A Trimodel SAR Semisupervised Recognition Method Based on Attention-Augmented Convolutional Networks

Sifan Yan, Yaotian Zhang, Fei Gao, Jinping Sun, *Member, IEEE*, Amir Hussain, and Huiyu Zhou

***Abstract*—Semisupervised learning in synthetic aperture radars (SARs) is one of the research hotspots in the field of radar image automatic target recognition. It can efficiently deal with challenging environments where there are insufficient labeled samples and large unlabeled samples in the SAR dataset. In recent years, consistency regularization methods in semisupervised learning have shown considerable improvement in recognition accuracy and efficiency. Current consistency regularization approaches suffer from two main shortcomings: first, extracting all of the relevant information in the image target is difficult owing to the inability of conventional convolutional neural networks to capture global relational information; second, the standard teacher–student regularization methodology causes confirmation biases due to the high coupling between teacher and student models. This article adopts an innovative trimodel semisupervised method based on attention-augmented convolutional networks to address the aforementioned obstacles. Specifically, we develop an attention mechanism incorporating a novel positional embedding method based on recurrent neural networks and integrate this with a standard convolutional network as a feature extractor, to improve the network's ability to extract global feature information from images. Furthermore, we address the confirmation bias problem by introducing a classmate model to the standard teacher–student structure and utilize the model to impose a weak consistency constraint designed on the student to weaken the strong coupling between the teacher and the student. Comparative experiments on the Moving and Stationary Target Acquisition and Recognition dataset show that our method outperforms state-of-the-art semisupervised methods in terms of recognition accuracy, demonstrating its potential as a new benchmark approach for the deep learning and SAR research community.**

***Index Terms*—Consistency regularization, convolutional networks, self-attention, semisupervised learning (SSL), synthetic aperture radar (SAR).**

Manuscript received 20 June 2022; revised 3 September 2022 and 12 October 2022; accepted 26 October 2022. Date of publication 31 October 2022; date of current version 14 November 2022. The work of Amir Hussain was supported by the U.K. Engineering and Physical Sciences Research Council under Grant EP/M026981/1, Grant EP/T021063/1, and Grant EP/T024917/1. The work of Huiyu Zhou was supported in part by the Royal Society Newton Advanced Fellowship under Grant NA160342 and in part by the European Union's Horizon 2020 Research and Innovation Program under Marie Sklodowska Curie Grant 720325. This work was supported in part by the National Natural Science Foundation of China under Grant 61771027 and Grant 61071139. *(Corresponding author: Fei Gao.)*

Sifan Yan, Yaotian Zhang, Fei Gao, and Jinping Sun are with the School of Electronic and Information Engineering, Beihang University, Beijing 100191, China (e-mail: siriusysf@buaa.edu.cn; zhangyaotian@buaa.edu.cn; feigao2000@163.com; sunjinping@buaa.edu.cn).

Amir Hussain is with the Cyber and Big Data Research Laboratory, Edinburgh Napier University, Edinburgh EH11 4BN, U.K. (e-mail: a.hussain@napier.ac.uk).

Huiyu Zhou is with the Department of Informatics, University of Leicester, Leicester LE1 7RH, U.K. (e-mail: hz143@leicester.ac.uk).

Digital Object Identifier 10.1109/JSTARS.2022.3218360

I. Introduction

SYNTHETIC aperture radar (SAR) sensors have been widely used because of their ability to work throughout the day in various weather situations, as well as their high resolution and penetrating ability [1], [2], [3], [4], [5]. With an increasing amount of data acquired by SAR imaging systems, SAR automatic target recognition (ATR) technology has become one of the research hotspots in the field of image cognition [6], [7]. A growing number of deep neural network (DNN) models have been applied to SAR ATR [8], [9]. Convolutional neural networks (CNNs), for example, have gradually become the standard model in the field of SAR image processing due to its powerful feature extraction capabilities [10], [11], [12]. Chen and Wang [13] converted SAR images into a set of feature maps to propose a new CNN model. Min et al. [14] proposed a micro CNN, which is a compressed form of deep convolutional neural networks (DCNNs) that utilizes a novel knowledge-distillation algorithm called gradual distillation. Huang et al. [15] applied an enhanced DCNN to learn the features of SAR images and the support vector machine to map features into output labels. Numerous studies have shown that CNN models can significantly increase SAR ATR accuracy.

However, state-of-the-art CNN models require a large number of labeled samples during the training process to attain high recognition accuracy. The speckle noise and clutters in SAR images make sample annotation challenging; therefore, gathering SAR labeled data is generally time consuming and expensive. Studies show that when the number of labeled samples is insufficient, the recognition accuracy of the CNN is significantly reduced [16], which severely limits the utilization of the CNN in SAR ATR problems. Numerous studies have employed few-shot learning techniques to raise the performance of DNN models in terms of recognition accuracy when there are not many labeled samples available [17], [18], [19]. Meanwhile, there is a lot of feature information in the unlabeled samples that can be used to improve the training effect of the model. In contrast, the acquisition of such unlabeled samples is simpler than labeled samples. Therefore, with a small number of labeled samples, a number of researchers have attempted to take advantage of some

additional unlabeled data to boost the DNN model's recognition ability. Yue et al. [20] introduced thresholding processing and the linear discriminant analysis method to the CNN to achieve a superior strategy of deep semisupervised learning (DSSL). Chen et al. [21] proposed a semisupervised method based on the consistency criterion, domain adaptation, and top-k loss to alleviate the need for labeled samples. Wang et al. [22] designed a new framework comprising a self-consistent augmentation rule, mixup-based mixture, and weighted loss, which allows a classification network to utilize unlabeled data during training.

In the context of SAR target recognition, DSSL has progressively become a popular research topic [23], [24]. Generative methods, consistency regularization methods, graph-based methods, pseudo-labeling methods, and hybrid methods are five types of DSSL [25]. The consistency regularization method, which is derived from network noise regularization [26], yields good results. Goodfellow et al. [27] demonstrated the advantages of adversarial noise over random noise. In addition to noise, the consistency constrained target's quality was also shown to be critical in the procedure. Bachman et al. [28] present a novel regularizer based on making the behavior of a pseudo-ensemble robust to the noise process that generates it. The ladder network [29], [30] is the first successful attempt toward using a teacher–student model that is inspired by a deep denoising autoencoder. An attention-augmented convolutional network (ACN), termed the Π model [31], and a temporal ensembling model [32] create two random augmentations for both labeled and unlabeled data to achieve regularization. The mean teacher [33] uses an exponential moving average (EMA) algorithm to create teacher and student models that are similar in structure but not in parameters. Virtual adversarial training (VAT) [34] utilizes the concept of adversarial attacks to improve targets with adversarial noise and make regularization more effective.

All of these teacher–student models, however, have some limitations. First, the teacher network is derived mostly from the EMA of the student network in all teacher–student strategies. The most typical model is a mean teacher, which introduces perturbations between networks by constructing another network from one network's EMA calculation, thus exploiting the consistency constraint's effectiveness. The teacher has the same parameters as the student in the Π model, the temporal ensembling, and VAT methods, which is equivalent to setting the average coefficient to zero. In these methods, two networks are strongly coupled, and the degree of coupling increases with training. As a result, the student uses the teacher's potentially inaccurate recognition results as a target for its own learning, making it difficult to achieve the optimal recognition performance [35]. To address this challenge, Ke et al. [35] presented a dual-student model, which replaces the teacher model with another student model while creating a stabilization constraint based on the concept of stable samples to make the model trainable. However, because of the coupling and variations in the teacher–student structure, the network's semisupervised training results show a significantly improved generalization performance, demonstrating the utility of its structural property. In this article, we present a new semisupervised consistency strategy in which a "classmate" is introduced to classic teacher–student structures to construct a trimodel consistency structure. The purpose of the classmate model is to reduce the teacher–student model's coupling by developing new guidance for the student. Furthermore, to ensure that the classmate model plays a positive role in the student model's training process, we utilize information entropy to construct a weak consistency constraint (WCC) to balance the two guides on the student.

Second, while CNNs are generally the best option due to their superior performance in deep semisupervised recognition tasks, they still lack the ability to capture global relational information against image transformation [36]. DSSL aims to make efficient use of unlabeled data to assist the network to learn from the labeled data. However, image targets contain a lot of global information in addition to local information, and the lack of such information can impede further improvement in semisupervised recognition performance when labeled data are insufficient. In this article, the self-attention mechanism is combined with a standard convolutional network to serve as a feature extractor, so that it can make better use of sample information. The self-attention mechanism is a type of attention mechanism (AM) that first appeared in natural language processing applications. It relies on little external data and thrives at capturing the internal association of data or features. Many subsequent computer vision studies have started to investigate integrating CNN structures with self-attention mechanisms, which has achieved impressive results [37], [38]. Some studies have even replaced the entire convolutional network structure with a self-attentional structure [39], [40]. Zhang et al. [41] proposed an AM-CNN model by combining the AM with deep convolutional networks and achieved better recognition performance. Ma et al. [42] proposed a attention graph convolution network, which combines an AM layer and graph convolution networks to achieve image segmentation in big SAR imagery data. Bello et al. [36] suggested a 2-D relative self-attentional approach to strengthen convolutional networks. Inspired by these approaches, this research proposes ACNs by combining the self-attention mechanism with convolutional networks in parallel. In order to adapt the AM to the image task, it is usually necessary to obtain positional information in the images. In [43], images are transformed into a sequence of image patches, while adding fixed position embedding to image patches to improve the effectiveness of self-attention for image targets. The encoding of positional information in this method is obtained by learning, which is a less complex way of position embedding. Bello et al. [36], on the other hand, extended the use of relative position encoding [44] to two dimensions and proposed a 2-D position encoding method. The CoordConv [45] method directly connects the position channel to the activation map. Although many methods succeed in achieving position embedding, they usually perform this in a more straightforward way and have limited effect on the improvement of self-attention mechanisms. While extracting position information and applying it to the learning process in a more efficient way necessitates a significant amount of additional computing, this is often at the expense of computational efficiency. We present a novel recurrent neural network (RNN)-based solution for extracting 2-D

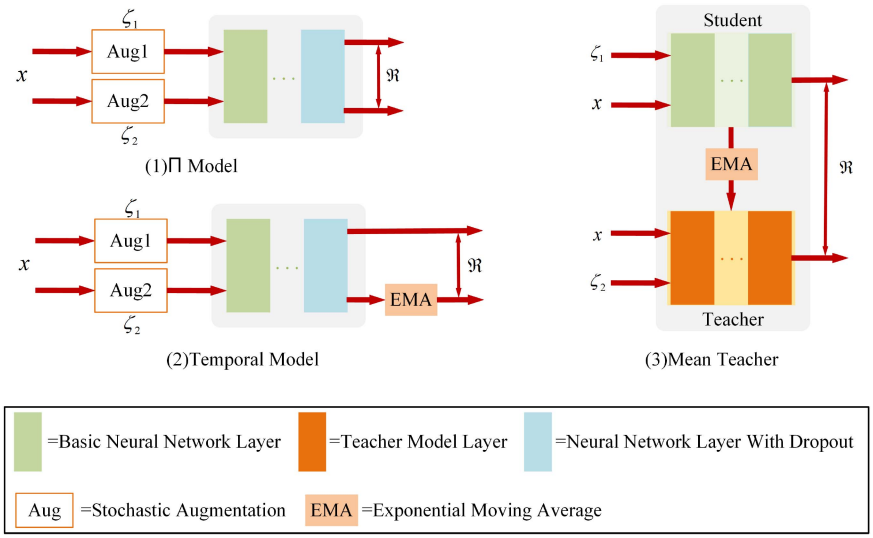

Fig. 1. Various architectures used for consistency regularization semisupervised methods. In addition to the identifiers in the figure, ζ denotes the perturbation noise. $\Re$ is the consistency constraint. x is the input data.

position information to simultaneously solve these two challenges. In particular, this enhances the effect of self-attention for image data by utilizing multiple RNNs to process short sequences in parallel with high efficiency.

