

Article

Radar Moving Target Detection Based on Small-Sample Transfer Learning and Attention Mechanism

Jiang Zhu ¹, Cai Wen ^{2,*}, Chongdi Duan ¹, Weiwei Wang ¹ and Xiaochao Yang ¹

¹ National Key Laboratory of Science and Technology on Space Microwave, Xi'an Institute of Space Radio Technology, Xi'an 710199, China; zhuj@cast504.com (J.Z.); duandc@cast504.com (C.D.); wangww83@cast504.com (W.W.); yangxc@cast504.com (X.Y.)

² School of Information Science and Technology, Northwest University, Xi'an 710068, China

* Correspondence: wencai@nwnu.edu.cn

Abstract: Moving target detection is one of the most important tasks of radar systems. The clutter echo received by radar is usually strong and heterogeneous when the radar works in a complex terrain environment, resulting in performance degradation in moving target detection. Utilizing prior knowledge of the clutter distribution in the space–time domain, this paper proposes a novel moving target detection network based on small-sample transfer learning and attention mechanism. The proposed network first utilizes offline data to train the feature extraction network and reduce the online training time. Meanwhile, the attention mechanism used for feature extraction is applied in the beam-Doppler domain to improve classification accuracy of targets. Then, a small amount of real-time data are applied to a small-sample transfer network to fine-tune the feature extraction network. Finally, the target detection can be realized by the fine-tuned network. Simulation experiments show that the proposed network can eliminate the influence of heterogeneous clutter on moving target detection, and the attention mechanism can improve clutter suppression under a low signal-to-noise ratio regime. The proposed network has a lower computational load compared to conventional neural networks, enabling its use in real-time applications on space-borne/airborne radars.

Keywords: radar moving target detection; deep convolutional networks; small-sample network; attention mechanism



Citation: Zhu, J.; Wen, C.; Duan, C.; Wang, W.; Yang, X. Radar Moving Target Detection Based on Small-Sample Transfer Learning and Attention Mechanism. *Remote Sens.* **2024**, *16*, 4325. <https://doi.org/10.3390/rs16224325>

Academic Editor: Chenguang Shi

Received: 30 September 2024

Revised: 28 October 2024

Accepted: 12 November 2024

Published: 20 November 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Moving target detection is an important task in spaceborne/airborne radar signal processing. However, when the radar operates in a down-looking mode, it often faces strong ground clutter that can spread in the space–time domain and submerge slow, weakly moving targets [1]. To eliminate the impact of clutter on target detection, space–time adaptive processing (STAP) was proposed in 1973 [2]. The method performs joint adaptive processing in the element-pulse domain to effectively filter out space–time coupled clutter and achieve coherent accumulation of targets. It is a key technology for radar clutter suppression and moving target detection on various moving platforms. STAP requires a certain number of independent and identically distributed (IID) training samples as support; whereas actual clutter environments are often non-homogeneous or non-stationary. The non-homogeneity of clutter mainly comes from different types of ground objects, terrain, strong isolated clutter, dense moving targets, etc. The non-stationarity of clutter mainly comes from non-side-looking arrays, nonlinear arrays, bistatic radar configurations, platform yaw, and equivalent yaw due to the rotation of the Earth [3–6]. The non-homogeneity or non-stationarity of clutter prevents radar systems from acquiring an adequate number of IID samples, thereby presenting substantial challenges to conventional STAP for detecting moving targets.

To address these issues, methods such as direct data domain (DDD) STAP [7], knowledge-aided (KA) STAP [8–10], and sparse recovery (SR) STAP [11–16] have been

proposed. These methods aim to achieve suboptimal clutter suppression performance under limited sample conditions. The DDD STAP method utilizes only the echo data from the range bin under detection, completely eliminating the issue of insufficient IID samples in non-stationary/non-uniform clutter environments. However, it is susceptible to array errors and suffers from significant aperture loss, preventing it from achieving optimal performance. Additionally, its spatial domain smoothing process limits its applicability to radars with uniform linear/planar arrays. KA STAP method employs prior information from the platform and radar system to pre-estimate the clutter covariance matrix (CCM) and pre-whiten the clutter. Nevertheless, due to internal and external unknown non-ideal factors such as antenna errors and clutter fluctuations, the estimated CCM deviates significantly from the actual values. Moreover, the problem of reasonable allocation of weight coefficients between the CCM constructed from prior knowledge and the CCM estimated by maximum likelihood has not been well resolved. SR STAP leverages the sparse distribution characteristics of clutter in the space–time domain. By imposing sparse constraints on the space–time spectral coefficients, it can reconstruct the clutter’s super-resolution space–time spectrum and calculate the corresponding CCM with a small number of observed samples. However, these methods are sensitive to array errors and typically rely on iterative optimization techniques for solution, often leading to long execution times, low efficiency, and difficulties in selecting hyperparameters, which are not conducive to practical applications. Despite these challenges, sparse recovery methods offer a new approach to enhancing STAP performance under small-sample conditions.

In the field of deep learning, moving target detection and classification mainly rely on the extraction of deep features by convolutional neural networks (CNNs). The strong generalization ability of CNNs can mitigate the impact of non-homogeneous clutter to some extent. In 2018, Liu Zhe proposed a CNN network model for moving target detection that can operate under non-homogeneous radar echo data conditions [17]. The application of autoencoder networks in radar signal processing allows for the reconstruction and simulation of radar echo signals. In 2019, Simon Wagner [18] utilized autoencoder networks to generate radar echo signals, comparing them with actual radar echo signals to locate the region of the target signal, thereby accurately detecting moving target signals. In 2020, Xiaolong Chen [19] used adversarial generative networks to suppress sea clutter, enhancing the detection performance of moving targets on the sea surface. In 2021, Keqing Duan [20] employed a CNN to reconstruct high-resolution angle-Doppler profiles, then used the obtained space–time two-dimensional clutter spectrum to reconstruct the covariance matrix. Finally, they used the reconstructed covariance matrix to calculate the STAP weights and achieve clutter suppression. Although convolutional neural networks can overcome the impact of non-homogeneous environments, their training requires a large amount of sample data, making it difficult to apply them to actual radar systems. Additionally, existing CNN-based methods require online learning of radar clutter features, which involves a high computational load and is not suitable for real-time radar processing. On the basis of clutter suppression, considering the impact of residual clutter and broadening, it is also necessary to combine with detectors such as CFAR to achieve target detection. Jian Xue [21–23] utilized a neural network and multisample percentiles to extract the shape characteristics of sea clutter, and designed adaptive and persymmetric detectors to mitigate the effects of heavy-tailed and nonhomogeneous and non-Gaussian sea clutter on target detection performance.

To overcome these limitations, this paper proposes a small-sample migration network (SSM-Net) based on deep learning. This method fully utilizes the prior knowledge of clutter distribution in the space–time domain. It first uses offline data to pretrain a feature extraction network to reduce the online training time of the deep neural network. Then, based on a small amount of collected real-time data, it fine-tunes the feature extraction network using a small-sample migration network. Finally, it uses the fine-tuned network to complete moving target detection. In terms of feature extraction, the paper also utilizes the aggregation of targets in the angle-Doppler plane and proposes a dual-channel attention mechanism

in the beam-Doppler domain to improve classification accuracy. Simulation experiments demonstrate that the network can effectively reduce the impact of non-homogeneous clutter on moving target detection and can extract the main features of moving targets under a low signal-to-clutter ratio (SCR). Compared to traditional convolutional neural networks, the proposed small-sample network requires fewer measured data samples and has lower computational cost, making real-time detection of moving targets based on deep learning possible and having great potential for practical applications in future radar systems.

2. Radar Signal Model and Problem Formulation of Moving Target Detection

2.1. Radar Signal Model

Assume the radar is mounted on a platform flying at a constant horizontal speed of v at an altitude H . A coordinate axis is established with the Earth's center as the origin. The direction of the platform's flight is the X-axis, as illustrated in Figure 1. Assume the airborne radar array consists of N elements placed in a uniform linear array with a side-looking configuration. θ and φ represent the azimuth and elevation angles of the antenna beam's direction for the current clutter patch, respectively. The pulse repetition interval is T_r , and the pulse repetition frequency is $f_r = 1/T_r$. During the coherent processing interval, M pulses are transmitted.

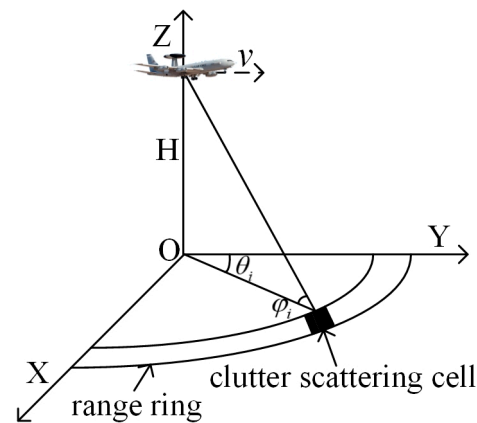


Figure 1. Radar operational geometry.

