefore provides empirical evidence that task-specific branch separation can alleviate interference between the two angular estimation tasks. Nevertheless, this result should be interpreted as experimental support for reduced feature interference rather than as a formal theoretical proof of complete feature disentanglement.

5. Conclusions

This paper presents a dual-branch CNN for 2-D DOA estimation using the sample covariance matrix of a URA. The shared encoder reuses common spatial information, while independent elevation and azimuth heads provide task-specific decision parameters. Consistent with the 61-dimensional output of each branch, angle estimation is formulated as multi-label occupancy prediction with sigmoid activations and weighted binary cross-entropy. The unordered marginal angle candidates are associated through a URA covariance-fitting criterion rather than an assumed source order. The resulting formulation is internally consistent with the stated network output dimension and provides a reproducible multi-source association procedure.

The present method still has clear limitations. Its outputs are constrained to the 1° training grid, so grid-mismatched inputs remain subject to quantization error and the method does not perform continuous off-grid estimation. The current inference also assumes a known source number $K = 2$ and uses a covariance-fitting pairing model whose baseline form is derived under mutually independent sources. Future work should investigate classification–regression or gridless outputs for continuous-angle estimation, extend the pairing model to unknown source numbers and correlated/coherent sources, and evaluate generalization to array perturbations and more complex array geometries.

Author Contributions: Conceptualization, F.L. and G.Z.; methodology, F.L.; software, F.L.; validation, Y.S., Y.B. and H.Z.; formal analysis, F.L.; writing—original draft preparation, F.L.; writing—review and editing, G.Z.; visualization, Y.S.; supervision, G.Z.; funding acquisition, Y.S. 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 62401619.

Data Availability Statement: The data that support the findings of this study are available from the author upon reasonable request.

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

References

1. Zhang, X.F.; Li, J.F.; Xu, D.Z.; Shi, C.G.; Cheng, H.W.; Wang, F.; Zhang, G. *Array Signal Processing and MATLAB Implementation*; Electronic Industry Press: Beijing, China, 2020; p. 464.
2. Qin, G.; Amin, M.G.; Zhang, Y.D. DOA Estimation Exploiting Sparse Array Motions. *IEEE Trans. Signal Process.* **2019**, *67*, 3013–3027. [[CrossRef](#)]
3. He, J.; Shu, T.; Li, L.; Truong, T.-K. Mixed Near-Field and Far-Field Localization and Array Calibration with Partly Calibrated Arrays. *IEEE Trans. Signal Process.* **2022**, *70*, 2105–2118. [[CrossRef](#)]
4. Mao, Z.; Liu, S.; Zhang, Y.D.; Han, L.; Huang, Y. Joint DoA-Range Estimation Using Space-Frequency Virtual Difference Coarray. *IEEE Trans. Signal Process.* **2022**, *70*, 2576–2592. [[CrossRef](#)]
5. Zheng, W.; Zhang, X.; Zhai, H. Generalized Coprime Planar Array Geometry for 2-D DOA Estimation. *IEEE Commun. Lett.* **2017**, *21*, 1075–1078. [[CrossRef](#)]
6. Xia, D.; Wang, X.; Han, J.; Xue, H.; Liu, G.; Shi, Y.; Li, L.; Cui, T.J. Accurate 2-D DoA Estimation Based on Active Metasurface with Nonuniformly Periodic Time Modulation. *IEEE Trans. Microw. Theory Tech.* **2023**, *71*, 3424–3435. [[CrossRef](#)]