In overview, in this study, a novel consistency-regularization-based DSSL method is proposed to alleviate the confirmation bias problem. Furthermore, an ACN is designed as a feature extraction network to optimize the utilization of the sample's information for semisupervised recognition. The originality and significance of our proposed method are outlined as follows.

1) ACNs are employed as a feature extraction network to capture both local and global information in images in the case of insufficient labeled samples. Furthermore, the RNN's superior sequence target processing capability is used to extract 2-D position information in the self-attention part.
2) A classmate model is introduced to present a trimodel consistency regularization procedure, alongside a WCC scheme to balance the influence of the classmate and the teacher in training.

The rest of this article is organized as follows. In Section II, consistency regularization and self-attention mechanisms are briefly introduced. Section III describes the principle of our method in detail. Comparative experiments are performed in Section IV. Finally, Section V concludes this article.

II. Preliminary

A. Consistency Regularization

Consistency regularization's principle is that an input should be forecast consistently even if it is subject to tiny disruptions. The teacher–student structure is the dominant model in consistency regularization methods. In these methods, perturbations are often added to the inputs or network parameters of two networks. The teacher sets targets for the student to improve, and then, the student learns by imposing consistency constraints on both of their outputs. Formally [37], we assume that dataset X consists of labeled and unlabeled samples. Let θ denote the weights of the basic student. The consistency constraint l is defined as

$$l = \sum_{x \in X} \Re(f(\theta, x), \mathrm{T}_x) \tag{1}$$

where $\Re(\cdot, \cdot)$ is the distance between two vectors. $f(\theta, x)$ is the prediction from model $f(\theta)$ for input x. T_x is the consistency target generated by the teacher model. The Π model [31] expresses this consistency constraint in the form of

$$l = \sum_{x \in X} \Re(f(\theta, x, \zeta_1), f(\theta, x, \zeta_2)) \tag{2}$$

where ζ_1 and ζ_2 are two different perturbation noises created for the training dataset samples. The structure of the Π model is shown in (1) in Fig. 1. The two networks in the Π model use data augmentations and dropout to introduce stochastic perturbations, which is also a teacher–student structure in formal terms. When the same sample with different random noise is propagated forward twice, the predictions obtained may be different. Then, the Π model minimizes the difference between the two predictions by learning. Temporal ensembling is similar to the Π model with the addition of the EMA algorithm. The structure of temporal ensembling is shown in (2) in Fig. 1. Temporal ensembling uses EMA to accumulate the predictions over epochs as T_x to reduce computational overhead. Formally, the consistency constraint of the temporal ensembling model is

$$l = \sum_{x \in X} \Re(f(\theta, x, \zeta_1), \mathrm{EMA}(f(\theta, x, \zeta_2))) \tag{3}$$

EMA is calculated as

$$v_t = \alpha v_{t-1} + (1 - \alpha) y_t \tag{4}$$

where v_t is the average of the output predictions of last $1/(1-\alpha)$ epochs. y_t is the output prediction for epoch t. α is an adjustable hyperparameter. Mean teacher is the most typical

teacher–student structure, and its structure is shown in (3) Fig. 1. The student model is involved in the learning process, while the teacher model is derived from the EMA calculations of the student. Different random perturbations are also applied to each of the two networks' inputs. Formally, the consistency constraint of the mean teacher model is

$$l = \sum_{x \in X} \Re(f(\theta, x, \zeta_1), f(\text{EMA}(\theta), x, \zeta_2)). \tag{5}$$

In addition, the VAT model [34] employs adversarial noise to generate higher quality T_x, which improves the training effect.

The teacher computed by the student will have a stronger generalization capacity than the student in these teacher–student structural models. This method of target network generation, however, results in a significant coupling between the two networks. This coupling effect has been visualized in [35]. Because of the coupling effect, if a student makes biased predictions for specific samples, the EMA teacher is more likely to accumulate the errors and force the student to follow, resulting in irreversible misclassification. This is a kind of the confirmation bias [33]. The quality of the targets provided by the teacher's model determines the success of consistency regularization in this model structure. Improving the quality of the targets can help mitigate the impacts of the confirmation bias, and mean teacher and VAT are two examples. Furthermore, the "valid" information that the teacher offers to the student varies as the network training progresses. The recognition accuracy of the teacher is still low at the start of the training, so the student should not regard it as a fully trusted target at this stage. As the recognition performance of the teacher and the student improves in the middle or later stages of training, the information that two networks may supply to each other at this point is accurate and effective. Thus, it is necessary to modify the degree of coupling between the teacher and the student to prevent confirmation biases.

B. Self-Attention Mechanism

In cognitive science, humans prefer to selectively focus on a portion of the information while disregarding the rest. This mechanism is known as the AM. Therefore, the AM in deep learning is concerned with determining which portions of the input to concentrate on and how to devoting limited information processing resources. Based on such properties, the AM can be employed as an effective way to extract global relational information from data.

The AM's purpose is to determine the degree of necessity for the source sequence to pay attention to each element in the target sequence. This degree can be represented by attention values. Attention values can be calculated by mapping a query and a set of key–value pairs to an output, where the query, keys, values, and output are all vectors. Query is an element in the target sequence, and each key–value pair corresponds to an element in the source sequence. The output is the weighted sum of the values, with the weight assigned to each value determined by the query's compatibility function with the corresponding key. In a self-attention mechanism, the source and target sequences are the same, which means self-attention deals with the information between each element in a sequence and the sequence itself.

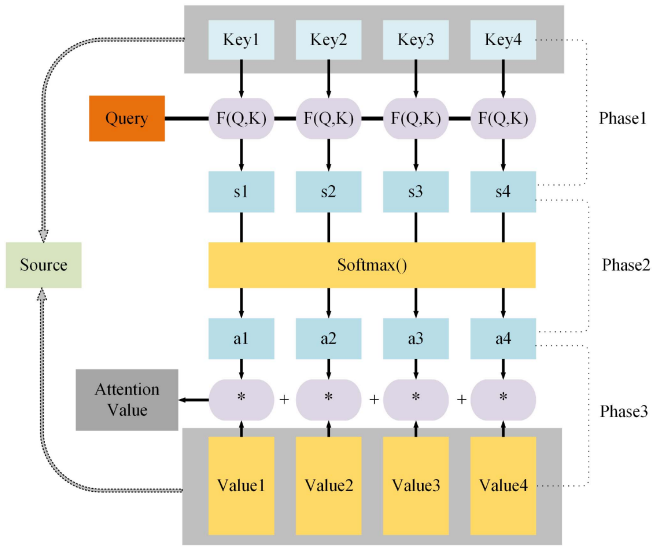

Fig. 2. Computational process of the attentional mechanism. To simplify the representation, let the source sequence contain four key–value pairs. Similarity calculation is performed in phase 1, where $F()$ is the dot product operation and s_i is the similarity score. Phase 2 performs Softmax normalization, and the resultant a_i is the weight of values. The weighted summation process of values is performed in phase 3. The final result is the attention value of the source sequence for query.

Fig. 2 shows the computational process of the attentional mechanism. The procedure can be split into three phases. The first phase is to calculate the similarity between the query and keys. We use the dot product to quantify this similarity

$$s_i = F(\text{query}, \text{key}_i) = \text{query} \cdot \text{key}_i \tag{6}$$

where s_i is the similarity score. The Softmax calculation is introduced in the second phase, which comes from the results of the previous phase. On the one hand, normalization can be performed, and on the other hand, the inherent mechanism of Softmax can emphasize the weights of significant elements. The output of Softmax is

$$a_i = \text{Softmax}(s_i) = \frac{e^{s_i}}{\sum_{j=1}^{N} e^{s_j}} \tag{7}$$

where a_i is the weighting factor of value$_i$. N is the size of the source sequence. In the third phase, a_i is used to weight the summation of values

$$\text{Attention} = \sum_{i=1}^{N} a_i \cdot \text{value}_i. \tag{8}$$

For image targets, their elements that match to the source and target sequences can be image patches [43], pixel dots, or features. In this case, the query, keys, and values vectors are generally converted to two dimensions and represented in matrix form. The attention value for the image target is calculated as

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V \tag{9}$$

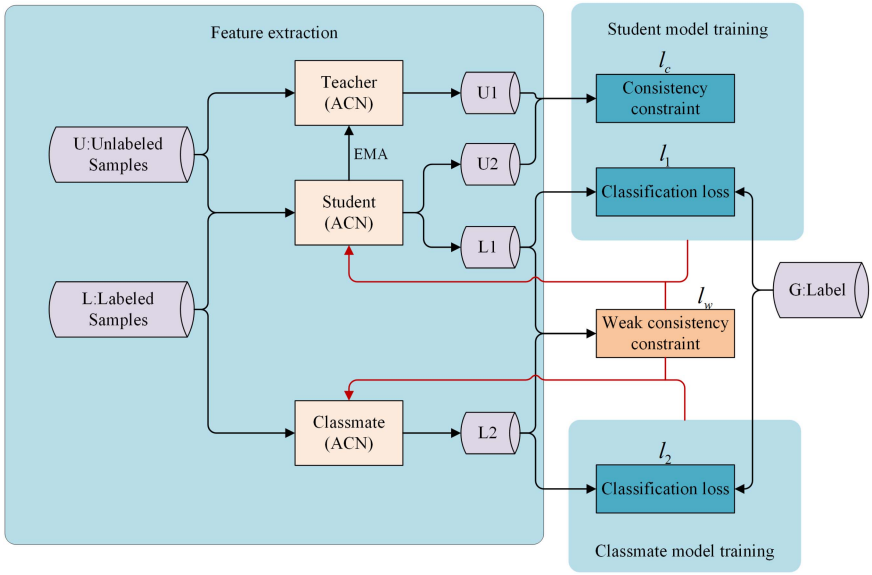

Fig. 3. Model overview. The proposed model mainly consists of a feature extraction part and two network training parts. Three ACNs are utilized to extract features from the labeled dataset L and the unlabeled dataset U to get the predicted outputs L1, L2, U1, and U2. The consistency constraint is then applied to U1 and U2 to obtain the loss l_c. The WCC is applied to L1 and L2 to obtain loss l_w. L1, L2, and true labels G are calculated to obtain the classification losses l_1 and l_2, respectively. l_1, l_c, and l_w are used for the student training, while l_2 and l_w are used for the classmate training.

where Q, K, and V are the matrix forms of queries, keys, and values. d_k is the depth of queries and keys. The acquired attention maps are also in the form of a matrix. It can be seen that the key principle of the self-attention mechanism is to create the weighted average of the information corresponding to the target elements. Unlike pooling or convolutional operations, the weights used in this process are dynamically computed by a similarity function between the elements. Therefore, the final computed attention value is decided by the interplay between the elements of the signal itself, rather than being predicted by their relative positions as in the convolutional procedure. This property enables the self-attention to gather long-range relational information without increasing the number of parameters. In an SAR semisupervised recognition task, collecting global information of samples can considerably compensate for the lack of information caused by a limited number of labeled samples, while the superior local information extraction capacity of convolutional networks should not be discarded. As a result, a feature extraction scheme that combines self-attentional and convolutional operations may be useful.

III. Methodology

We first define the system parameters. The training dataset $X = [L, U] \in R^{d\times n}$ consists of two parts. $L = [x_1, x_2, \ldots, x_l] \in R^{d\times l}$ represents the labeled dataset and $U = [x_{l+1}, x_{l+2}, \ldots, x_{l+u}] \in R^{d\times u}$ represents the unlabeled dataset, where d denotes the dimension of the samples. n, l, and u represent the number of samples in X, L, and U respectively. $G = [y_1, y_2, \ldots, y_l] \in R^{1\times l}$ denotes the labels. The labeled samples are input to the student and classmate models, while the unlabeled samples are input to the teacher and student models and provide L1, L2 and U1, U2, respectively.