Assume there are N_c clutter cells uniformly distributed in the azimuth within the range cell to be detected, as shown in Figure 1. The normalized Doppler frequency of each clutter cell is given by

$$f_d(\varphi, \theta) = \frac{2V \cos \varphi \cos \theta}{\lambda f_r}, \quad (1)$$

where λ is the radar operating wavelength, and f_r is the pulse repetition frequency. The normalized spatial frequency of the clutter block is

$$f_s(\varphi, \theta) = \frac{d \cos \varphi \cos \theta}{\lambda}. \quad (2)$$

where d is the element spacing. For a uniform linear array, the temporal and spatial steering vectors for each clutter cell are

$$\begin{cases} \mathbf{w}_d(f_d(\theta, \varphi)) = [1, e^{j\pi f_d(\theta, \varphi)}, \dots, e^{j\pi(M-1)f_d(\theta, \varphi)}]^T, \\ \mathbf{w}_s(f_s(\theta, \varphi)) = [1, e^{j\pi f_s(\theta, \varphi)}, \dots, e^{j\pi(N-1)f_s(\theta, \varphi)}]^T. \end{cases} \quad (3)$$

Assuming each clutter cell is statistically independent, and considering array errors and clutter fluctuations, the space–time snapshot data for each range cell can be expressed as the superposition of the echo signals from clutter cells at different ambiguous ranges [24]:

$$\mathbf{x}_c = \sum_{i=1}^{N_c} \alpha_i [\gamma \odot \mathbf{w}_d(\theta_i, \varphi_i)] \otimes [\mathbf{e}(\theta_i, \varphi_i) \odot \mathbf{w}_s(\theta_i, \varphi_i)], \quad (4)$$

where α_i represents the echo amplitude of the i -th clutter cell, which is related to the radar transmit power, antenna gain, clutter cell reflectivity, terrain scattering coefficient, etc. $\gamma = [\gamma_1, \gamma_2, \dots, \gamma_M]^T$ represents the fluctuating component between coherent pulses; $\mathbf{e}(\theta, \varphi) = [e_1(\theta, \varphi), e_2(\theta, \varphi), \dots, e_N(\theta, \varphi)]^T$ represents the array error component; $\odot$ and $\otimes$ denote the Hadamard product and Kronecker product between matrices, respectively.

Similarly, the echo signal of a moving target can be expressed as

$$\mathbf{x}_t = \alpha_t \mathbf{w}_d(f_t) \otimes [\mathbf{e}(\theta_t, \varphi_t) \odot \mathbf{w}_s(\theta_t, \varphi_t)], \quad (5)$$

where α_t represents the complex amplitude of the signal; and f_t , θ_t , and φ_t represent the normalized Doppler frequency, azimuth angle, and elevation angle of the moving target, respectively.

2.2. Moving Target Detection

Airborne radar moving target detection can be described as a binary hypothesis testing problem:

$$\begin{cases} H_0 : \mathbf{y} = \mathbf{x}_c + \mathbf{n}, \\ H_1 : \mathbf{y} = \mathbf{x}_t + \mathbf{x}_c + \mathbf{n}, \end{cases} \quad (6)$$

where $\mathbf{y}$ represents the received echo, $\mathbf{n}$ represents the Gaussian white noise component, H_0 stands for scene echoes with targets, and H_1 stands for scene echoes without targets. This problem is a typical target detection problem in clutter.

Traditional radar moving target detection methods can generally be divided into two categories: the first category directly uses multi-channel adaptive detectors for target detection, such as the generalized likelihood ratio (GLRT) and adaptive matched filter (AMF); the second category first filters and then detects, i.e., uses the traditional STAP clutter suppression followed by constant false alarm rate (CFAR) detection [25]. The performance of these methods largely depends on the estimation accuracy of the clutter statistics (i.e., the clutter covariance matrix). However, estimating the clutter covariance matrix requires a large number of IID samples, which is difficult to meet in actual heterogeneous environments. Therefore, the detection performance of traditional methods can sharply decline in heterogeneous clutter environments.

In the field of deep learning, the training and learning of convolutional networks do not depend on changes in data characteristics, making deep learning an area of interest in radar applications in recent years. Current deep learning-based airborne radar target detection methods have shown good generalization ability to heterogeneous clutter scenes [17]. However, it is worth noting that the training of convolutional neural networks requires a large number of data samples and an extremely long training time, making current deep learning-based airborne radar moving target detection still theoretical and difficult to apply in actual systems. To overcome the aforementioned limitations, this paper proposes a few-shot radar moving target detection network based on pre-training and transfer learning. We will introduce the components of this network in detail.

3. Structure of Small-Sample Moving Target Detection Network

The small-sample migration network (SSM-Net) proposed in this paper is based on space-borne/airborne radar prior knowledge. According to (1) and (2), the distribution of clutter in the radar's space–time two-dimensional plane is coupled. The specific distribution trajectory is determined by the radar system, platform attitude, platform height, platform

velocity, and observation scenario. Therefore, we can obtain a large number of training samples through offline simulation based on prior knowledge of the system, platform, and scenario parameters. Of course, in the actual radar working environment, due to factors such as platform attitude, platform yaw, and array amplitude-phase errors, the actual collected clutter may have some deviations from the simulated clutter. This paper uses simulated clutter and actual clutter to train and correct the network, respectively, and finally uses SSM-Net to realize the detection of radar moving targets.

The training process of the small-sample migration network model consists of two parts: the first step is to train the feature extraction network based on simulated radar echo data, which is offline network training; the second step is to fine-tune the feature extraction network using a small-sample migration network with a small amount of collected real-time data, which is online training; and finally, the fine-tuned network is used to complete the detection of moving targets. It should be noted that the simulated clutter echoes need to be augmented based on factors such as the type of detection targets and interference environments in order to provide sufficient samples for network model training.

The network model is composed of a feature extraction network (FENet) that combines convolutional neural networks with attention mechanisms, and an improved Siamese network [26–30]. For a detailed description of each module of the network, refer to the following text. The overall network model is illustrated in Figure 2.

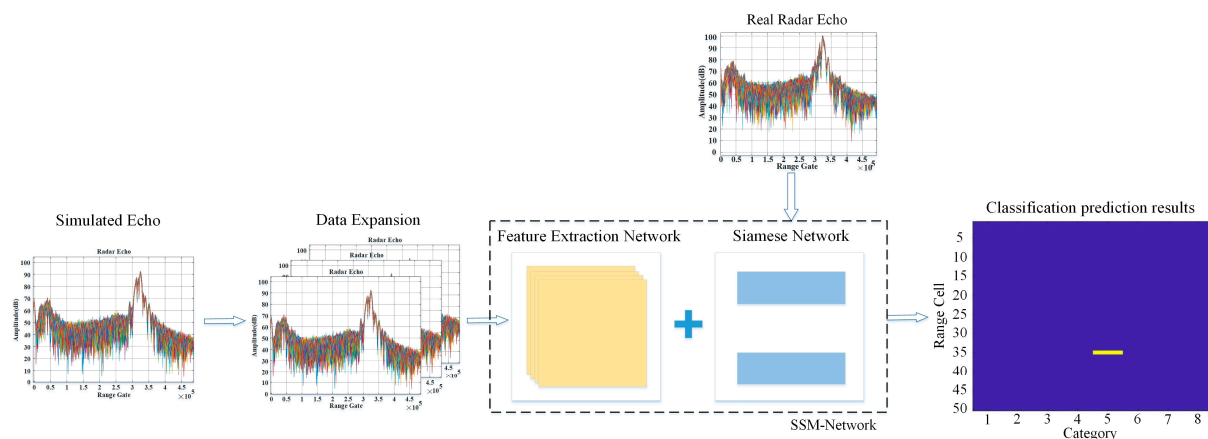


Figure 2. Small-sample network architecture diagram.

3.1. Feature Extraction Network Model Based on Attention Mechanism

Drawing on traditional CNN networks and combining the characteristics of radar echo data, this paper proposes a deep learning feature extraction network (FENet) model based on attention mechanism. To better extract the data features of clutter and moving target signals, this paper converts the simulated radar echo signal to the beam-Doppler domain and divides the data into real and imaginary parts for input to FENet. Since targets are point-distributed in the angle-Doppler plane, while clutter is strip-distributed, they often occupy only a few cells in the beam-Doppler domain. Therefore, another benefit of deep learning in the beam-Doppler domain is that it facilitates the use of attention mechanisms to enhance feature extraction performance and efficiency. As shown in Figure 3, the FENet model proposed in this paper consists of 4 convolutional layers, 2 pooling layers, 4 attention modules, and 2 fully connected layers. The input is the radar's three-dimensional echo signal in the beam-Doppler domain, and the output is the category to which the moving target signal belongs.