7. Florio, A.; Coviello, G.; Talarico, C.; Avitabile, G. Adaptive DDS-PLL Beamsteering Architecture Based on Real-Time Angle-of-Arrival Estimation. In Proceedings of the 2024 IEEE 67th International Midwest Symposium on Circuits and Systems (MWSCAS), Springfield, MA, USA, 11–14 August 2024; pp. 628–631. [\[CrossRef\]](#)
8. Florio, A.; Coviello, G.; Avitabile, G. Low-Complexity Angle-of-Arrival Estimation Using 1-Bit Conversion. *IEEE Trans. Instrum. Meas.* **2026**, *75*, 8001913. [\[CrossRef\]](#)
9. Schmidt, R. Multiple Emitter Location and Signal Parameter Estimation. *IEEE Trans. Antennas Propag.* **1986**, *34*, 276–280. [\[CrossRef\]](#)
10. Wax, M.; Shan, T.-J.; Kailath, T. Spatio-Temporal Spectral Analysis by Eigenstructure Methods. *IEEE Trans. Acoust. Speech Signal Process.* **1984**, *32*, 817–827. [\[CrossRef\]](#)
11. Zhang, X.; Xu, L.; Xu, L.; Xu, D. Direction of Departure (DOD) and Direction of Arrival (DOA) Estimation in MIMO Radar with Reduced-Dimension MUSIC. *IEEE Commun. Lett.* **2010**, *14*, 1161–1163. [\[CrossRef\]](#)
12. Roy, R.; Kailath, T. ESPRIT-Estimation of Signal Parameters via Rotational Invariance Techniques. *IEEE Trans. Acoust. Speech Signal Process.* **1989**, *37*, 984–995. [\[CrossRef\]](#)
13. Yang, P.; Yang, F.; Nie, Z.-P. Doa Estimation with Sub-Array Divided Technique and Interpolated Esprit Algorithm on a Cylindrical Conformal Array Antenna. *Prog. Electromagn. Res.* **2010**, *103*, 201–216. [\[CrossRef\]](#)
14. Mathews, C.P.; Zoltowski, M.D. Eigenstructure Techniques for 2-D Angle Estimation with Uniform Circular Arrays. *IEEE Trans. Signal Process.* **1994**, *42*, 2395–2407. [\[CrossRef\]](#)
15. Haardt, M.; Zoltowski, M.D.; Mathews, C.P.; Nossek, J. 2D Unitary ESPRIT for Efficient 2D Parameter Estimation. In *Proceedings of the 1995 International Conference on Acoustics, Speech, and Signal Processing*; IEEE: Detroit, MI, USA, 1995; Volume 3, pp. 2096–2099.
16. Tayem, N.; Kwon, H.M. L-Shape 2-Dimensional Arrival Angle Estimation with Propagator Method. In *Proceedings of the 2005 IEEE 61st Vehicular Technology Conference*; IEEE: Stockholm, Sweden, 2005; Volume 1, pp. 6–10.
17. Donoho, D.L. Compressed Sensing. *IEEE Trans. Inform. Theory* **2006**, *52*, 1289–1306. [\[CrossRef\]](#)
18. Candes, E.J.; Romberg, J.; Tao, T. Robust Uncertainty Principles: Exact Signal Reconstruction from Highly Incomplete Frequency Information. *IEEE Trans. Inform. Theory* **2006**, *52*, 489–509. [\[CrossRef\]](#)
19. Malioutov, D.; Cetin, M.; Willsky, A.S. A Sparse Signal Reconstruction Perspective for Source Localization with Sensor Arrays. *IEEE Trans. Signal Process.* **2005**, *53*, 3010–3022. [\[CrossRef\]](#)
20. Wu, C.; Elangage, J. Multiemitter Two-Dimensional Angle-of-Arrival Estimator via Compressive Sensing. *IEEE Trans. Aerosp. Electron. Syst.* **2020**, *56*, 2884–2895. [\[CrossRef\]](#)
21. Chai, Y. Compressed Sensing Based DOA Estimation Algorithm. In Proceedings of the 2024 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB), Toronto, ON, Canada, 19–21 June 2024; pp. 1–5.
22. Hu, Y.; Zhao, T.; Wang, Y. Direct simulation of two-dimensional isotropic or anisotropic random field from sparse measurement using Bayesian compressive sampling. *Stoch. Environ. Res. Risk Assess.* **2019**, *33*, 1477–1496. [\[CrossRef\]](#)
23. Yang, Z.; Xie, L.; Zhang, C. Off-Grid Direction of Arrival Estimation Using Sparse Bayesian Inference. *IEEE Trans. Signal Process.* **2013**, *61*, 38–43. [\[CrossRef\]](#)
24. Liu, L.; Liu, Q. Covariance Reconstruction for Coherent Signal DOA Estimation with Limited Snapshots. *IEEE Trans. Aerosp. Electron. Syst.* **2025**, *61*, 15001–15013. [\[CrossRef\]](#)
25. Ye, Z.; Liu, C. 2-D DOA Estimation in the Presence of Mutual Coupling. *IEEE Trans. Antennas Propag.* **2008**, *56*, 3150–3158. [\[CrossRef\]](#)
26. Deng, W.; Li, K.; Zhao, H. A Flight Arrival Time Prediction Method Based on Cluster Clustering-Based Modular with Deep Neural Network. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 6238–6247. [\[CrossRef\]](#)
27. Chen, H.; Li, Y.; Zheng, B.; Chen, T. Hyperspectral remote sensing image classification based on domain-level complementarity of spatial-spectral component. *Neural Netw.* **2026**, *198*, 108610. [\[CrossRef\]](#) [\[PubMed\]](#)
28. Pastorino, M.; Randazzo, A. A Smart Antenna System for Direction of Arrival Estimation Based on a Support Vector Regression. *IEEE Trans. Antennas Propag.* **2005**, *53*, 2161–2168. [\[CrossRef\]](#)
29. 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. [\[CrossRef\]](#)
30. Bialer, O.; Garnett, N.; Tirer, T. Performance Advantages of Deep Neural Networks for Angle of Arrival Estimation. In *Proceedings of the ICASSP 2019—2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*; IEEE: Brighton, UK, 2019; pp. 3907–3911.
31. Elbir, A.M. DeepMUSIC: Multiple Signal Classification via Deep Learning. *IEEE Sens. Lett.* **2020**, *4*, 7001004. [\[CrossRef\]](#)
32. Hu, S.; Zeng, C.; Liu, M.; Tao, H.; Zhao, S.; Liu, Y. Robust DOA Estimation Using Deep Complex-Valued Convolutional Networks with Sparse Prior. In Proceedings of the 2023 6th International Conference on Information Communication and Signal Processing (ICICSP), Xi'an, China, 23–25 September 2023; pp. 234–239.
33. 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* **2023**, *24*, 91. [\[CrossRef\]](#) [\[PubMed\]](#)