As shown in Fig. 3, the training process of our method is composed of three parts: feature extraction, student model training, and classmate model training. In the feature extraction part, we use three ACNs to obtain features from labeled and unlabeled samples individually to get the output predictions. The student and the classmate are trained simultaneously. In the student model training part, we establish a normal consistency constraint l_c and a WCC l_w between the student and the other two networks, respectively. These two constraints are employed as semisupervised components of the loss function, which guides the student training. EMA calculation is used to obtain the teacher's parameter from the student in each epoch. Meanwhile, we train the independently initialized classmate with labeled samples in the classmate training part, and l_w also operates on the classmate model to increase its training effect. Next, the feature extraction process of the ACN and the trimodel consistency regularization training method are described in detail.

A. Attention-Augmented Convolutional Networks

Convolutional networks have been considerably successful in many computer vision applications, especially in image classification. Local and global features are the two types of features that can be obtained in images. Convolutional layers excel at extracting local features because they can enforce locality through a confined receptive field while reducing the number of parameters using weight sharing. However, this nature of CNN leads to the lack of capability to learn the image's global context, which is normally required for superior target recognition [46]. We present a self-attentional approach to gain additional global relational information from images and so increase the utilization of image information. The self-attention mechanism is a form of AM that focuses on the importance of each element

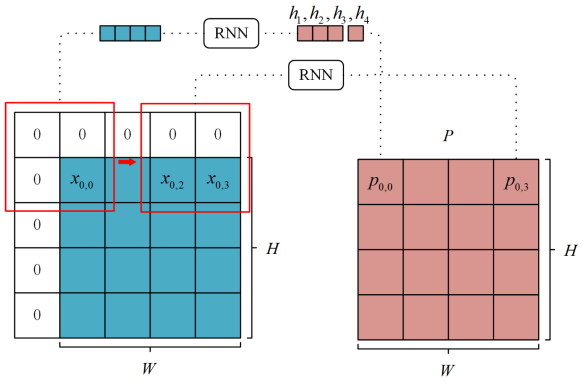

Fig. 4. Principle of 2-D LocalRNN. To simplify the representation, we set the size of the input image to be $H \times W = 4 \times 4$, and complement 0 to the left and upper sides. A local short sequence of length 4 for pixel $x_{i,j}$ is formed, which is input into the RNN to retrieve the corresponding four hidden states h_1, h_2, h_3, and h_4, where h_4 represents the position information $p_{i,j}$ of $x_{i,j}$. P is the position information map.

in a sequence to the sequence itself. As a result, it excels at capturing the internal correlation of data or features, which is a complement to the CNN's capabilities. In this article, we integrate the self-attentive mechanism with a convolutional network to create attention-augmented convolutional module (ACM) and then replace the convolutional layer in the CNN with the ACM to construct ACNs to gain improved image feature extraction capabilities.

1) Two-Dimensional Positional Embeddings Based on the RNN: Self-attention in images is permutation equivariant without explicit information about positions, which means for any permutation transformation π of image X

$$\text{Attention}(\pi(X)) = \pi(\text{Attention}(X)). \tag{10}$$

Therefore, self-attention treats all the information at each position equally and cannot distinguish among them, making it less effective for targets with complex structures such as images. To solve this issue, numerous positional embedding algorithms that extract spatial information and integrate it into the learning process have been proposed to improve self-attention. The image transformer [47] extends the sinusoidal waves from the original transformer [48] to 2-D inputs to create a 2-D positional embedding. Bello et al. [36] proposed a 2-D position encoding system that extended the use of relative position representations [44] to two dimensions. However, current position embedding methods in image attention face the problems of both excessive computation and insufficient improvement for self-attention. As a solution, we propose to extend the use of LocalRNN in sequences [49] to two dimensions and design a 2-D positional embedding method based on the RNN. It utilizes the unique parallel computation of LocalRNN to extract the spatial information with high efficiency.

Fig. 4 depicts the principle of the 2-D LocalRNN. We set up a square window that moves on the input image and use the RNN to extract the local structure of the lower right pixel in the window, and this local structure can be used as the position information of this pixel. Specifically, we set the size of the window to 2.

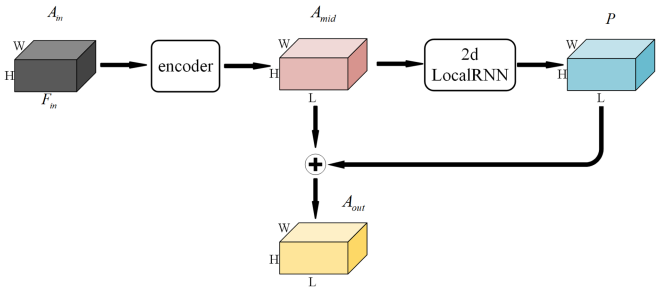

Fig. 5. Two-dimensional positional embedding module. The input map A_{in} is encoded to obtain the intermediate map A_{mid} containing pixel information. A_{mid} is then processed by the 2-D LocalRNN module to obtain the position information map P. Finally, P and A_{mid} are summed by the corresponding elements to deliver the positional embedding and produce the final output A_{out}.

For the pixel vector $x_{i,j} \in \mathbb{R}^L$, the RNN sequentially processes a short local sequence $x_{i-1,j-1}, x_{i-1,j}, x_{i,j-1}, x_{i,j}$ and outputs four hidden states, with the last hidden state used as the position information $p_{i,j}$ of the pixel

$$h_1, h_2, h_3, h_4 = \text{RNN}(x_{i-1,j-1}, x_{i-1,j}, x_{i,j-1}, x_{i,j}) \tag{11}$$

$$p_{i,j} = h_4 \tag{12}$$

where RNN() denotes the RNN cell, while long short-term memory (LSTM) is used in this article. h_1, h_2, h_3, h_4 is the output sequence of the RNN cells. $p_{i,j} \in \mathbb{R}^L$ is the position information of $x_{i,j}$, where L is the depth of pixel vectors. In this way, the spatial structural information of a target position and its surrounding elements is retrieved by using the RNN's sequence processing capabilities. In addition, only the neighboring positions preceding the processing position are included within the LocalRNN window in order to process a position without integrating future information and to enable the model to process the sequence in an autoregressive way. We employ the complementary 0 operation to ensure that there are enough pixels in the window for the RNN operation while processing the edge positions. Finally, the position information map P containing the spatial information of each pixel can be obtained.

The overall procedure of 2-D position embedding is shown in Fig. 5. The input map $A_{\text{in}} \in \mathbb{R}^{H \times W \times F_{\text{in}}}$ may be the initial SAR image or feature map, where H and W are the height and width of the input, respectively. F_{in} is the number of input filters of the input map. A_{in} is first fed through an encoder that gives each pixel of the input map the information of dimension L through a learnable linear transformation. The encoder outputs the intermediate matrix $A_{\text{mid}} \in \mathbb{R}^{H \times W \times L}$. To obtain the position information map P, the intermediate map is subjected to 2-D LocalRNN operations, which are executed in parallel. Finally, the embedding of 2-D information is achieved by summing the matching elements of the intermediate map and the position information map. The whole process can be expressed as

$$A_{\text{mid}} = A_{\text{in}} * w \tag{13}$$

$$A_{\text{out}} = A_{\text{mid}} + \text{LocalRNN}(A_{\text{mid}}) = A_{\text{mid}} + P \tag{14}$$

where $w \in \mathbb{R}^{F_{\text{in}} \times L}$ is learned linear transformation. A_{out} obtained from the above positional embedding module can be used

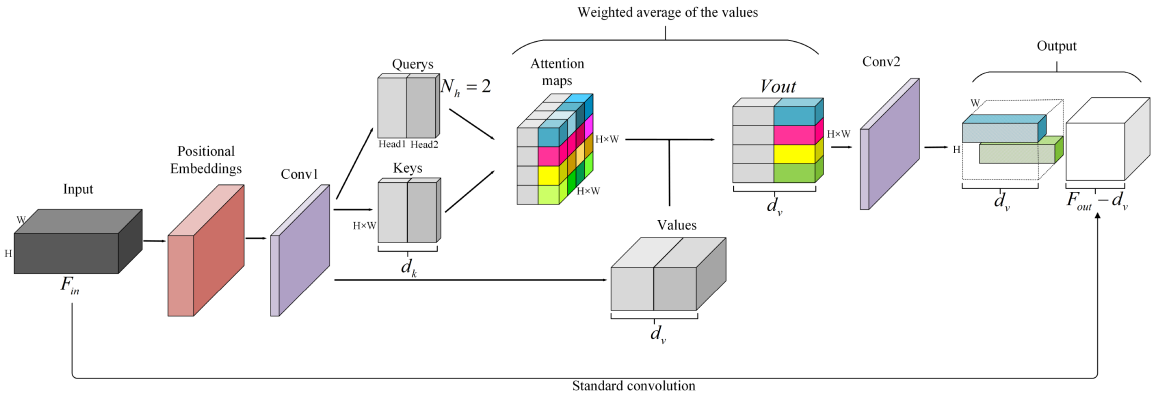

Fig. 6. ACM structure. The input is embedded with position information and then enters convolutional layer conv1 to calculate matrices queries Q, keys K, and values V, where d_v is the depth of values. N_h attention maps are calculated by multiplying Q and K to weight average V to obtain $Vout$. The results are then mixed with a pointwise convolution conv2. Multihead self-attention is implemented in parallel to a standard convolution operation and the outputs are concatenated to the final F_{out}-dimensional output.

as a new input map in the self-attention calculation part, which helps self-attention overcome the problem of the permutation equivariant with considerable calculation efficiency.

The properties of the RNN are used in our method to extract position information from image targets. The RNN has a limited ability to capture the long-term dependencies and the time complexity of its computation is linearly related to the length of the sequence. However, the 2-D LocalRNN module in our method does not suffer from the above problem because its processing targets are short sequences in fixed-size windows. Meanwhile, the RNN processes every short sequences independently, so it is very simple to implement parallel operation and can achieve high processing efficiency. Self-attention for image data can be considerably improved by using this positional embedding module.

2) Attention-Augmented Convolutional Module: In image processing tasks, the self-attention mechanism is usually performed in two steps: 1) acquisition and embedding of image position information and 2) self-attentive computation in matrix form. The specific implementation is usually to use the input map containing the positional information as the matrix to be processed. Then, the source and target matrices are obtained by linear transformation. Finally, the self-attentional output map is obtained by matrix similarity calculation and weighting operation. Our proposed method is also based on this idea for the construction of the network structure. Fig. 6 illustrates the structure of the ACM. The self-attention part and the standard convolution part are computed in parallel, and the outputs of them are concatenated as the layer's output. In the self-attention part, the multihead self-attentional approach is used, where N_h is the number of heads. The size of the input is $H \times W \times F_{\text{in}}$, where H and W denote the height and width of the input image, respectively, and F_{in} is the number of input filters of the input image. The input map is first extracted by the 2-D positional embedding module for spatial position information. Then, the convolutional layer conv1 is utilized to compute the three matrices Q, K, and V, which denote queries, keys, and values in image attention. The specific implementation is to use a convolutional layer as a linear transform to transform the input map containing the position embedding into a map of large depth and then split it into three matrices corresponding to the source and the query. Formally, this process can be viewed as three learnable linear transformation matrices to linearly transform the input map separately. We flatten the input map $I_x \in \mathbb{R}^{H\times W\times F_{\text{in}}}$, which has been positionally embedded, into a 2-D matrix $I_x \in \mathbb{R}^{HW\times F_{\text{in}}}$, and the attention values are computed as

$$\text{Attention} = \text{softmax}\left(\frac{(I_xW_q)(I_xW_k)^T}{\sqrt{d_k}}\right)(I_xW_v) \tag{15}$$

where $W_q, W_k \in \mathbb{R}^{F_{\text{in}}\times d_k}$ and $W_v \in \mathbb{R}^{F_{in}\times d_v}$ are learned linear transformations that map the input to queries $Q = I_xW_q \in \mathbb{R}^{HW\times d_k}$, keys $K = I_xW_k \in \mathbb{R}^{HW\times d_k}$, and values $V = I_xW_v \in \mathbb{R}^{HW\times d_v}$. d_k denote the depth of queries and keys, while d_v denote the depth of values. Attention maps are obtained by QK^T. Then, we complete the weighted average calculation of V with attention maps as weights

$$Vout = \sum_{i=1}^{d_k}\sum_{j=1}^{d_k}\text{softmax}(q_i * k_j) * v_j \tag{16}$$

where $Vout \in \mathbb{R}^{HW\times d_v}$ is the result of the weighted average of V. q_i and v_j are rows i and j of Q and V, respectively. k_j is row j of K. $Vout$ goes through a pointwise convolution layer conv2 to mix the results of the multihead calculation and then is reshaped to match the original volume's spatial dimensions and size. The output of this section contains the information of the global relationships between the pixels in the images.