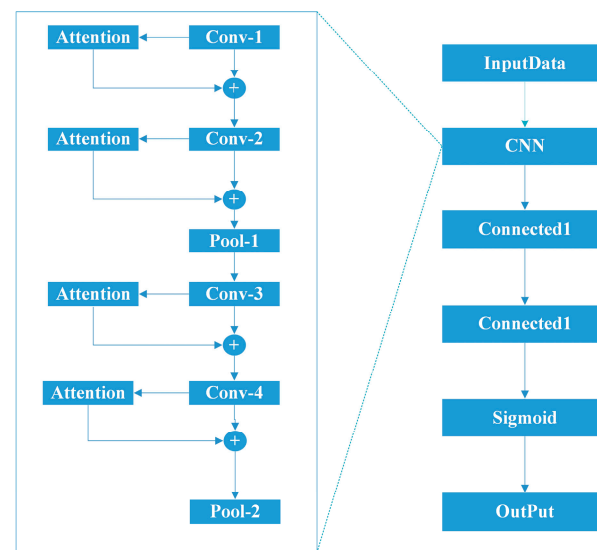


Figure 3. Feature extraction network diagram.

The convolutional layer is used for feature extraction, mainly extracting the main features of the moving target from the input data:

$$F^c = \text{Relu}(W_c * Y + b_c) \quad (7)$$

In the network, the convolutional layers utilize the Rectified Linear Unit (ReLU) activation function and zero-padding operations. Let F^c represent the feature matrix after the convolutional network; and W^c denote the convolution kernel with dimensions $f \times f \times c$, where f represents the length and width of the convolution kernel, and c is the number of convolution kernels, i.e., the number of output feature channels. $*$ denotes the convolution operation, b_c is the bias vector, and the output feature dimensions are $h \times w \times c$, where h and w are the length and width of the output data, respectively. Their values are related to the convolution stride s_c , and the specific relationship is as follows:

$$\begin{aligned} h &= \lceil M/s_c \rceil \\ w &= \lceil N/s_c \rceil \end{aligned} \quad (8)$$

where $\lceil \cdot \rceil$ represents the ceiling operation, which rounds up to the nearest integer.

In the field of radar signal processing, distinguishing between targets and clutter is a core challenge. Targets typically appear as point distributions on the angle-Doppler plane, while clutter exhibits strip-like patterns. To enhance the training efficiency of the pre-trained network model, as well as to improve the detection capability for targets and the suppression of clutter, this paper introduces an attention module for weighted processing of the features extracted by the convolutional network.

The attention module used in this paper is based on the prior distribution characteristics of targets and clutter for weight extraction, specifically employing the convolutional block attention module (CBAM) [30]. CBAM consists of two parts: the spatial attention module and the channel attention module. When processing the input radar signal, the spatial dimension refers to the beam-Doppler two-dimensional space of the data, while the channel dimension corresponds to the real and imaginary parts of the data. CBAM can dynamically adjust the focus on channel and spatial information according to the learning process of the network, and the output-normalized attention parameters can, to some extent, achieve the suppression of clutter and the enhancement of moving targets. In this way, CBAM not only accelerates the training speed of the model but also improves the performance of the model in actual complex environments, enabling it to detect moving

targets more accurately. This mechanism is crucial for enhancing the performance of the few-shot radar moving target detection network.

In the CBAM, the channel attention module is illustrated in Figure 4. Firstly, global average pooling and global maximum pooling are performed on the convolutional feature maps along the channel dimension, resulting in two feature maps of size $1 \times 1 \times c$, where c is the number of channels.

$$\begin{cases} F_{avg}^c = \text{mean}(F^c) \\ F_{max}^c = \text{max}(F^c) \end{cases} \quad (9)$$

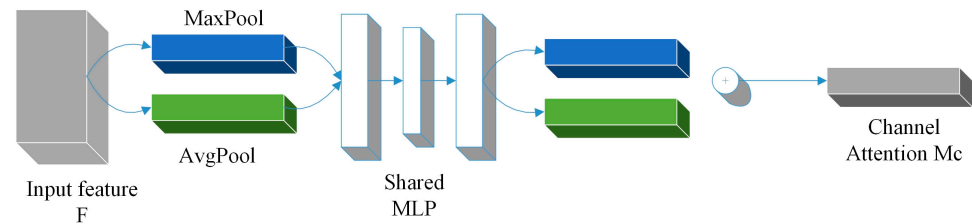


Figure 4. Channel attention module network model.

In the equation, F_{avg}^c represents the feature vector obtained after the average pooling operation, and F_{max}^c denotes the feature vector obtained from the maximum pooling operation. $\text{max}()$ and $\text{mean}()$ represent the maximum pooling operation and the average pooling operation, respectively.

The obtained feature vectors F_{avg}^c and F_{max}^c are then fed into a two-layer neural network, MLP (multi-layer perceptron), separately. The size of the feature matrix output by the MLP is the same as the size of the input feature matrix. The activation function used in the network is *Relu*. This process results in the updated feature vectors F_{avg}' and F_{max}' .

$$\begin{cases} F_{avg}' = \text{Relu}(W_1(W_0(F_{avg}^c) + b_0) + b_1) \\ F_{max}' = \text{Relu}(W_1(W_0(F_{max}^c) + b_0) + b_1) \end{cases} \quad (10)$$

In the equation, F_{avg}' represents the feature parameters extracted by the neural network from the average pooling; F_{max}' denotes the feature parameters extracted by the neural network from the maximum pooling.

The feature vectors F_{avg}' and F_{max}' are summed together, and then normalized using the sigmoid function to generate the final channel attention feature vector $M_c(F)$.

$$M_c(F) = \text{Sigmoid}(F_{avg}' + F_{max}') \quad (11)$$

The channel attention feature vector $M_c(F)$ represents the weight of each channel in the input data, indicating the importance or contribution of each channel to the overall feature representation.

In the CBAM, the spatial attention module, as illustrated in Figure 5, concatenates the feature vectors F_{max}^c and F_{avg}^c to perform a convolution operation on them. After this, the resulting feature map is subjected to a global normalization of weights using the sigmoid function, which produces the updated weight vector matrix $M_s(F)$.

$$M_s(F) = \text{Softmax}\left(f^{s \times s}\left(\left[F_{avg}^c; F_{max}^c\right]\right)\right) \quad (12)$$

where $[\cdot]$ represents the concatenation operation, $f^{s \times s}(\cdot)$ denotes the convolution operation, s is the size of the convolution kernel, and $\text{Softmax}()$ serves as a classification identifier. The weights of the spatial attention module are primarily used to weight the convolutional network for individual channels.

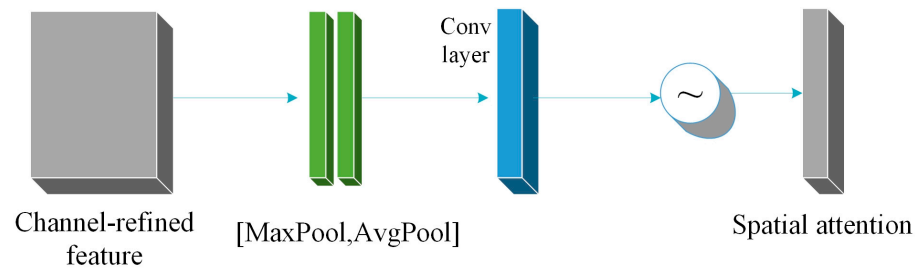


Figure 5. Spatial attention module network model.

The pooling layer is used for data dimensionality reduction, compressing the features after convolution:

$$F^P = \gamma^P(F^c; k_p, s_p) \quad (13)$$

where F^P is the feature matrix after pooling, γ^P represents the down-sampling operator, k_p is the window length of the down-sampling operator, and s_p is the down-sampling step size. After pooling, the dimension of the feature matrix is reduced, which not only reduces the number of parameters for the subsequent convolution but also prevents overfitting.

A fully connected layer flattens the feature matrix extracted by the convolution operation and simultaneously compresses the flattened feature values:

$$y^f = \gamma^f(\langle W_f, \text{vec}(F^P) \rangle + b_f) \quad (14)$$

where y^f is the output feature of the fully connected layer; $\text{vec}(\cdot)$ denotes the vectorization of a matrix, which transforms a two-dimensional feature matrix into a vector; $\langle \cdot \rangle$ represents the inner product of two vectors; W_f is the weight vector; b_f is the bias vector; and γ^f represents the activation function. In the FENet network, the first fully connected layer uses the tanh activation function, and the second layer uses the sigmoid activation function.