34. Guo, Y.; Zhang, Z.; Huang, Y. Dual Class Token Vision Transformer for Direction of Arrival Estimation in Low SNR. *IEEE Signal Process. Lett.* **2024**, *31*, 76–80. [\[CrossRef\]](#)
35. Merkofer, J.P.; Revach, G.; Shlezinger, N.; Van Sloun, R.J.G. Deep Augmented Music Algorithm for Data-Driven Doa Estimation. In Proceedings of the ICASSP 2022—2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Singapore, 23–27 May 2022; pp. 3598–3602.
36. Yuan, Y.; Wu, S.; Ma, Y.; Huang, L.; Yuan, N. KR Product and Sparse Prior Based CNN Estimator for 2-D DOA Estimation. *AEU—Int. J. Electron. Commun.* **2021**, *137*, 153780. [\[CrossRef\]](#)
37. Alireza, F.; Ashkan, J.; Parisa, T.; Reza, J.; Saeed, A. Deep learning based 2D-DOA estimation using L-shaped arrays. *J. Frankl. Inst.* **2024**, *361*, 1. [\[CrossRef\]](#)
38. Sallam, T.; Wang, Q.; Attiya, A.M. High-Resolution Multiple-Source 2D DOA Estimation Using Convolutional Neural Network with Robustness to Array Imperfections. *IEEE Access* **2025**, *13*, 93115–93123. [\[CrossRef\]](#)
39. Zheng, H.; Shi, Z.; Zhou, C.; Vorobyov, S.A.; Gu, Y. Deep Tensor 2-D DOA Estimation for URA. *IEEE Trans. Signal Process.* **2024**, *72*, 4065–4080. [\[CrossRef\]](#)
40. Du, B.; Cui, W.; Ba, B.; Xu, H.; Gao, W. An L-Shaped Three-Level and Single Common Element Sparse Sensor Array for 2-D DOA Estimation. *Sensors* **2023**, *23*, 6625. [\[CrossRef\]](#) [\[PubMed\]](#)
41. Srivastava, S.; Prabhu, S.; Ramesh, S.; Pratapneni, S.; Abraham, A.; Bhandary, S.V. Visualizing the Indicators of Diabetic Retinopathy Learnt by Convolutional Neural Networks. In Proceedings of the 2017 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), Coimbatore, India, 14–16 December 2017; pp. 1–3.
42. Kim, S.; Song, W.-J.; Kim, S.-H. Infrared Variation Optimized Deep Convolutional Neural Network for Robust Automatic Ground Target Recognition. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA, 21–26 July 2017; pp. 195–202.
43. Hu, J.; Shen, L.; Sun, G. Squeeze-and-Excitation Networks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
44. He, H.; He, S.; Huang, T. License Plate Recognition Based on Three Different Neural Networks. In Proceedings of the 2022 IEEE 4th International Conference on Civil Aviation Safety and Information Technology (ICCASIT), Dali, China, 12–14 October 2022; pp. 215–220.
45. Loshchilov, I.; Hutter, F. Decoupled Weight Decay Regularization. *arXiv* **2019**, arXiv:1711.05101.
46. Smith, L.N. Cyclical Learning Rates for Training Neural Networks. In Proceedings of the 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), Santa Rosa, CA, USA, 24–31 March 2017; pp. 464–472.

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.