Existing studies show that self-attention possesses enough competitiveness to completely replace convolutional networks [39]. But instead of absolutely abandoning the concept of convolution, we combine it with the self-attention mechanism and augment the former with the latter. To limit the size of the network, we use only one layer of standard convolution in the convolution part to obtain the local feature mapping. If the outputs of the two parts are combined in a summation manner, the features obtained from each of the two parts may constrain each other and cannot be best utilized at the same time.

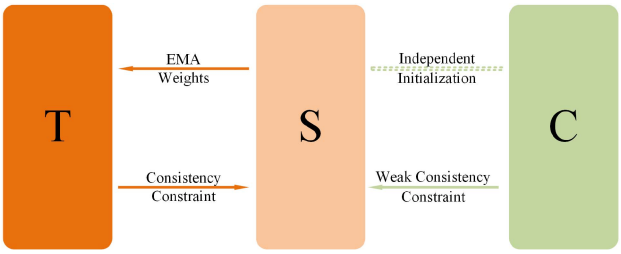

Fig. 7. Relationship between the three models. The classmate and the teacher guide the student's training together. The teacher is made from the EMA of the student and imposes the consistency constraint on the student. The classmate is initialized independently from the student and imposes the WCC on the student.

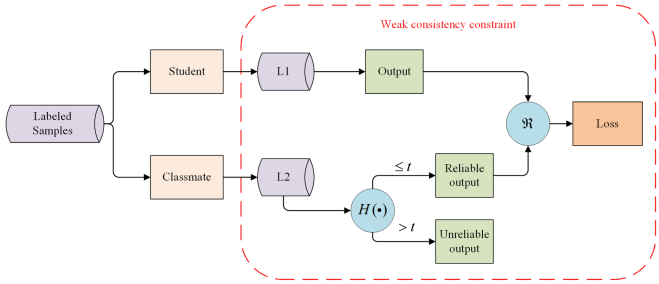

Fig. 8. WCC. The student and the classmate process the labeled samples simultaneously to obtain the predicted outputs L1 and L2. $H()$ denotes the information entropy function. The outputs with information entropy less than the credibility threshold t in L2 are reliable outputs, which are computed with the outputs of the same samples in L1 to obtain the consistency loss.

In addition, since the depth of the self-attention map must be d_v, using this merging method will limit the size of the ACM's output. Based on the above considerations, we concatenate the convolutional feature mapping that enhances locality with a self-attentional feature mapping that can model the long-range relationships as the output of the ACM, as shown in Fig. 6. The output depth of standard convolution part is set to $F_{\text{out}} - d_v$, where F_{out} denotes the freely settable depth of ACM's outputs.

B. Trimodel Consistency Regularization

1) Weak Consistency Constraint: As shown in Fig. 7, the purpose of introducing the classmate model is to impose a new consistency constraint on the student as a part of the student's loss function, thus lowering the coupling effect between the teacher and student models and alleviating the confirmation bias problem. However, owing to the performance disadvantage associated with the supervised training strategy of the classmate model, it cannot enforce too strong a constraint on the student; otherwise, it may not only fail to suppress bias, but also affect the effectiveness of the teacher as a target for the learning of the student. Therefore, in the trimodel training procedure, the teacher should continue to operate as the primary guide for the student's learning, and the classmate should only serve as a supplement. Based on the preceding discussion, we design a WCC to allow the classmate to play a weak guidance role to the student, hence successfully reducing the effects of the confirmation bias.

We introduce information entropy to design the WCC module, and the principle is shown in Fig. 8. The output vector of the ACN for an input sample is the predicted probability value that the sample belongs to each category. The uncertainty of the predicted value can be measured by the information entropy. A larger value of information entropy indicates that the uncertainty of the predicted value is greater, which means the prediction vector is close to the edge of the classification surface and its probability of belonging to each category is uniform. Therefore, we determine the reliability degree of this output by calculating the information entropy of this prediction vector. We define the reliable outputs to be outputs with information entropy values less than or equal to the credibility threshold t, and unreliable outputs to be the opposite. When sample x_i is input to model $f(\theta)$, the output is assumed to be $f(\theta, x_i) = [q_1^i, q_2^i, \ldots, q_N^i]$, where q_k^i denotes the probability that sample x_i belongs to category k and N denotes the number of categories. The information entropy of this output is calculated as

$$H(f(\theta), x_i) = -\sum_{k=1}^{N} q_k^i \log_2(q_k^i + \sigma) \tag{17}$$

where $H(f(\theta), x)$ is the information entropy of the output of model $f(\theta)$ for sample x. σ is a small constant, which is used to cope with the situation of $q_k^i = 0$. We set $\sigma = 1 \times 10^{-12}$. The determination of reliable output is then performed. If the output is unreliable, it will not be involved in the calculation of the loss function in this training epoch. Otherwise, a component of constraint loss is calculated between this reliable output and the output of the student model for the same sample. A larger t indicates a higher performance condition required for the classmate to guide the student, and if the performance of the classmate model is not sufficient to obtain prediction output that satisfy the threshold for most samples, the classmate model will barely participate in the training of the teacher–student model. As a result, it is particularly important to choose an appropriate t. Formally, the WCC is

$$\begin{aligned} &\text{WCC}(f(\theta_1), f(\theta_2), X) \\ &\quad = \sum_{x \in X} F(f(\theta_1, x), f(\theta_2, x)) \end{aligned} \tag{18}$$

where WCC() is the WCC function. θ_1 and θ_2 denote the weights of the student and classmate models, respectively. $F()$ is expressed as

$$F(f(\theta_1, x), f(\theta_2, x)) = \begin{cases} 0, & H(f(\theta_2)x) \geq t \\ \Re(f(\theta_1, x), f(\theta_2, x)), & \text{else} \end{cases} \tag{19}$$

where $\Re(\cdot, \cdot)$ is the distance between two vectors. As the training proceeds, the recognition performance of the classmate model continues to improve and the amount of its reliable outputs will gradually increase. Meanwhile, as the performance of the student and the teacher gradually increases, the effective guidance provided by the classmate to the student will gradually deteriorate. In summary, by imposing the WCC to the student from the classmate in the early stage of training, the performance

impact of the strong coupling in the teacher–student structure can be effectively weakened.

2) Loss Function: The training process of the trimodel method is shown in Fig. 3. A conventional consistency constraint is imposed between the student model and the teacher model, and the WCC is imposed between the student model and the classmate model, and the mean square error (MSE) is used for both the constraint functions. The student and the classmate are initialized independently. The student is processed by EMA in each epoch to obtain the teacher, which can achieve better generalization performance than the student and thus serve as a learning target for the student model.

The student model is trained using a semisupervised strategy, in which both the labeled and unlabeled samples are used in each epoch, and its loss function has three components: 1) the classification loss between the predicted outputs of the labeled samples and the true labels; 2) the consistency constraint loss between the predicted output of the student model and the teacher model for unlabeled samples, respectively; and 3) the WCC loss between the predicted output of the student model and the classmate model for labeled samples, respectively. Formally, the loss function is as follows

$$\text{Loss}_{\text{stu}} = l_1 + a * l_c + b * l_w \tag{20}$$

where l_1 is the classification loss, using the cross-entropy loss function. a and b are hyperparameters. l_c is the consistency constraint loss between the student and the teacher

$$l_c = \sum_{x \in U} \text{MSE}(f(\theta_1, x), f(\text{EMA}(\theta_1), x)) \tag{21}$$

where θ_1 denotes the weights of the student model. MSE() is the MSE function. $l_w = \text{WCC}(f(\theta_1), f(\theta_2), L)$ is the WCC loss between the student and the teacher. The function $F()$ in the WCC WCC() at this point is performed as

$$F(f(\theta_1, x), f(\theta_2, x)) = \begin{cases} 0, & H(f(\theta_2)x) \geq t \\ \text{MSE}(f(\theta_1, x), f(\theta_2, x)), & \text{else} \end{cases} \tag{22}$$

where θ_2 denotes the weights of the classmate model. Since the classmate model is only supervised training, its performance will gradually fail to keep up with the performance of the teacher–student model as the training progresses. And the attenuation of the confirmation bias was mainly performed in the early stage of training, because the predictions of the student and the teacher had substantially stabilized in the late stage. We, therefore, introduce a weight factor to make b decrease to zero as training proceeds. The expression of b is

$$b = \frac{E - e}{E} * c \tag{23}$$

where c is a hyperparameter, E is the total number of training epochs, and e is the current number of training epochs.

The classmate model is trained simultaneously with the student model, using a supervised training approach with labeled samples L. Unlike conventional supervised learning, the loss function of the classmate model training consists of two components: 1) the classification loss between the predicted outputs of the labeled samples and the true labels and 2) the WCC loss between the student and the classmate. The addition of the latter can enhance the supervised training effect of the classmate model and further improve its performance improvement effect on the student model. Formally, the loss function is as follows:

$$\text{Loss}_{\text{cla}} = l_2 + d * l_w \tag{24}$$

where l_2 is the classification loss, using the cross-entropy loss function. d is a hyperparameter. Our proposed approach is essentially an inductive training method that aims to train a recognition model in a semisupervised manner using labeled and unlabeled samples and, then, use this model to achieve the task of recognizing new data. In this respect, our approach is consistent with the traditional SAR ATR. We use the generated teacher model with better generalization performance to predict the labels of the test dataset samples to evaluate the performance of the model after the training is completed.

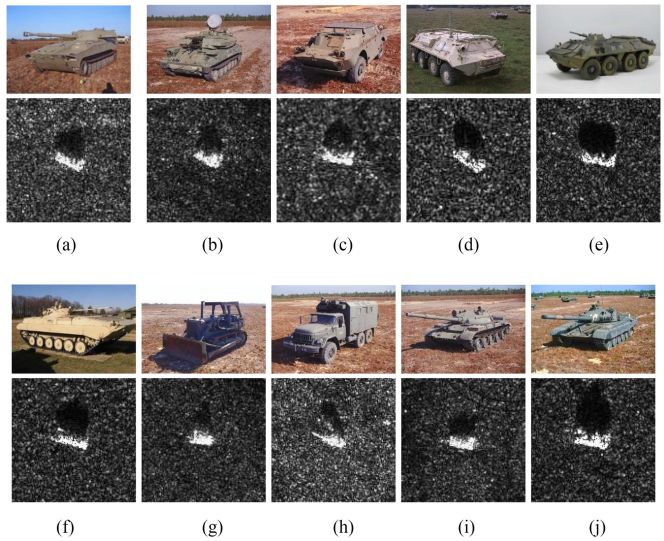

Fig. 9. SAR and optical images of ten targets in the MSTAR dataset. (a) 2S1. (b) ZSU234. (c) BRDM-2. (d) BTR60. (e) BTR70. (f) BMP2. (g) D7. (h) ZIL131. (i) T62. (j) T72.

IV. Experiments

A. Dataset

In this article, we use the MSTAR dataset, which is a benchmark dataset in the field of SAR image recognition, to validate the proposed method. This dataset uses a radar resolution of 0.3 m × 0.3 m and operates in the X-band, using HH polarization. MSTAR includes ten types of ground targets: T62, T72, BMP2, BRDM2, BTR60, BTR70, D7, ZIL131, 2S1, and ZSU234, and their optical images and SAR images are compared, as shown in Fig. 9. It can be seen that although the optical images of various types of the targets are clearly different from each other, their SAR images are difficult to identify due to their imaging nature.