The final recognition layer employs a softmax classifier, which determines the probability of the output data from the fully connected layer belonging to the k -th class target as follows:

$$y_k = p(k|Y) = \frac{e^{(\langle w_k, y^f \rangle + b_k)}}{\sum_{k=0,1,\dots,K-1} e^{(\langle w_k, y^f \rangle + b_k)}} \quad (15)$$

The prediction of FENet is the probability values of multiple categories. To achieve better convergence, the loss function used in the experiment is mean squared error, and its specific mathematical expression is

$$Loss(\Theta_{FEN}) = \frac{1}{M_{total}} \sum_{m=1}^{M_{total}} \left(\sum_{k=0}^K |y_k - I_k|^2 \right) \quad (16)$$

where M_{total} represents the total number of samples during batch training, and I_k denotes the classification label vector corresponding to the query sample. After constructing the FENet network model, it is necessary to solve for the parameters of the network model using training data. Let Θ_{FEN} denote the set of all parameters in the FENet model, then the training process of the model is described as follows:

$$\Theta_{FEN} = \text{argmin } Loss(\Theta_{FEN}) \quad (17)$$

3.2. Few-Shot Transfer Learning Network Model

Considering that there are often deviations between the offline training data and the real-time data collected by radar due to non-ideal factors such as array errors and platform yaw, directly applying a feature extraction network trained on offline data to real-time airborne radar data usually cannot achieve satisfactory moving target detection effects. On

the other hand, if real-time data collected by airborne radar are directly used for network training, the excessively long training time is not sustainable for airborne radar real-time processing systems. Therefore, this paper proposes a few-shot network based on the twin network model [31], hoping to realize moving target detection with the assistance of a small number of real measured data samples. In order to achieve accurate detection of moving targets with a small number of data samples, the proposed few-shot network model integrates the feature extraction network and the twin network. Using the idea of transfer learning, and combining with the real-time echo data collected, the pretrained FENet model is fine-tuned to enable the adjusted network model to generalize non-ideal factors such as array errors, and finally achieve satisfactory prediction accuracy.

The few-shot network model is shown in Figure 6, where $f_1(\theta)$ and $f_2(\theta)$ are both well-trained FENet models. The training process of the few-shot network is as follows: the network extracts features of the support dataset; the network extracts features of the query dataset; then the similarity between the two is calculated, and the loss function is constructed by comparing with the sample labels, thus completing the fine-tuning training of the network.

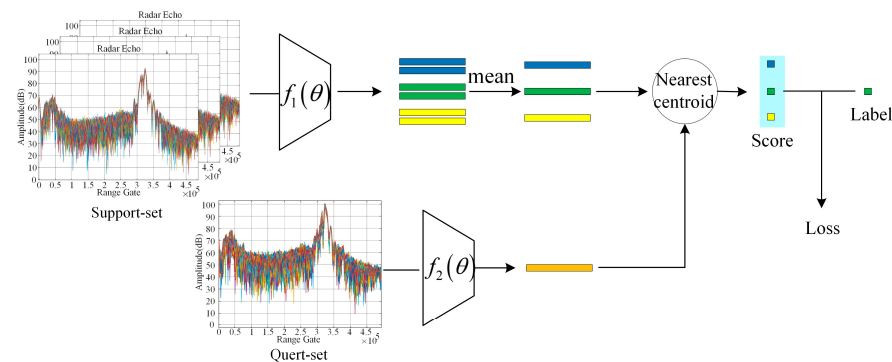


Figure 6. Few-shot transfer learning network model.

In the few-shot network, when calculating the similarity function of feature vectors, we use the nearest centroid cosine similarity function. The dataset used during the training of the few-shot network includes the support dataset and the query dataset. To better distinguish which class the query sample belongs to, we can regard this situation as a classification of clustering and use the nearest centroid method to determine which class the training sample belongs to. The mathematical expression of the centroid of each class is

$$w_k = \frac{1}{|A_k|} \sum_{x \in A_k} f_{\Theta_{SSM}}(Y) \quad (18)$$

In the above equation, A_k represents the samples belonging to class k in the support dataset. In the training of the few-shot network, the probability of the sample Y belonging to class k is characterized by the cosine similarity between the feature vector of sample Y and the centroid of class k , denoted as y_{k-ssm} , with the mathematical expression below:

$$y_{k-ssm} = p(k|Y) = \frac{\exp(\tau \cdot \langle f_{\theta}(Y), w_k \rangle)}{\sum_{k=0,1,\dots,K-1} \exp(\tau \cdot \langle f_{\theta}(Y), w_k \rangle)} \quad (19)$$

In the above equation, $\langle \cdot, \cdot \rangle$ represents cosine similarity, and τ is a learnable parameter added to scale the value of cosine similarity. τ primarily scales the value of the cosine similarity and needs to be adjusted through cross-validation. The adjustment range is from 0.05 to 0.2, and it is generally set to 0.1.

After the few-shot correction, the network outputs a probability vector of the class to which the sample belongs. The sample labels are also similar to those in the FENet network. Therefore, the loss function of the correction network still uses the mean squared error function.

$$Loss(\Theta_{SSM}) = \frac{1}{M_{total-SSM}} \sum_{m=1}^{M_{total-SSM}} \left(\sum_{k=0}^K |y_{k-ssm} - I_k|^2 \right) \quad (20)$$

where $M_{total-SSM}$ represents the total number of samples during batch training, and I_k denotes the classification label vector corresponding to the query sample. After constructing the SSM-Net network model, it is necessary to solve for the parameters of the network model using training data. Let Θ_{SSM} denote the parameter set in the SSM-Net model, then the training process of the model is described as follows:

$$\Theta_{SSM} = \operatorname{argmin} Loss(\Theta_{SSM}) \quad (21)$$

Once the parameters of the network have been updated with real small-sample data obtained through radar, the SSM-Net network has adapted to the real radar scene echoes and system spatiotemporal errors. This network can be directly applied to the real radar echoes for moving target detection, obtaining detection results of moving targets at different distances and speeds.

4. Construction of the Dataset

4.1. Clutter Data Generation

To a certain extent, the more training samples there are, the better the training of the network model for convolutional neural networks. To address this issue, this paper generates the training dataset required for FENet through computer simulation based on the high-fidelity airborne radar clutter modeling and simulation method described in Ref [24]. The radar system parameters used during the clutter simulation process are listed in Table 1.

Table 1. Basic parameters of few-shot simulation data.

Parameter	Value
Number of elements in pitch	1
Number of elements in azimuth	16
Number of pulses	64
Pulse repetition frequency	4000 Hz
Element spacing	0.25 m
Signal-to-clutter-plus-noise ratio	40 dB
Wavelength	0.5 m
Carrier aircraft altitude	4000 m
Carrier aircraft speed	120 m/s

Based on the compliance relationship between the clutter spatial incident cone angle and the Doppler frequency, the clutter space–time distribution ridge line can be accurately calculated. However, in actual working conditions, the collected clutter may have errors. For example, array errors, channel errors, and wind-induced internal movement of the clutter can cause the clutter to become decorrelated in the spatial domain and the temporal domain, respectively, resulting in the clutter power spectrum spreading on the space–time plane. We also consider these factors in the actual simulation, which can be described as follows:

1. Array errors and finite bandwidth channel errors can both be represented as spatial weighting of the array signal vector. The weighting of the radar echo signal can be represented as

$$\mathbf{t}_s = [\eta_1 e^{j\omega_1}, \eta_2 e^{j\omega_2}, \dots, \eta_M e^{j\omega_M}]^T \odot [\varepsilon_1 e^{jv_1}, \varepsilon_2 e^{jv_2}, \dots, \varepsilon_M e^{jv_M}]^T \quad (22)$$

where η and ω represent the amplitude and phase errors on the array element, and ε and v represent the amplitude and phase errors on the channel. Generally, η and ε are in the vicinity of 1, and ω and v are within 5° .

2. The internal movement of the clutter is related to wind speed and carrier frequency, specifically expressed as vector weighting in the temporal domain of the clutter. The mathematical expression is

$$\mathbf{t}_e = \left(\frac{r}{r+1} + \frac{1}{r+1} \frac{(\beta\lambda)^2}{(\beta\lambda)^2 + (4\pi T_r)^2} \right)^{\frac{1}{2}} \boldsymbol{\zeta} \quad (23)$$

where r is a function of wind speed and carrier frequency, β is a parameter related to wind speed, and $\boldsymbol{\zeta}$ is Gaussian white noise.

4.2. Data Expansion

The training of the FENet model requires a large dataset. To better generate the corresponding training and test datasets, we use the simulated radar echo data as the background clutter data and add moving target signals with different powers to the clutter data to generate the datasets we need for training. During the dataset expansion process, we use the clutter power, target power, and the range cell where the target is located as variable factors to expand the dataset. Assuming that the 33rd range cell containing the moving target is considered as the cell under test, and there are 5 guard cells on both sides of the cell under test, the range gates where the training data are selected are located outside the guard cells. For each training, we select 8 range gates, and the range gates selected for the training data are shown in Figure 7.

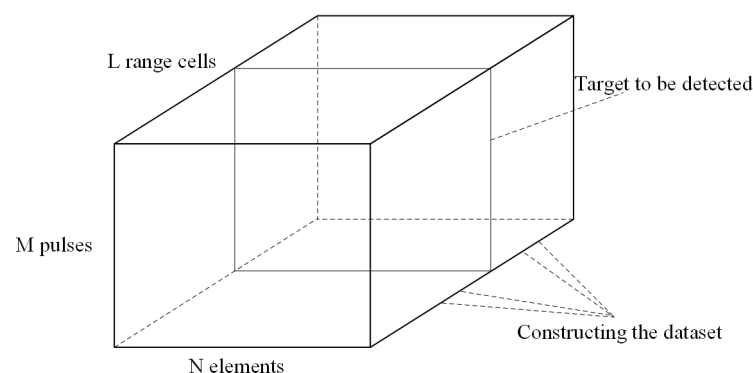


Figure 7. Schematic diagram of data selection and expansion.

To cover a wider range of actual scenarios, this paper considers expanding the echo signals in the simulation by addressing the following three situations. Certainly, the following three data models were extended based on typical issues encountered in target detection, but they do not encompass all possible situations, such as actively deployed deceptive jamming. If there are new data extension and application scenario requirements, additional data models can be incorporated into the training samples to enhance the algorithm's adaptability to these scenarios.

1. On top of the original simulated clutter, adding moving target signals with different power levels for expansion.
2. To simulate the yaw of the radar platform during flight and the equivalent yaw effect caused by the Earth's rotation under satellite platforms, the yaw factor must be considered when simulating clutter. Here, the platform yaw angles are set to 3° , 5° , and 10° , and the simulated clutter data are expanded by adding moving target signals with different power levels.
3. In actual radar echoes, other range gates outside the detection range gate may contain interference signals with different power levels. To simulate the interference process, a moving target signal with a power level different from the target is randomly added to the selected training dataset as an interference signal, and then the simulated clutter data are expanded by adding moving target signals with different power levels.

The datasets resulting from the three different background expansions mentioned above correspond to the first type of dataset (one), the second type of dataset (two), and the third type of dataset (three). To satisfy the training requirements of FENet, this paper expands the simulated echo data with three different clutter backgrounds. After expansion, each dataset reaches a size of $L_R \times M \times N \times S_{pan}$, where L_R is the number of range gates in the constructed dataset, and S_{pan} is the expansion multiple.

The expansion of moving target signals is divided into a total of K classes, with the 0th class being pure clutter signals, meaning no moving target signals are added. For the classes where targets exist, each class corresponds to the normalized Doppler frequency of the moving target as follows:

$$f_{t,k} = -\frac{1}{2} + \frac{k-1}{K-1}, k=1, 2, \dots, K-1 \quad (24)$$

In moving target detection, it is assumed that the target is located at the center of the radar's beam, and its spatial frequency is fixed. Only the extended target data corresponding to different velocities need to be generated. The extended training data $Y_{k,h}$ is as follows:

$$Y_{k,h} = x_{t,k,h} + x_c + n \quad (25)$$

where $x_{t,k,h}$ represents the extended moving target data with a normalized Doppler frequency of w_k and an amplitude of $\alpha_{t,h}$, and the specific expression is as follows:

$$x_{t,k,h} = \alpha_{t,h} w_d(f_{t,k}) \otimes [e(\theta_t, \varphi_t) \odot w_s(\theta_t, \varphi_t)] \quad (26)$$

The FENet dataset is generated based on prior knowledge of radar systems through simulation, while the dataset for the few-shot network is composed of real radar echo data. Real radar echo data inherently contain unknown spatiotemporal errors, which distinguishes them from simulated echo data and is a limitation that cannot be fully replicated in simulations.

When constructing the dataset for training SSM-Net, real radar echo data are employed. The query dataset is expanded using L_R range gate data, with different moving target signals added across these range gates. In future applications, the number of moving target signals added to the dataset can be varied as per experimental requirements. For the support dataset, we continue to use the dataset from the feature extraction network, and the batch training method is illustrated in Figure 6. For each class, two samples are selected, resulting in a total of 16 data samples that form the support dataset. The validation dataset for the small-sample network consists of all range gate data from the actual radar signals.

5. Experimental Validation

This section validates the performance of the proposed SSM-Net method through simulation experiments. We have obtained echo data from 500 range gates through simulation, and the dataset expansion method and parameters are shown in Section 4. The

dataset is divided into training and test datasets at an 8:2 ratio. During the training of the feature extraction network model, when the batch training is a multiple of 20, the prediction accuracy of the test dataset is output. Training is terminated and the network model parameters are saved when the training accuracy of the test dataset has an error of no more than 0.01% or the batch training reaches 200 times.

The input dimension size to the network during model training is $batch_size \times 16 \times 64 \times 2$, where $batch_size$ is the size of the data batch during network training, and the label is the normalized Doppler frequency of the moving target corresponding to each sample, with a total of K categories, including the moving target of the actual radar echo. The optimization strategy used during network training is Adam, with $batch_size$ set to 404 and a learning rate of 0.001. The few-shot network training uses a dataset with a batch size of 1, an optimization strategy of Adam, a learning rate of 0.005, and epochs set to 3.

5.1. Training of the Feature Extraction Network

Although the training of the FENet model is based on pre-training with simulated clutter, it is crucial for the performance of target detection. The fine-tuning of the few-shot network also relies on the training performed with FENet. The expansion parameters for the training dataset are shown in Table 2.

Table 2. Dataset expansion parameters.

Parameter	Value
Number of range gates	8
Signal-to-clutter-plus-noise ratio	−15 dB to 20 dB
Number of moving targets	8
Expansion factor	101

The training accuracy and loss function of FENet after 200 batches of training are shown in Table 3. To achieve a one-to-one correspondence between the dataset and the trained FENet model, FENet trained with the one, two, and three datasets is used as the training dataset, and the trained network models are saved as One-FENet, Two-FENet, and Three-FENet.

Table 3. FENet training results.

Data	Loss	Training Accuracy
One	0.0975	0.9864
	0.1152	0.9612
Two	0.1264	0.9654
	0.1864	0.9432
Three	0.1717	0.9502
	0.2017	0.9378

To test the actual test accuracy of the feature extraction network model, this paper constructs three validation echo datasets with different clutter backgrounds. At the 35th range gate of the echo, a moving target signal of the fifth class is added, and the signal-clutter-to-noise ratio (SCNR) in the echo signal is 10 dB. The validated echo data are input into the trained One-FENet, Two-FENet, and Three-FENet network models, and the moving target prediction results are shown in Figure 8. All three network models can detect the moving target signal well.

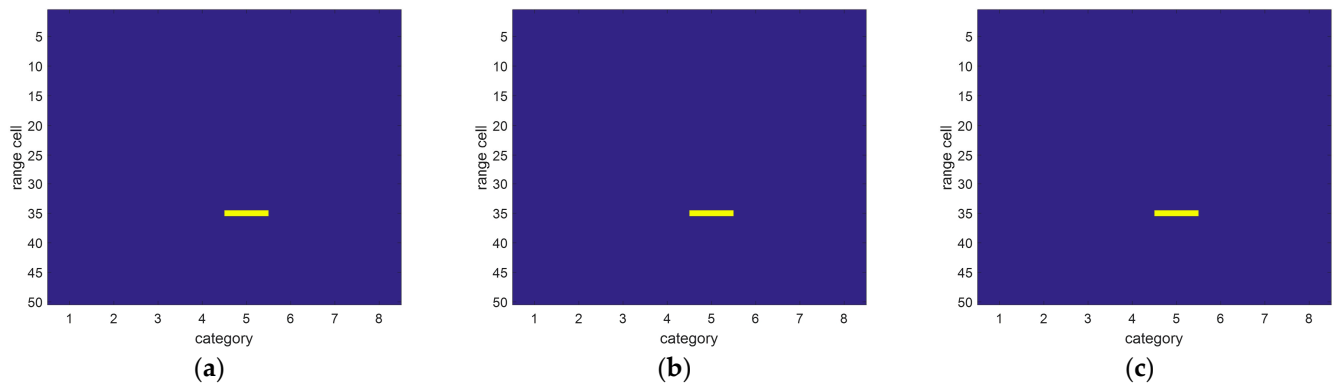


Figure 8. Prediction results of validation dataset: (a) One-FENet model detection results; (b) Two-FENet model detection results; (c) Three-FENet model detection results.