In the original dataset, the size of the images is 128×128, and the target subject is concentrated in the center of an image.

TABLE I
TRAINING AND TESTING DATASETS IN EXPERIMENTS

Type	Tops	Model	Training set		Testing set	
			Depression	Number	Depression	Number
2S1	Artillery	B_01	17°	299	15°	274
ZSU234		D_08	17°	299	15°	274
BRDM2	Truck	E_71	17°	298	15°	274
BTR60		K10YT_7532	17°	256	15°	195
BMP2		SN_9563	17°	233	15°	195
BTR70		C_71	17°	233	15°	196
D7		92V_13015	17°	299	15°	274
ZIL131		E_12	17°	299	15°	274
T62	Tank	A_51	17°	299	15°	273
T72		#A64	17°	232	15°	196
			Sum:2747		Sum:2425	

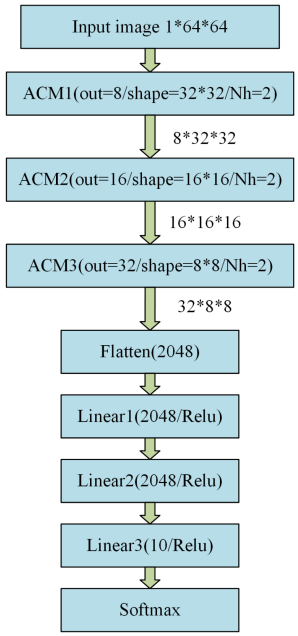

Fig. 10. Structure of the ACN. The network contains three ACM modules, one flatten layer, three fully connected layers, and one Softmax layer.

To reduce the interference of background clutters, we uniformly crop it to 64×64 at the center. Meanwhile, two different pitch angles of 15° and 17° are included for each type of the target. The dataset used in this article includes both the training and testing datasets, and Table I lists the details.

B. Experimental Settings

As shown in Fig. 10, the feature extraction network we employ contains three ACMs and three fully connected layers. Each ACM contains three convolutional layers and one LocalRNN layer. The first two convolutional layers conv1 and conv2 assist in the self-attention operation. The third convolutional layer is to extract local features as part of the final output, with a convolutional kernel size of 3×3 and a number of 32 convolutional kernels. The sequence processing part of the LocalRNN layer uses an LSTM with four input and output channels, and each LocalRNN layer is added with residual [50] and layernorm [51] connections, with dropout set to 0.5. The specific structure of the ACM is described in Section III. The fully connected layers Linear1, Linear2, and Linear3 have 2048, 2048, and 10 output channels, respectively, with the rectified linear unit as an activation function.

The CNN we utilized in our experiments has the identical structure as the ACN, except that the ACMs are replaced with convolutional modules. Each convolutional module consists of a convolutional layer, a batch normalization layer, and a max-pooling layer. Each convolutional layer has 64 convolutional cores with a convolutional kernel size of 3×3. The max-pooling layers use 2×2 convolution.

During the training process, we only crop the input images to 64×64 uniformly, without extra preprocessing and data expansion. ACM1, ACM2, and ACM3 have 8, 16, and 32 output channels, with the size of the output feature maps being 32×32, 16×16, and 8×8, respectively. N_h in the multihead self-attention is set to 2. The threshold t for determining the reliability degree in the WCC is set to 1.5. Fig. 11 shows the difference between our approach and the traditional SAR ATR way of using the dataset. In SAR ATR experiments, the dataset is usually divided into two parts: the training dataset is all available labeled data used for model training, while the data in the testing dataset is set to be unlabeled data used to test the recognition ability of the model. In contrast, in our SAR semisupervised recognition method, the training dataset consists of a small portion of labeled data and a larger portion of unlabeled data, both of which are available for training the model. In addition to this, we set up a test dataset that also consists of unlabeled samples that have never been observed in the model training, which is equivalent the newly added data to be recognized in real applications. In this regard, inductive semisupervised learning is the same as SAR ATR, which is a recognition method with new data generalization capability that can perform recognition tasks on newly obtained data.

Based on the analysis above, we treat the MSTAR dataset in the experiments as follows: The training dataset is divided into a labeled dataset L and an unlabeled dataset U. The same number of samples from each class in the training dataset is randomly selected and added to L. Since the training dataset

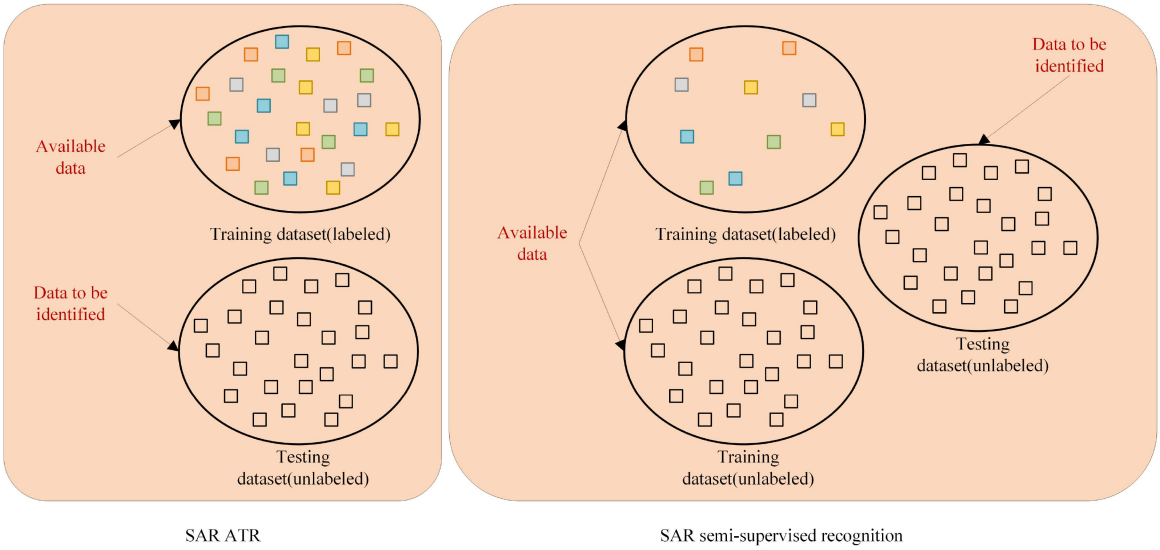

Fig. 11. Different ways of applying the dataset in SAR ATR and in our method.

TABLE II
NUMBERS OF SAMPLES IN L AND U UNDER DIFFERENT PARTITIONS OF THE TRAINING DATASET

	Number of L	Number of U	Number of T
1	300	2447	2425
2	400	2347	2425
3	500	2247	2425
4	600	2147	2425
5	800	1947	2425
6	1000	1747	2425

was partitioned for the purpose of evaluating the results of comparison experiments or ablation experiments under each partition, and not for the purpose of making cross-sectional comparisons between the results under each partition. Therefore, in order to make full use of the data volume of the dataset to enrich the unlabeled sample part, we add the remaining samples to U after constructing the labeled dataset L. In addition, there is a fixed number of unlabeled samples forming the test dataset T. After training the model with L and U, we use the trained model to identify the samples in T to obtain the experimental results. We set up six experimental groups based on the different ways of dividing L and U, and the sample sizes of each dataset in each group is shown in Table II. We utilize the Adam optimizer for optimization when training the feature extraction network, with the following parameters: $\eta = 0.001$, $\beta_1 = 0.9$, and $\beta_2 = 0.99$. Since differences in the randomly selected labeled samples may affect the experimental results, we repeat each experiment ten times to take the average.

C. Evaluation Indicators

We mainly use recognition accuracy and the Kappa score as the main evaluation metrics in the experiments. The recognition accuracy refers to the ratio of the number of correctly recognized samples to the number of all the samples. The Kappa score is a method for evaluating consistency, and we can use it to assess the accuracy of a multiclassification model. The closer the value is to 1, the higher the model classification accuracy. The calculation of Kappa score is based on the confusion matrix, which can accurately measure the recognition accuracy of each class. The definition of Kappa score is shown in (25), where p_o represents the relative observed agreement between the recognition results of the testing data and the true labels and p_e represents the hypothetical probability of the chance agreement

$$\text{Kappa} = \frac{p_o - p_e}{1 - p_e}. \tag{25}$$

D. Ablation Experiments

1) Evaluation of the ACN Feature Extraction Module: We compare the feature extraction capability of the ACN proposed in this article with that of the conventional CNN and the Transformer model [43], all of which use the trimodel consistency regularization method for training. The Transformer model uses the ViT-Base architecture from [43]. The results are shown in Table III.

It can be seen that our method outperforms the CNN under each dataset division. The reason is that the self-attention part and the convolutional part of our feature extraction module can extract both global and local information in the sample images simultaneously. Compared with CNNs that extract only local information, our strategy can significantly improve feature extraction and information utilization of images when the labeled samples is insufficient. In addition, the difference in performance between the two methods is more significant when the number of labeled samples is smaller. The improvement in recognition accuracy of the ACN method reaches 7.67% when L = 300, while the improvement is only 2.19% when L = 1000. This is because when the number of labeled samples increases, the CNN network's generalization capacity grows, and the gap between the two feature extraction networks is smaller, the performance

TABLE III
ACN, CNN, AND VIT FEATURE EXTRACTION CAPABILITY COMPARISON

Method	CNN		ViT-Base		ACN(ours)	
	Accuracy	Kappa	Accuracy	Kappa	Accuracy	Kappa
L=300	78.02	75.23	75.43	73.32	**85.69**	**80.98**
L=400	84.31	79.25	78.83	75.02	**88.82**	**87.41**
L=500	86.66	84.97	83.21	78.73	**91.59**	**89.42**
L=600	88.92	87.53	85.32	83.62	**93.73**	**92.45**
L=800	91.76	89.66	88.62	87.25	**94.72**	**93.53**
L=1000	93.73	92.51	90.11	88.14	**95.92**	**94.58**

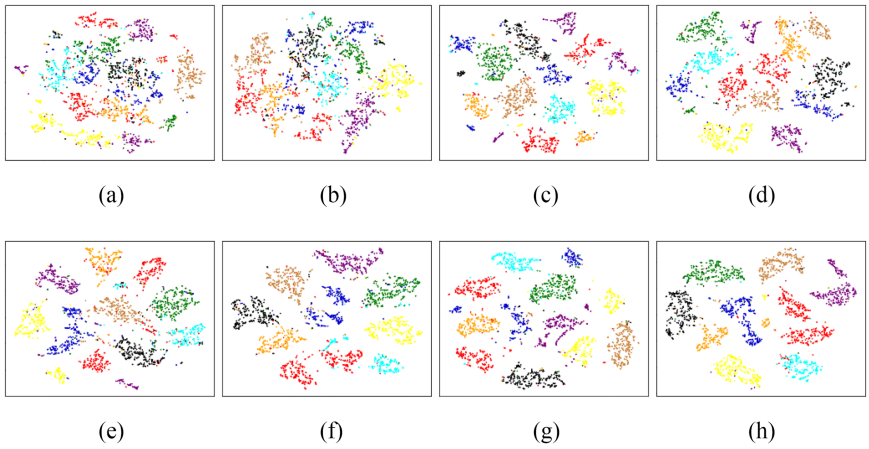

Fig. 12. Distribution of the feature vectors outputted by our model and the CNN model. (a)–(d) represent the supervised CNN model's output and (e)–(h) represent our model's output. Different colors represent different classes. (a) L = 400. (b) L = 600. (c) L = 800. (d) L = 1000. (e) L = 400. (f) L = 600. (g) L = 800. (h) L = 1000.

advantage of our method decreases. Even so, our method has a distinct advantage in each experimental partition.

The Transformer network, in comparison, has unsatisfactory recognition performance across all partitions, and each indicator is even lower than the CNN network. This is because the ViT network has a very large architecture with a huge number of parameters, while the dataset has a small amount of data, and training with the Transformer network would lead to severe overfitting. Besides, the Transformer model completely adopts a self-attention mechanism for network construction, giving up the ability to extract local features of images that convolutional networks can provide. The ACN for the combination of convolutional networks and self-attention, in contrast, is more adapted to the feature extraction task of SAR images.

Next, we use visual figures to illustrate the effectiveness of the ACN. We select partitions 2, 4, 5, 6 in Table II and extract the feature vectors of the test samples output by the trained ACN model and the CNN model. We then transform the features into 2-D feature vectors using the t-distributed stochastic neighborhood embedding (t-SNE) method. The training method for both the models uses the trimodel training. The distribution of the feature vectors obtained from the two models is shown in Fig. 12, where the points of different colors represent samples of different classes. It can be seen that as the number of labeled samples increases, the recognition performance of both the CNN and the ACN is gradually enhanced. Also, the degree of confusion between different classes of sample clusters in the visualization graph gradually reduces. In addition, it can be seen that compared with the CNN method, our method can effectively reduce the within-class distance and increase the between-class distance for the same number of labeled samples. This means that our model can better extract image features and, thus, train better recognition performance, which is consistent with the experimental results shown in Table III.

TABLE IV
COMPARISON OF THE RECOGNITION ACCURACY OF NETWORKS USING CNN, ACN WITHOUT POSITIONAL EMBEDDING, AND ACN WITH POSITIONAL EMBEDDING BASED ON LOCALRNN

Model	L:300	L:400	L:500	L:600	L:800	L:1000
CNN	78.02	84.31	86.66	88.92	91.76	93.73
ACN(without LocalRNN)	83.02	86.28	88.85	90.93	92.45	94.32
ACN(with LocalRNN)	**85.69**	**88.82**	**91.59**	**93.73**	**94.72**	**95.92**

2) Evaluation of the 2-D Positional Embeddings Based on the RNN: In the ACN feature extraction module proposed in this article, we employ an RNN-based 2-D location embedding module to enhance self-attention. Table IV shows the comparison of recognition accuracy under three network models using CNN, ACN without positional embedding, and ACN with positional embedding based on LocalRNN. It can be seen that

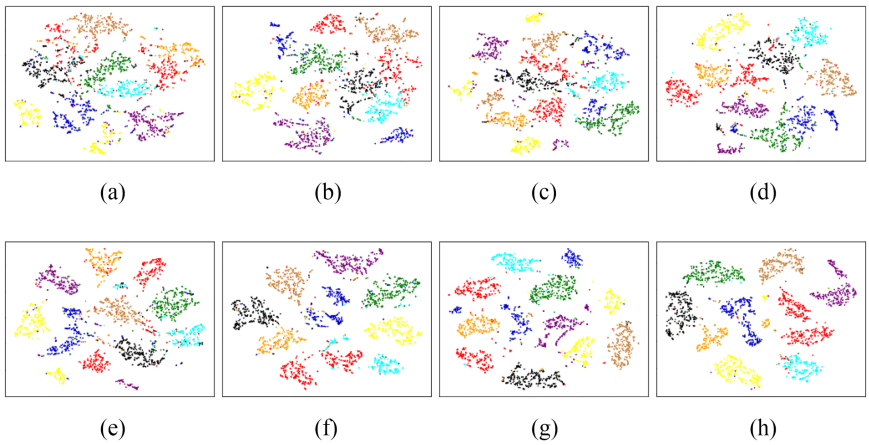

Fig. 13. Distribution of the feature vectors outputted by the ACN under trimodel training and supervised training. (a)–(d) represent the output of the ACN trained by the trimodel method, and (e)–(h) represent the output of the ACN trained by the supervised method. Different colors represent different classes. (a) L = 400. (b) L = 600. (c) L = 800. (d) L = 1000. (e) L = 400. (f) L = 600. (g) L = 800. (h) L = 1000.

TABLE V
PERFORMANCE COMPARISON OF OUR TRAINING METHOD WITH THE TEACHER–STUDENT METHOD AND THE SUPERVISED METHOD

Method	L:300	L:400	L:500	L:600	L:800	L:1000
Supervised	77.37	80.62	82.43	86.23	90.49	92.08
Teacher-student	84.27	88.77	90.57	91.67	93.58	94.27
Tri-model	**85.69**	**88.82**	**91.59**	**93.73**	**94.72**	**95.92**

the recognition accuracy of the ACN is higher than that of the CNN in every partition even in the absence of position information. This is because the self-attention augmented convolutional network extracts image features better than the traditional convolutional network and so achieves superior recognition performance. However, when the number of labeled samples in the training dataset increases, the performance advantage of the ACN without positional embedding over the CNN becomes small. At the same time, the recognition accuracy of the ACN with the 2D-LocalRNN module is the highest in each partition. The reason is that the 2D-LocalRNN module improves the effectiveness of self-attention in processing image data, resulting in more accurate global relationship information extracted by the self-attention unit.

3) Evaluation of Trimodel Consistency Regularization Method: We introduce a classmate model in this study to establish a WCC on the student model to lessen the coupling of the teacher–student, which is the difference between the trimodel consistency regularization method and the traditional teacher–student method. To validate the effectiveness of our approach, we compare our training method with the conventional teacher–student method and the supervised method, evaluated with recognition accuracy. The feature extraction network used is ACN, and the results are shown in Table V. It can be seen that the traditional teacher–student structure has high performance improvement over the supervised training method, most significantly at L = 300, reaching an accuracy improvement of 8.15%, while the smallest advantage is at L = 1000, with a 2.19% improvement in recognition accuracy. This is because the semisupervised approach utilizes richer sample information compared to the supervised approach. To break the performance bottleneck of the teacher–student structure, our solution expands on the teacher–student structure by weakening the coupling impact of this structure by introducing an independent new model. Compared to the teacher–student, the trimodel structure has good performance advantages in all partitions. Our method achieves the highest performance improvement of 8.32% at L = 300 and the smallest performance improvement of 3.84% at L = 1000 compared to the supervised method. In the above comparison between semisupervised and supervised methods, when the number of labeled samples becomes larger, the difference in the amount of information between semisupervised and supervised methods decreases, so the recognition performance gap gradually narrows.

In addition, we also plot the t-SNE diagram in Fig. 13 to visualize the performance of our semisupervised and supervised approaches. As shown in Fig. 13, our method has a larger between-class distance and less between-class confounding, which indicates that the network trained by the trimodel method outperforms the network trained by the supervised method. This is consistent with the experimental results in Table V.

Based on the above discussion, each of the modules used in our approach is productive. The comparison of the metrics shows that the incorporation of the modules we designed all achieved an improvement in the overall model recognition performance.

1) The use of the ACN network plays the most significant role in improving the performance of the model in each experimental partition.
2) The 2-D positional embedding module enhances the effectiveness of the self-attention mechanism on the image targets to a certain extent and achieves an enhancement effect on the model performance.
3) Our proposed trimodel semisupervised training method achieves significant performance improvement compared to the supervised method, and the introduction of the classmate model enhances the performance of the traditional teacher–student SSL method.

TABLE VI
RECOGNITION ACCURACY OF OUR METHOD, LADDER NETWORK, Π MODEL, TEMPORAL ENSEMBLING MODEL, MEAN TEACHER, DUAL STUDENT, AND UDA MODEL WITH DIFFERENT PARTITIONS OF THE TRAINING DATASET

Methods	L:300	L:400	L:500	L:600	L:800	L:1000
Ladder Network	71.90	78.03	80.73	84.10	88.37	91.55
Π Model	72.09	79.96	81.05	87.63	90.20	92.62
Temporal Ensembling	74.58	81.58	85.80	88.29	91.23	93.88
Mean Teacher	77.68	84.44	88.42	89.91	90.59	92.92
Dual Student	81.95	85.54	89.11	91.34	92.54	93.76
UDA Model	83.06	86.72	89.77	92.03	93.12	94.52
Ours	**85.69**	**88.82**	**91.59**	**93.73**	**94.72**	**95.92**

TABLE VII
COMPARISON OF THE RECOGNITION ACCURACY OF DIFFERENT POSITIONAL ENCODING METHODS IN IMAGE SELF-ATTENTION

Position Encoding	L:300	L:400	L:500	L:600	L:800	L:1000
Fixed [43]	82.18	84.79	88.51	91.25	92.01	93.34
Sinusoidal [48]	83.51	86.13	89.67	92.26	92.88	94.05
CoordConv [45]	82.13	85.68	88.98	90.23	91.59	93.02
Relative [44]	85.10	88.79	91.63	93.52	94.50	95.88
2D-LocalRNN	**85.69**	**88.82**	**91.59**	**93.73**	**94.72**	**95.92**

E. Comparison With Other Semisupervised Methods

On the MSTAR dataset, we validate various semisupervised recognition methods, including ladder network [30], Π model [31], temporal ensembling model [32], mean teacher [33], dual student [35], and unsupervised data augmentation (UDA) model [52]. First, each comparative method is introduced as follows.

1) *Ladder network:* This method is the earliest teacher–student structure approach to achieve consistency regularization by adding Gaussian noise to each neural network layer.
2) Π *model:* Unlike the perturbation used in the ladder network, the Π model creates two random augmentations of a sample for both labeled and unlabeled data and introduces random perturbation between the two networks via dropout.
3) *Temporal ensembling:* This method incorporates EMA calculation based on random data augmentation, where the output predictions of the network are calculated by EMA and subjected to the consistency constraint.
4) *Mean teacher:* This method applies EMA calculation to the network for the first time. The student model is calculated by EMA to obtain the teacher model, thus adding perturbations between the two networks.
5) *Dual student:* This method abandons the teacher–student structure and uses two models with independent initialization. The concept of stable sample is also introduced to establish the stability constraint between the two models, avoiding the performance bottleneck problem of the teacher–student structure.
6) *UDA model:* This method follows the consistency regularization framework and replaces simple noise addition with high-quality data enhancement methods such as AutoAugment, RandAugment, etc., thus improving the recognition performance of the model.

Table VI shows the comparison results of the recognition accuracy of the selected six comparison methods with our method. It can be seen that our method achieves the best recognition accuracy in all the training dataset partitions, which reflects the effectiveness and superiority of our algorithm.

Our method has huge performance improvement compared to the ladder network. The reason is that the ladder network only uses simple Gaussian noise to add perturbations between networks and uses a feature extraction network that is not capable of fully extracting features from images. Compared with the Π model, the temporal ensembling model, and the mean teacher model, our method also has significant improvement in recognition accuracy, especially in the partitions with few labeled samples. This is because these methods use a student–teacher structure, and the strong coupling between the two models makes it difficult to continue improving their performance. Our method, however, weakens this coupling by adding a parallel training model and has a much superior training effect. The dual-student approach replaces the teacher model with another student model in order to eliminate the impact of the teacher–student structure's characteristics on recognition performance and to create more stable constraint between the two models. In contrast, on the one hand, our method maintains the excellent performance of the traditional teacher–student structure and designs weakening the confirmation bias on its basis; on the other hand, our method's feature extraction network can extract both local and global information in the image target, which means a stronger information extraction ability. Therefore, our method has better performance. In addition, the UDA model applies the data augmentation approach commonly used in supervised learning to semisupervised learning, proposes the TSA method to deal with the data imbalance between labeled and unlabeled samples, and improves the consistency regularization from the perspective of data. In contrast, our method optimizes from the perspective of networks by enhancing the feature extraction of the networks while improving the training process of the network, which acquires a weak recognition performance advantage.