In the airborne radar moving target detection system, the probability of detection of moving targets under different SCNR conditions is the focus of experimental research and an important indicator to measure the performance of an airborne radar system. To verify the impact of adding attention mechanisms to the feature extraction network, a comparative experiment is constructed, comparing the feature extraction network with the feature extraction network without attention mechanisms (2C-CNN). As shown in Figure 9, under high SCNR conditions, FENet and 2C-CNN have similar prediction accuracy on the validation dataset and can both reach about 92. However, under low-SCNR conditions, especially at -25 dB to -12 dB, the detection performance of the proposed FENet is about 5% higher on average than the prediction accuracy of the traditional 2C-CNN dataset, further demonstrating the superiority and feasibility of the FENet model after the introduction of attention mechanisms.

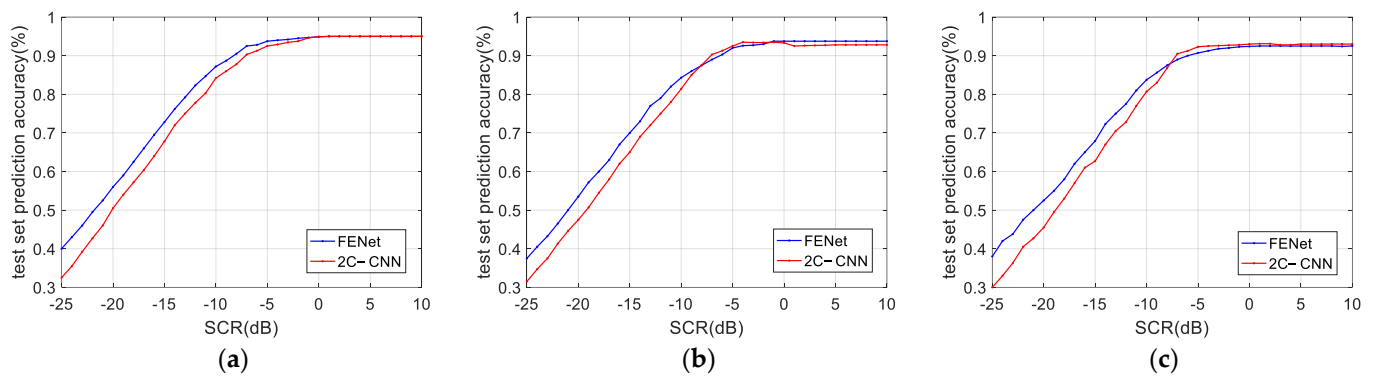


Figure 9. Comparison experiment under low-SCR conditions: (a) Comparison experiment on one dataset; (b) comparison experiment on two dataset; (c) comparison experiment on three dataset.

To further validate the impact of attention mechanisms on the internal structure of the feature extraction network, three sample data from three validation datasets were selected, with the moving target in the sixth category. The three sample data were input into the corresponding trained network models, and the softmax function at the end of the fully connected layer in the feature extraction network was removed to observe the impact of the attention mechanism on the sample prediction classification probability. In Figure 10a–c, the final classification prediction probabilities of the three samples are shown. The comparison results of each group show that the prediction probability of the moving target of the sixth category reaches 62% when attention mechanisms are added, which is about 10% higher than without attention mechanisms.

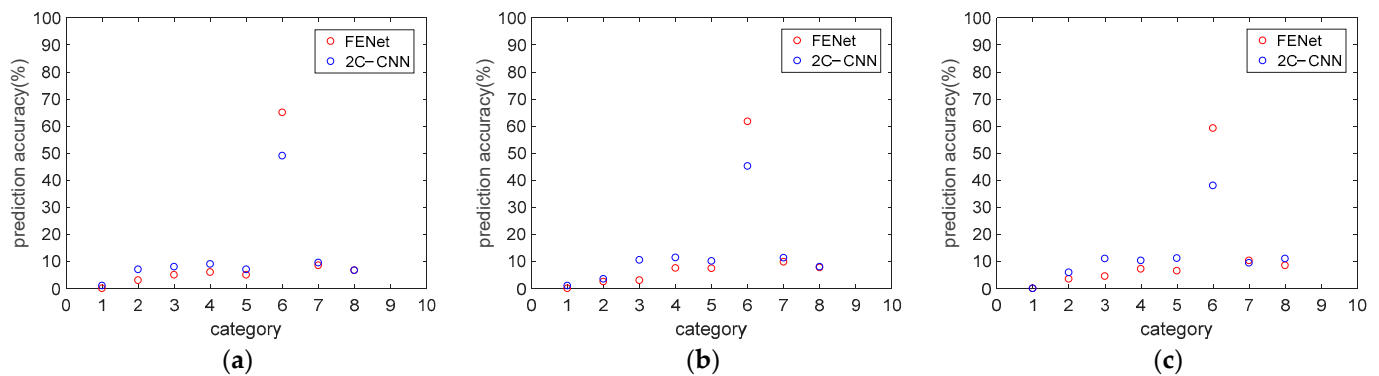


Figure 10. The impact of attention mechanism on classification prediction: (a) Classification prediction probability for data one; (b) classification prediction probability for data two; (c) classification prediction probability for data three.

5.2. Training of the Few-Shot Transfer Network

In the few-shot network, the number of samples in the few-shot dataset can also affect the prediction results. To find the optimal number of samples per class in the few-shot dataset, we constructed 1-shot, 5-shot, and 8-shot few-shot datasets, corresponding to 8, 40, and 64 sample data in the training dataset, respectively. Then, we conducted experimental validations on the trained One-FENet model. The results, as shown in Table 4, indicate that when there are five samples per class, the prediction accuracy of the validation data is already close to 94%. Considering the difficulty of obtaining actual radar samples, we chose the 5-shot dataset for subsequent validation experiments.

Table 4. The impact of sample size on the prediction accuracy of SSM-Net.

Data Sample	Loss	Prediction Accuracy
1-shot	0.5346	0.871
5-shot	0.3369	0.938
8-shot	0.3243	0.941

In this study, we conducted comparative experiments with the traditional STAP+CFAR (STAP uses the EFA method, and CFAR uses the cell-averaging CFAR, with a CFAR detection threshold of 13 dB) and the few-shot network proposed in Ref. [31]. The data used were from the validation dataset with 500 range gates, and different power moving target signals were added to each range gate. The clutter-to-noise ratio in the background clutter was 40 dB. The experimental results are shown in Figure 11. Figure 11a is the experimental comparison diagram under the One-FENet training model; Figure 11b is the experimental comparison diagram under the Two-FENet training model; Figure 11c is the experimental comparison diagram under the Three-FENet training model. From an overall comparison of the experimental diagrams, it can be seen that when the SNR changes from -5 dB to -10 dB, the detection accuracy of the traditional STAP+CFAR method drops straight to about 0, while the prediction accuracy of SSM-Net and MB-FSNet still remains above 90%; in the comparison between the SSM-Net model and MB-FSNet model, when the SCR is from -30 dB to -25 dB, under the One-FENet and Three-FENet models, the prediction accuracy of SSM-Net is about 15% higher than that of MB-FSNet, and under the Two-FENet model, the prediction accuracy of SSM-Net is about 5% higher than that of MB-FSNet. This experiment fully demonstrates the feasibility of the network.

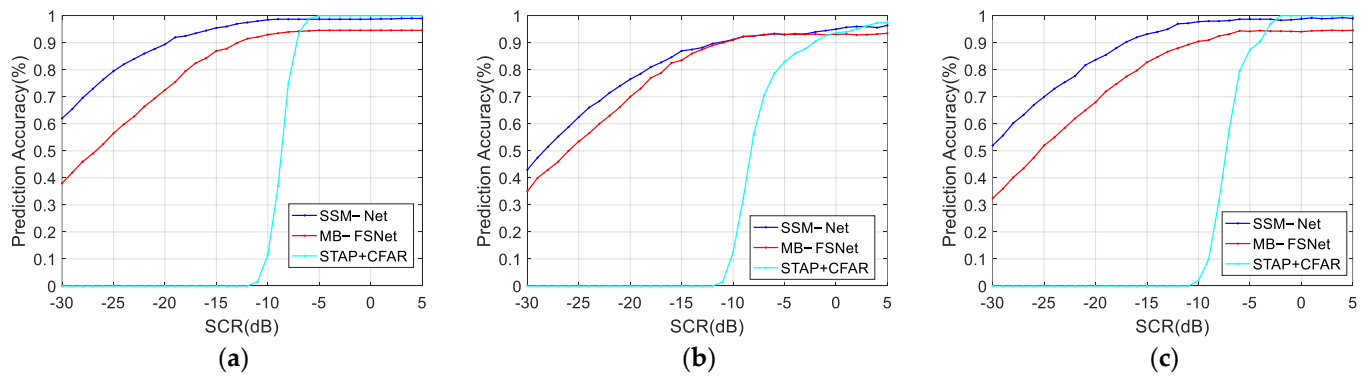


Figure 11. Comparison results with the few-shot network and traditional STAP + CFAR method: (a) Comparison experiment under One-FENet model; (b) comparison experiment under Two-FENet model; (c) comparison experiment under Three-FENet model.