F. Discussions

1) Comparison of Different Positional Embedding Methods: Table VII shows the performance of our 2D-LocalRNN positional encoding method compared with several other methods. It can be seen that our 2D-LocalRNN method achieves the highest recognition accuracy under each partition, especially with an average accuracy advantage of 3.14% compared to the CoordConv method. The reason is that CoordConv, although not permutation equivariant, does not satisfy translation equivariance, which is a required property when processing images. The fixed position encoding method used in Transformer [43] is one of the simplest position encoding methods, which is based on the principle that the position encoding is obtained by learning during the training of the model. The validity of the positional information obtained in this way is relatively insufficient, so its

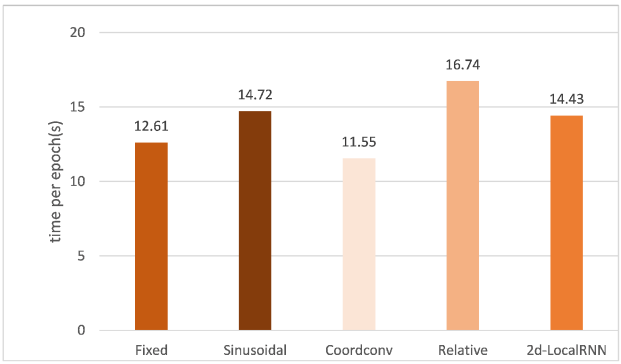

Fig. 14. Training time per epoch with the five positional encoding methods.

recognition performance is similar to that of CoordConv and lower than that of our method. The sinusoidal position encoding method proposed in [48] is a special form of fixed position encoding. The sinusoidal encoding approach can achieve higher effectiveness of position information extraction, and therefore, its recognition accuracy is higher than fixed position encoding method. However, its effectiveness is still inferior to that of the relative position encoding method, and the recognition accuracy is lower than the Relative method and our method. At the same time, our method does not have a significant advantage in recognition accuracy compared to the Relative method. The Relative method is able to extract accurate and valid position information with a high computational effort, so the performance is not weaker than our method.

Next, we compare the efficiency of different positional encoding methods. We choose a partition with L = 500 for the experiment and count the time to train one epoch under each positional encoding method. All the algorithms are implemented in Pytorch 1.8.0 and trained and tested on an Nvidia GeForce RTX 2080Ti graphics card with 11 GB of video memory. We use Cuda9.1 to speed up GPU computing. The results are shown in Fig. 14. It can be seen that the training time per epoch of our method is 14.43 s, which has a great advantage in training efficiency compared with 16.74 s of the Relative method. This is because the processing of short sequences with RNNs to obtain position information in our method, which is easy to achieve parallel computation and fast processing speed, thus achieving efficiency improvement while maintaining recognition performance. The fixed position encoding uses model learning to obtain position encoding with a shorter training time, which has a slight efficiency advantage over our method, but has a greater disadvantage in recognition accuracy. The sinusoidal position encoding method is comparable to our method in terms of training efficiency due to the more complex encoding strategy. In contrast, CoordConv has a simpler computational process, and although its recognition accuracy is not high, the training time per epoch is the shortest at 11.55 s.

2) Hyperparameters in the Loss Function: In our training method, the loss function of the student model consists of three components, with two hyperparameters a and b to adjust the proportion of the three components. The following section concentrate on the training effect of the weak consistency loss between the classmate and student models in terms of the proportion of the loss function. We set $a = 100$ and b is calculated as

$$b = \frac{E - e}{E} * c. \tag{26}$$

We set c to 0, 1, 50, 100, and 1000 and test under each partition, respectively. Fig. 15 shows the effect of different c on the recognition accuracy under each partition. When $c = 0$, it means the classmate model is not constrained to the student model, and the recognition accuracy is now low. When $c = 1$ and $c = 50$, this part of the loss improves the student model's training to some extent, and the improvement peaks situates at $c = 50$. At this point, the classmate model lessened the coupling effect between the student and teacher models, and the best balance between the three models is obtained. And when c is further increased to 100 and 1000, the classmate model plays a side effect on the training of the teacher–student model, and the recognition accuracy in multiple partitions is even lower than the case of $c = 0$. This is because, if the loss function's WCC component is too large, the classmate model will gradually overtake the teacher model's dominant position in consistency training and the classmate model's lackluster performance will seriously mislead the student model's growing learning trajectory. In addition, different c can have a significant impact on the recognition accuracy when the labeled samples are small. As the number of labeled samples increases, the recognition ability of the ACN improves. Therefore, the influence of the classmate model on the teacher–student model progressively fades, and the accuracy change curve due to changing the value of c tends to be flat. Based on the above analysis, setting $c = 50$ allows the addition of the classmate model to optimize the teacher–student model to the best performance.

TABLE VIII
RECOGNITION ACCURACY OF OUR METHOD AT DIFFERENT a WHEN L = 600

a	0	1	50	100	1000	10000
Accuracy	88.23	90.61	92.96	93.73	92.23	91.56

Then, we set $c = 50$ and set the hyperparameters a to 0, 1, 50, 100, 1000, 10 000, and experiment under the partition of L = 600. The recognition accuracies of our method under different a are shown in Table VIII. It can be seen that the recognition accuracy is lowest at $a = 0$. As a increases, the recognition accuracy rises and reaches its highest at $a = 100$. This illustrates the effectiveness of the loss function l_c for model training. This loss function is the consistency constraint of the teacher model on the student model, and if its weight is too large, the effectiveness of the classification loss l_1 will be affected, which will lead to the degradation of the final recognition performance.

In addition, the WCC part of the loss function of the classmate model is weighted by the hyperparameter d. The role of this part of the loss is to assist the classification loss to optimize the training of the classmate model. We set d to 0, 1, 50, 100, 1000, and 10 000 and evaluate the recognition accuracy of the

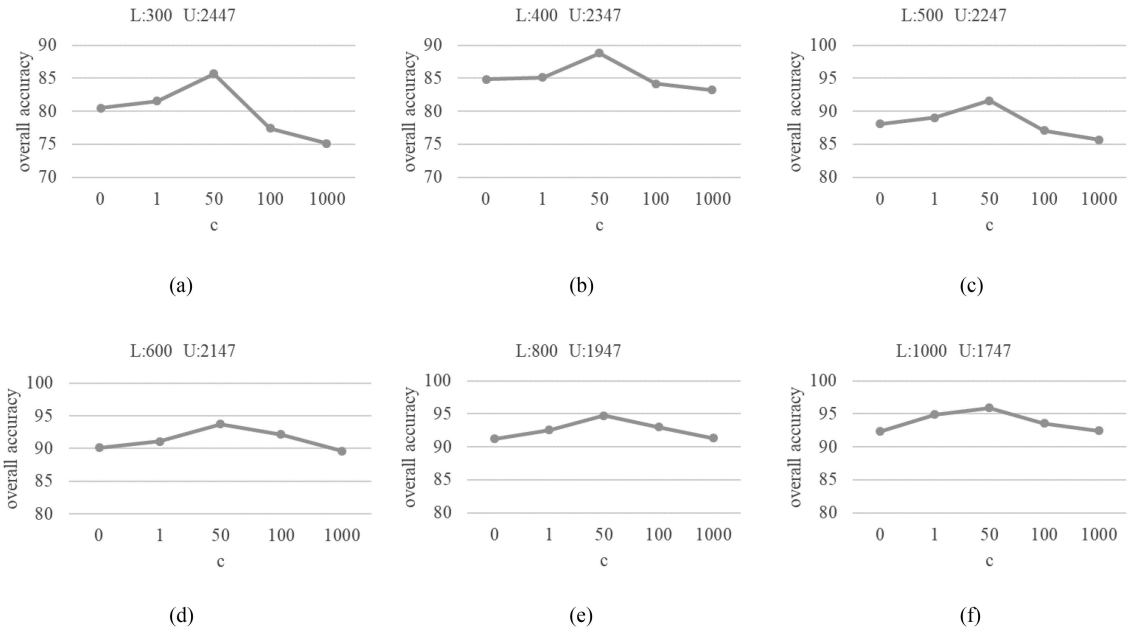

Fig. 15. Recognition accuracy of our method for different selections of c under each partition. (a) L = 400. (b) L = 600. (c) L = 800. (d) L = 1000. (e) L = 400. (f) L = 600.

TABLE IX
RECOGNITION ACCURACY OF THE CLASSMATE MODEL AT DIFFERENT d WHEN L = 600

d	0	1	50	100	1000	10000
Accuracy	87.89	88.05	88.78	89.62	87.46	86.52

classmate model under L = 600 partition. The experimental results are shown in Table IX. It can be seen that the recognition accuracy is lowest at $d = 0$. The recognition accuracy rises and reaches its highest at $d = 100$ as d increases. This reflects the optimization of the WCC part for the training of the classmate model. As the weights increase further, the WCC partially affects the dominant role of classification loss in the loss function, so the recognition performance decreases sharply. Next, we verify the effect of the size of the credibility threshold t in the WCC on the training effect. The significance of the credibility threshold t is to balance the strength of the weak consistency constraint imposed by the classmate model on the student model, and it cannot be too large or too small. The information entropy of the output is calculated as follows:

$$H(f(\theta), x_i) = -\sum_{k=1}^{N} q_k^i \log_2(q_k^i + \sigma). \tag{27}$$

According to the information entropy property, the choice of the size of the credibility threshold t is mainly related to the number of dataset categories N. The maximum possible value of H is $\log_2 N$. For instance, the output information entropy in the MSTAR dataset ranges from 0 to 3.32 when $N = 10$. The optimal t size for this dataset can be found by conducting several sets of experiments on t values in this range. Therefore, we set t to 0.5, 1, 1.5, 2, and 2.5 and tested it under the partition L = 600. As shown in Fig. 16, the network is trained optimally

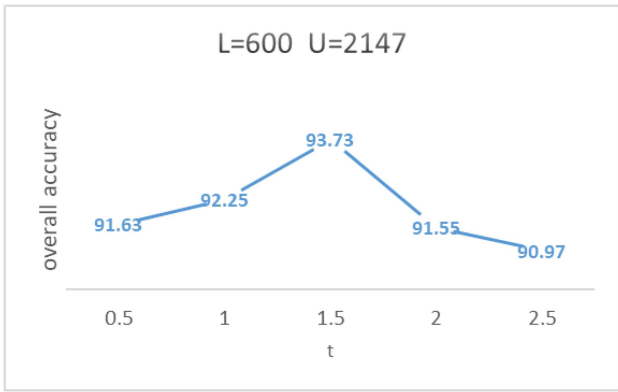

Fig. 16. Recognition accuracy with different credibility thresholds t at L = 600.

at $t = 1.5$. The final performance of the teacher model decreases when t is increased or decreased. This is because the threshold t determines the strength of the constraint played by the WCC: If it is too small, the role played by the classmate model on the student model at the early stage of training will not be sufficient to affect the strong coupling between the teacher and student models; if it is too large, the reliability of the output of the classmate model will be significantly reduced and will play a less effective role in guiding the student model and may even mislead the student model to a large extent. Based on the above discussion, choosing an appropriate credibility threshold t can effectively improve the training effect of the trimodel method.

V. CONCLUSION

Inspired by the AM and semisupervised learning mechanisms in the human cognitive process, a trimodel-consistency-based

semisupervised method was proposed that utilizes an attention-augmented convolutional model for SAR target recognition. Specifically, in the feature extraction part, we combined the self-attentional mechanism with a standard convolutional network, which effectively improves the ability of the network to extract both the local and global features of the image. Furthermore, we designed a 2D-LocalRNN positional embedding module to efficiently extract 2-D position information of images utilizing the properties of RNNs, which enhances the performance of the self-attention module. In the trimodel-consistency-based semisupervised training part, we introduced an independently initialized classmate model to the traditional teacher–student structure to weaken the confirmation bias and designed a weak consistency regularization method to adjust the influence of the classmate model on the teacher–student model. Our proposed feature extractor and semisupervised training method successfully utilized feature information from a large number of unlabeled samples and only a small number of labeled samples, while alleviating performance drawbacks caused by high coupling between teacher and student models, leading to improved model recognition performance. Experiments on the MSTAR dataset demonstrated the effectiveness of our proposed method. The recognition accuracy of our method reached 85.69% when the labeled samples were only 1/10 of the sample dataset and 95.92% when the labeled samples were 1/3 of the sample dataset. The recognition performance is shown to exceed that obtained by classical consistency-regularization-based semisupervised recognition methods, such as mean teacher and dual student, for each dataset partitioning case. Future work will aim to optimize our developed model and evaluate its performance–complexity tradeoff on a range of datasets to further demonstrate its utility as a benchmark resource for the DNN and SAR research community.

REFERENCES

[1] F. Gao, Y. Yang, J. Wang, J. Sun, E. Yang, and H. Zhou, "A deep convolutional generative adversarial networks (DCGANS)-based semi-supervised method for object recognition in synthetic aperture radar (SAR) images," *Remote Sens.*, vol. 10, no. 6, 2018, Art. no. 846.

[2] G. Wang, S. Tan, C. Guan, N. Wang, and Z. Liu, "Multiple model particle filter track-before-detect for range ambiguous radar," *Chin. J. Aeronaut.*, vol. 26, no. 6, pp. 1477–1487, 2013.

[3] F. Ma, F. Gao, J. Sun, H. Zhou, and A. Hussain, "Weakly supervised segmentation of SAR imagery using superpixel and hierarchically adversarial CRF," *Remote Sens.*, vol. 11, no. 5, 2019, Art. no. 512.

[4] F. Ma, F. Zhang, D. Xiang, Q. Yin, and Y. Zhou, "Fast task-specific region merging for SAR image segmentation," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5222316.

[5] F. Ma, F. Zhang, Q. Yin, D. Xiang, and Y. Zhou, "Fast SAR image segmentation with deep task-specific superpixel sampling and soft graph convolution," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2021, Art. no. 5214116.

[6] H. Chen, F. Zhang, B. Tang, Q. Yin, and X. Sun, "Slim and efficient neural network design for resource-constrained SAR target recognition," *Remote Sens.*, vol. 10, no. 10, 2018, Art. no. 1618.

[7] F. Zhang, X. Yao, H. Tang, Q. Yin, Y. Hu, and B. Lei, "Multiple mode SAR raw data simulation and parallel acceleration for Gaofen-3 mission," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 6, pp. 2115–2126, Jun. 2018.

[8] J. Ding, B. Chen, H. Liu, and M. Huang, "Convolutional neural network with data augmentation for SAR target recognition," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 3, pp. 364–368, Mar. 2016.

[9] Y. Guo, Z. Sun, R. Qu, L. Jiao, F. Liu, and X. Zhang, "Fuzzy superpixels based semi-supervised similarity-constrained CNN for PolSAR image classification," *Remote Sens.*, vol. 12, no. 10, 2020, Art. no. 1694.

[10] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi, "Deep feature extraction and classification of hyperspectral images based on convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 10, pp. 6232–6251, Oct. 2016.

[11] M. Amrani and F. Jiang, "Deep feature extraction and combination for synthetic aperture radar target classification," *J. Appl. Remote Sens.*, vol. 11, no. 4, 2017, Art. no. 042616.

[12] J. Zhao, W. Guo, S. Cui, Z. Zhang, and W. Yu, "Convolutional neural network for SAR image classification at patch level," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2016, pp. 945–948.

[13] S. Chen and H. Wang, "Sar target recognition based on deep learning," in *Proc. Int. Conf. Data Sci. Adv. Analytics*, 2014, pp. 541–547.

[14] R. Min, H. Lan, Z. Cao, and Z. Cui, "A gradually distilled CNN for SAR target recognition," *IEEE Access*, vol. 7, pp. 42190–42200, 2019.

[15] F. Gao, T. Huang, J. Sun, J. Wang, A. Hussain, and E. Yang, "A new algorithm for SAR image target recognition based on an improved deep convolutional neural network," *Cogn. Comput.*, vol. 11, no. 6, pp. 809–824, 2019.

[16] B. Liu, X. Yu, P. Zhang, X. Tan, A. Yu, and Z. Xue, "A semi-supervised convolutional neural network for hyperspectral image classification," *Remote Sens. Lett.*, vol. 8, no. 9, pp. 839–848, 2017.

[17] M. Yang, X. Bai, L. Wang, and F. Zhou, "Mixed loss graph attention network for few-shot SAR target classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2021, Art. no. 5216613.

[18] L. Wang, X. Bai, C. Gong, and F. Zhou, "Hybrid inference network for few-shot SAR automatic target recognition," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 11, pp. 9257–9269, Nov. 2021.

[19] L. Wang, X. Bai, R. Xue, and F. Zhou, "Few-shot SAR automatic target recognition based on Conv-BiLSTM prototypical network," *Neurocomputing*, vol. 443, pp. 235–246, 2021.

[20] Z. Yue et al., "A novel semi-supervised convolutional neural network method for synthetic aperture radar image recognition," *Cogn. Comput.*, vol. 13, no. 4, pp. 795–806, 2021.

[21] K. Chen, Z. Pan, Z. Huang, Y. Hu, and C. Ding, "Learning from reliable unlabeled samples for semi-supervised SAR ATR," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, 2022, Art. no. 4512205.

[22] C. Wang et al., "Semisupervised learning-based SAR ATR via self-consistent augmentation," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 6, pp. 4862–4873, Jun. 2021.

[23] J. Ji, F. Shao, R. Sun, N. Zhang, and G. Liu, "A TSVM based semi-supervised approach to SAR image segmentation," in *Proc. Int. Workshop Educ. Technol. Training Geosci. Remote Sens.*, 2008, vol. 1, pp. 495–498.

[24] X. R. Zhang, C. Yang, and L. C. Jiao, "Semi-supervised SAR target recognition based on Laplacian regularized least squares classification," *J. Softw.*, vol. 21, no. 4, pp. 586–596, 2010.

[25] X. Yang, Z. Song, I. King, and Z. Xu, "A survey on deep semi-supervised learning," 2021, *arXiv:2103.00550*. [Online]. Available: https://arxiv.org/abs/2103.00550

[26] J. Sietsma and R. J. Dow, "Creating artificial neural networks that generalize," *Neural Netw.*, vol. 4, no. 1, pp. 67–79, 1991.

[27] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. Int. Conf. Learn. Representations*, 2015.

[28] P. Bachman, O. Alsharif, and D. Precup, "Learning with pseudo-ensembles," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 3365–3373.

[29] M. Pezeshki, L. Fan, P. Brakel, A. Courville, and Y. Bengio, "Deconstructing the ladder network architecture," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 2368–2376.

[30] A. Rasmus, M. Berglund, M. Honkala, H. Valpola, and T. Raiko, "Semi-supervised learning with ladder networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 3546–3554.

[31] M. Sajjadi, M. Javanmardi, and T. Tasdizen, "Regularization with stochastic transformations and perturbations for deep semi-supervised learning," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 1171–1179.

[32] S. Laine and T. Aila, "Temporal ensembling for semi-supervised learning," in *Proc. Int. Conf. Learn. Representations*, 2017.

[33] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1195–1204.

[34] T. Miyato, S.-I. Maeda, M. Koyama, and S. Ishii, "Virtual adversarial training: A regularization method for supervised and semi-supervised learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 8, pp. 1979–1993, Aug. 2019.

[35] Z. Ke, D. Wang, Q. Yan, J. Ren, and R. W. Lau, "Dual student: Breaking the limits of the teacher in semi-supervised learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 6727–6735.
[36] I. Bello, B. Zoph, A. Vaswani, J. Shlens, and Q. V. Le, "Attention augmented convolutional networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 3285–3294.
[37] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7794–7803.
[38] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 213–229.
[39] P. Ramachandran, N. Parmar, A. Vaswani, I. Bello, A. Levskaya, and J. Shlens, "Stand-alone self-attention in vision models," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 68–80.
[40] H. Wang, Y. Zhu, B. Green, H. Adam, A. Yuille, and L.-C. Chen, "Axial-DeepLab: Stand-alone axial-attention for panoptic segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 108–126.
[41] M. Zhang et al., "Convolutional neural network with attention mechanism for SAR automatic target recognition," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, 2020, Art. no. 4004205.
[42] F. Ma, F. Gao, J. Sun, H. Zhou, and A. Hussain, "Attention graph convolution network for image segmentation in big SAR imagery data," *Remote Sens.*, vol. 11, no. 21, 2019, Art. no. 2586.
[43] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Representations*, 2021.
[44] P. Shaw, J. Uszkoreit, and A. Vaswani, "Self-attention with relative position representations," in *Proc. North Amer. Chapter Assoc. Comput. Linguistics (NAACL)*, 2018.
[45] R. Liu et al., "An intriguing failing of convolutional neural networks and the CoordConv solution," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 9628–9639.
[46] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie, "Objects in context," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, 2007, pp. 1–8.
[47] N. Parmar et al., "Image transformer," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 4055–4064.
[48] A. Vaswani et al., "Attention is all you need," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6000–6010.
[49] Z. Wang, Y. Ma, Z. Liu, and J. Tang, "R-transformer: Recurrent neural network enhanced transformer," in *Proc. Int. Conf. Learn. Representations*, 2020.
[50] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[51] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," in *Proc. Conf. Workshop Neural Inf. Process. Syst.*, 2016.
[52] Q. Xie, Z. Dai, E. Hovy, T. Luong, and Q. Le, "Unsupervised data augmentation for consistency training," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2020, vol. 33, pp. 6256–6268.

Sifan Yan received the B.S. degree in electronic and information engineering in 2020 from Beihang University, Beijing, China, where he is currently working toward the M.E. degree in signal and information processing.

His current research interests include target recognition and remote sensing image processing.

Yaotian Zhang received the Ph.D. degree in signal and information processing from Beihang University, Beijing, China, in 2010.

He is currently an Associate Professor with the School of Electronic and Information Engineering, Beihang University. His research interests include statistical signal processing, high-resolution radar signal processing, target tracking, image understanding, and robust beamforming.

Fei Gao received the B.S. degree in electrical automation and the M.S. degree in electromagnetic measurement technology and instrument from Xi'an Petroleum Institute, Xi'an, China, in 1996 and 1999, respectively, and the Ph.D. degree in signal and information processing from Beihang University, Beijing, China, in 2005.

He is currently a Professor with the School of Electronic and Information Engineering, Beihang University. His research interests include target detection and recognition, image processing, and deep learning for applications in remote sensing.

Jinping Sun (Member, IEEE) received the M.Sc. degree in communication and electronic system and Ph.D. degree in signal and information processing from Beihang University, Beijing, China, in 1998 and 2001, respectively.

He is currently a Professor with the School of Electronic and Information Engineering, Beihang University. His research interests include statistical signal processing, high-resolution radar signal processing, target tracking, image understanding, and robust beamforming.

Amir Hussain received the B.Eng., M.S. and Ph.D. degrees in electronic and electrical engineering from the University of Strathclyde, Scotland, U.K., in 1990, 2002, and 2006, respectively.

Following postdoctoral and senior academic positions with the University of the West of Scotland, Paisley, U.K., from 1996 to 1998, with the University of Dundee, Dundee, U.K., from 1998 to 2000, and with the University of Stirling, Stirling, U.K., from 2000 to 2018, he joined Edinburgh Napier University, Edinburgh, U.K., as the Founding Head of the Cognitive Big Data and Cybersecurity Research Lab and the Centre for Artificial Intelligence and Data Science. His research interests include cognitive computation, machine learning, and computer vision.

Huiyu Zhou received the B.Eng. degree in radio technology from the Huazhong University of Science and Technology, Wuhan, China, the M.S. degree in biomedical engineering from the University of Dundee, Dundee, U.K., and the Ph.D. degree in ratio technology, biomedical engineering, and computer vision from Heriot-Watt University, Edinburgh, U.K.

He is currently a Professor with the School of Informatics, University of Leicester, Leicester, U.K. His research interests include medical image processing, computer vision, intelligent systems, and data mining.