To further verify the accuracy and superiority of SSM-Net in actual radar moving target detection, we used the first 50 range gate data containing moving target signals from the three types of simulated echo signals as the detection units, with the echo signal SCR at -25 dB. We used the SSM-Net and STAP+CFAR methods to detect them, and the results are shown in Figure 12. Figure 12a,c,e are the moving target detection results of SSM-Net based on One-FENet, Two-FENet, and Three-FENet, respectively. Figure 12b,d,f are the detection results of STAP+CFAR in the corresponding three types of echoes. The comparison shows that in these three cases, the SSM-Net model can accurately detect moving targets without false alarms, while the traditional STAP+CFAR method has different numbers of false alarms in each dataset, further indicating that in low-SCR situations the SSM-Net model proposed in this paper has better detection results.

5.3. Training Complexity Analysis

The spatial complexity of the convolutional neural network determines the number of model parameters. Due to the limitation of the curse of dimensionality, the more parameters a model has, the more data are required to train the model, and the more expensive the hardware required for training. For a CNN, its spatial complexity formula is

$$O\left(\sum_{l=1}^D K_l^2 \times C_{l-1} \times C_l + \sum_{l=1}^D M_l^2 \times C_l\right) \quad (27)$$

where D represents the number of convolutional layers in the neural network, C_l represents the number of output channels of the convolutional layer, C_{l-1} represents the number of output channels of the previous convolutional network, M_l represents the side length of the output feature map of the convolution kernel, and K_l represents the side length of each convolution kernel.

This paper compares SSM-Net with the FENet and 2C-CNN networks in terms of parameter quantity, and the specific results are shown in Table 5. Although the training accuracy of the SSM-Net network is slightly lower than that of FENet, the training time of SSM-Net can be neglected in practical applications, and it is significantly better than FENet in terms of training samples and computational complexity. Compared to the 2C-CNN model, when the prediction accuracy of the SSM-Net and 2C-CNN networks is similar, SSM-Net is superior to 2C-CNN in other aspects.

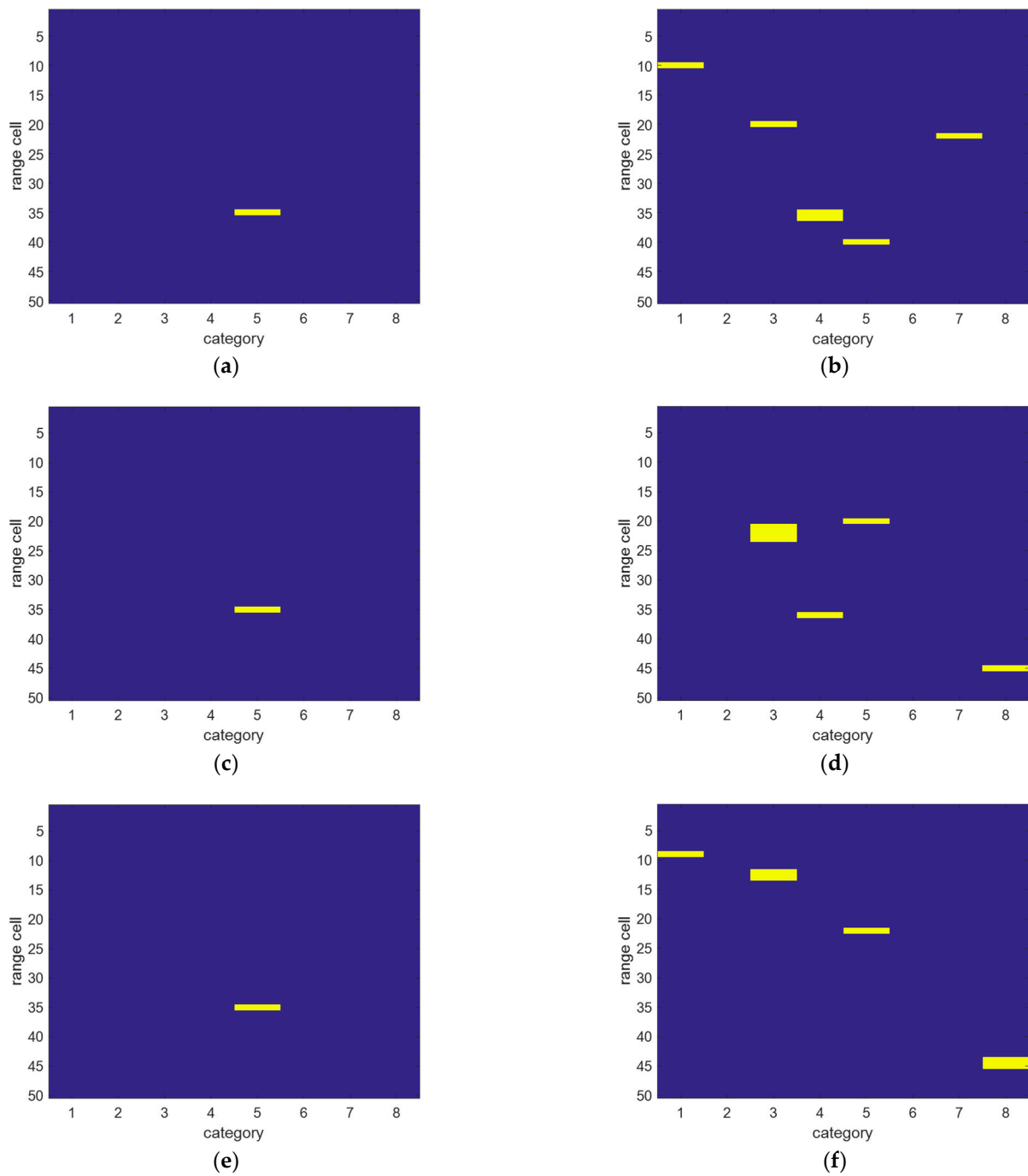


Figure 12. Moving target detection results with SSM-Net and STAP+CFAR: (a) Detection results under One-FENet model; (b) detection results for the first category using STAP + CFAR; (c) detection results under Two-FENet model; (d) detection results for the second category using STAP + CFAR; (e) detection results under Three-FENet model; (f) detection results for the third category using STAP + CFAR.

Table 5. Parameter complexity analysis.

Model	Parameters	Prediction Accuracy
SSM-Net	6.233×10^6	0.938
FENet	1.246×10^{10}	0.9672
2C-CNN	8.203×10^9	0.9297

The time complexity of the convolutional neural network determines the training/prediction time of the model. If the complexity is too high, it will lead to a large amount of time spent on model training and prediction, making it unable to quickly verify ideas and improve the model, and also unable to achieve fast prediction (affecting real-time airborne radar target detection). For a CNN, its time complexity calculation formula is

$$O\left(\sum_{l=1}^D M_l^2 \times K_l^2 \times C_{l-1} \times C_l\right) \quad (28)$$

In the SSM-Net model of this paper, the training process of the FENet model takes a long time, but the training of the FENet model is completed offline. The online operation only involves the training of the twin network part, and the required training dataset is very small, thus greatly reducing the online computation. The system used in this experiment is the Alibaba Cloud system, with 4 cores and 1 G memory. The simulation system uses MATLAB, and the network training uses the PyCharm 2018 software with Python 3.7 as the programming language. As shown in Table 6, the online time required by the SSM-Net proposed in this paper is three orders of magnitude lower than that of FENet and 2C-CNN, which is consistent with the parameter complexity analysis. Although the offline time required by SSM-Net is longer, this does not affect the prediction speed of the radar detection system. To some extent, SSM-Net overcomes the problem of the difficulty in applying convolutional neural networks to practical radar detection systems.

Table 6. Time complexity analysis.

Method	Offline Time (s)	Online Time (s)
SSM-Net	5342	2.5
FENet	0	5240
2C-CNN	0	4143

6. Conclusions

The method proposed in this paper utilizes extended pre-training samples to complete the primary training of the network in an offline state, significantly reducing the time consumed for online training. Additionally, by introducing an attention mechanism into the model, the network's ability to correctly identify targets from clutter under low signal-to-noise ratio conditions is enhanced, improving recognition accuracy. Furthermore, considering the deviation between simulated training samples and actual samples, the network model undergoes secondary correction using collected real-time data on the basis of a transfer network, which further enhances the network's adaptability in real-world environments. In terms of detection performance, the proposed method maintains 90% target prediction accuracy even when the signal-to-noise ratio is below -5 dB; this is an improvement on typical distributed detection methods (STAP+CFAR). When compared to the MB-FSNet network, the proposed method shows a performance improvement of 5% to 15% across different data types. Additionally, while there is a 3% decrease in accuracy compared to FENet, the accuracy still reaches 93%. At the same time, the computational complexity decreases by nearly a thousand times, making it promising for real-time detection on-board.

Nevertheless, this approach also has certain limitations, such as the need for prior knowledge about the radar platform and system parameters during the pre-training process of the SSM-Net model. This implies that different training sets and networks may be necessary for varying flight parameters. To tackle this challenge, we plan to explore the use of adversarial generation networks to automatically produce a large volume of training data. Additionally, considering various types of electromagnetic interference during the pre-training process is another key direction of our future research.

Author Contributions: Conceptualization, J.Z. and C.W.; methodology, J.Z.; software, J.Z.; validation, J.Z., C.W. and W.W.; data curation, J.Z. and X.Y.; writing—original draft preparation, J.Z.; writing—review and editing, C.W. and C.D.; visualization, W.W.; supervision, C.D. and X.Y.; project administration, J.Z.; funding acquisition, W.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Key Laboratory Foundation, grant number 2023-JCJQ-LB-007.

Data Availability Statement: Dataset available on request from the authors.

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

References

- Himed, B.; Soumekh, M. Synthetic aperture radar-moving target indicator processing of multi-channel airborne radar measurement data. *IEE Proc. Radar Sonar Navig.* **2006**, *153*, 532–543. [\[CrossRef\]](#)
- Roman, J.R.; Rangaswamy, M.; Davis, D.W.; Zhang, Q.; Himed, B.; Michels, J.H. Parametric adaptive matched filter for airborne radar applications. *IEEE Trans. Aerosp. Electron. Syst.* **2000**, *36*, 677–692. [\[CrossRef\]](#)
- Stoica, P.; Li, J.; Zhu, X.; Guerci, J.R. On Using a priori Knowledge in Space-Time Adaptive Processing. *IEEE Trans. Signal Process.* **2008**, *56*, 2598–2602. [\[CrossRef\]](#)
- Melvin, W.L. *Space-Time Adaptive Processing for Radar*; Academic Press Library in Signal Processing: Cambridge, MA, USA, 2014; Volume 2, pp. 595–665.
- Klemm, R. Space-Time Adaptive Processing: Principles and Applications. *Electron. Commun. Eng. J.* **1999**, *11*, 172. [\[CrossRef\]](#)
- Wang, Y.; Liu, W.; Xie, W.; Zhao, Y. Reduced-rank space-time adaptive detection for airborne radar. *Sci. China Inf. Ence* **2014**, *57*, 1–11. [\[CrossRef\]](#)
- Sarkar, T.K.; Wang, H.; Park, S.; Adve, R.; Koh, J.; Kim, K.; Zhang, Y.; Wicks, M.C.; Brown, R.D. A deterministic least-squares approach to space-time adaptive processing (STAP). *IEEE Trans. Antennas Propag.* **2009**, *49*, 91–103. [\[CrossRef\]](#)
- Melvin, W.; Wicks, M. Knowledge-based space-time adaptive processing for airborne early warning radar. *IEEE Aerosp. Electron. Syst. Mag.* **1998**, *13*, 37–42. [\[CrossRef\]](#)
- Bergin, J.S.; Teixeira, C.M.; Techau, P.M.; Guerci, J.R. Improved clutter mitigation performance using knowledge-aided space-time adaptive processing. *IEEE Trans. Aerosp. Electron. Syst.* **2006**, *42*, 997–1009. [\[CrossRef\]](#)
- Wu, Y.; Tang, J.; Peng, Y. On the Essence of Knowledge-Aided Clutter Covariance Estimate and Its Convergence. *IEEE Trans. Aerosp. Electron. Syst.* **2011**, *47*, 569–585. [\[CrossRef\]](#)
- Li, J.; Zhu, X.; Stoica, P.; Rangaswamy, M. High Resolution Angle-Doppler Imaging for MTI Radar. *IEEE Trans. Aerosp. Electron. Syst.* **2010**, *46*, 1544–1556. [\[CrossRef\]](#)
- Sun, K.; Meng, H.; Wang, Y.; Wang, X. Direct Data Domain STAP using Sparse Representation of Clutter Spectrum. *Signal Process.* **2011**, *91*, 2222–2236. [\[CrossRef\]](#)
- Sen, S. OFDM Radar Space-Time Adaptive Processing by Exploiting Spatio-Temporal Sparsity. *IEEE Trans. Signal Process.* **2013**, *61*, 118–130. [\[CrossRef\]](#)
- Yang, X.; Sun, Y.; Zeng, T.; Long, T.; Sarkar, T.K. Fast STAP Method Based on PAST with Sparse Constraint for Airborne Phased Array Radar. *IEEE Trans. Signal Process.* **2016**, *64*, 4550–4561. [\[CrossRef\]](#)
- Yang, Z.; Lamare, R.; Liu, W. Sparsity-Based STAP Design Based on Alternating Direction Method with Gain/Phase Errors. *IEEE Trans. Aerosp. Electron. Syst.* **2017**, *53*, 2756–2768.
- Duan, K.; Wang, Z.; Xie, W.; Chen, H.; Wang, Y. Sparsity-based STAP algorithm with multiple measurement vectors via sparse Bayesian learning strategy for airborne radar. *IET Signal Process.* **2017**, *11*, 544–553. [\[CrossRef\]](#)
- Liu, Z.; Ho, D.K.; Xu, X.; Yang, J. Moving target indication using deep convolutional neural network. *IEEE Access* **2018**, *6*, 65651–65660. [\[CrossRef\]](#)
- Wagner, S.; Barth, K.; Brüggewirth, S. A deep learning SAR ATR system using regularization and prioritized classes. In Proceedings of the 2017 IEEE Radar Conference (RadarConf), Seattle, WA, USA, 8–12 May 2017; pp. 772–777.
- Mou, X.; Chen, X.; Guan, J.; Dong, Y.; Liu, N. Sea clutter suppression for radar PPI images based on SCS-GAN. *IEEE Geosci. Remote Sens. Lett.* **2020**, *18*, 1886–1890. [\[CrossRef\]](#)
- Duan, K.; Chen, H.; Xie, W.; Wang, Y. Deep learning for high-resolution estimation of clutter angle-Doppler spectrum in STAP. *IET Radar Sonar Navig.* **2022**, *16*, 193–207. [\[CrossRef\]](#)
- Xue, J.; Sun, M.; Liu, J.; Xu, S.; Pan, M. Shape Parameter Estimation of K-Distributed Sea Clutter Using Neural Network and Multisample Percentile in Radar Industry. *IEEE Trans. Ind. Inform.* **2023**, *19*, 7602–7612. [\[CrossRef\]](#)
- Xue, J.; Liu, J.; Xu, S.; Pan, M. Adaptive Detection of Radar Targets in Heavy-Tailed Sea Clutter With Lognormal Texture. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–11. [\[CrossRef\]](#)
- Xue, J.; Xu, S.; Liu, J. Persymmetric Detection of Radar Targets in Nonhomogeneous and Non-Gaussian Sea Clutter. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–9.

24. Melvin, W.L.; Showman, G.A. An approach to knowledge-aided covariance estimation. *IEEE Trans. Aerosp. Electron. Syst.* **2006**, *42*, 1021–1042. [[CrossRef](#)]
25. Gandhi, P.P.; Kassam, S.A. Analysis of CFAR processors in nonhomogeneous background. *IEEE Trans. Aerosp. Electron. Syst.* **2002**, *24*, 427–445. [[CrossRef](#)]
26. Zhu, J.; Yao, G.; Zhang, G.; Li, J.; Yang, Q.; Wang, S.; Ye, S. Survey of Few Shot Learning of Deep Neural Network. *Comput. Eng. Appl.* **2021**, *57*, 22–33.
27. Chopra, S.; Hadsell, R.; LeCun, Y. Learning a similarity metric discriminatively, with application to face verification. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Diego, CA, USA, 20–25 June 2005; IEEE: Piscataway, NJ, USA, 2005; Volume 1, pp. 539–546.
28. Wetzel, S.J.; Ryczko, K.; Melko, R.G.; Tamblyn, I. Twin neural network regression. *arXiv* **2020**, arXiv:2012.14873. [[CrossRef](#)]
29. Dai, Y.; Zhang, K.; Maharjan, S.; Zhang, Y. Deep reinforcement learning for stochastic computation offloading in digital twin networks. *IEEE Trans. Ind. Inform.* **2020**, *17*, 4968–4977. [[CrossRef](#)]
30. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. CBAM: Convolutional Block Attention Module. *Eur. Conf. Comput. Vis.* **2018**, *175*, 3–19.
31. Sung, F.; Yang, Y.; Zhang, L. Learning to compare: Relation network for few-shot learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 1199–1208.

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.