SAR Target Incremental Recognition based on Features with Strong Separability

Fei Gao, Lingzhe Kong, Rongling Lang, Jinping Sun, *Member, IEEE*, Jun Wang, Amir Hussain, and Huiyu Zhou

***Abstract*—With the rapid development of deep learning technology, many SAR target recognition algorithms based on convolutional neural networks have achieved exceptional performance on various datasets. However, conventional neural networks are repeatedly iterated on a fixed dataset until convergence, and once they learn new tasks, a large amount of previously learned knowledge is forgotten, leading to a significant decline in performance on old tasks. This paper presents an incremental learning method based on strong separability features (SSF-IL) to address the model's forgetting of previously learned knowledge. The SSF-IL employs both intra-class and inter-class scatter to compute the feature separability loss, in order to enhance the linear separability of features during incremental learning. In the process of learning new classes, an intra-class clustering loss is proposed to replace the conventional knowledge distillation. This loss function constrains the old class features to cluster around the saved class centers, maintaining the separability among the old class features. Finally, a classifier bias correction method based on boundary features is designed to reinforce the classifier's decision boundary and reduce classification errors. SAR target incremental recognition experiments are conducted on the MSTAR dataset, and the results are compared with several existing incremental learning algorithms to demonstrate the effectiveness of the proposed algorithm.**

***Index Terms*—incremental learning, SAR target recognition, feature separability, intra-class clustering, bias correction.**

I. INTRODUCTION

SYNTHETIC Aperture Radar (SAR) is capable of acquiring high-resolution images in almost any weather and at any time of day [1]. Recently, automated target recognition of SAR images based on neural network models have achieved remarkable performance on various datasets [2], [3], [4]. However, conventional neural networks are "closed" to the dataset. The learning process of neural networks involves repeated iterations on a fixed training set until convergence, ultimately enabling them to classify all classes in the current task [5], [6]. Once learning a new task, these models will encounter "catastrophic forgetting", which results in a significant decline in performance on the old tasks [7]. In practical applications, SAR images of various targets are usually obtained through multiple detections and batches, rather than all at once [8], [9], [10]. This necessitates that classification models are able to process new samples and even learn new classes while preserving previously learned knowledge. Currently, conventional neural networks cannot meet these practical requirements.

Driven by the demand to handle dynamic data in real-world scenarios, researchers have turned their attention to the field of incremental learning. Fig. 1 illustrates the general process of class incremental learning algorithms, in which the neural network is first trained on a base task with several classes as the base model, followed by the arrival of new tasks containing new classes over time. Then the new model is initialized with the previous model and updated to adapt to new classes. As new tasks arrive sequentially, the neural network can recognize an increasing number of classes, thereby achieving continuous learning.

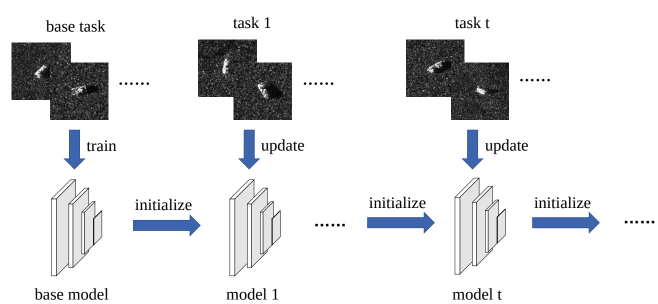


Fig. 1. Incremental learning algorithm flowchart.

Currently, the core challenge faced by incremental learning algorithms is to adapt to new data while preserving previously learned knowledge [11]. The main reasons for the performance degradation on old tasks include weight drift, inter-task confusion, and data imbalance [10], [12], [13]. Neural network weights are iteratively trained to achieve the optimal solution for the current task. While learning a new task, the weights related to old tasks are updated to adapt to new data, leading to a decline in performance on previous tasks. Besides, classes in different incremental learning tasks may be quite similar, which makes it difficult for the model to fully distinguish between them. Data imbalance refers to the situation where, due to the constraints of memory and data security, complete old data is inaccessible in the process of incremental learning,

This work was funded by the National Natural Science Foundation of China under Grant 62371022 and the UK Engineering and Physical Sciences Research Council (EPSRC) under Grant EP/M026981/1, Grant EP/T021063/1, and Grant EP/T024917/. *Corresponding authors: Rongling Lang, and Jinping Sun.*

Fei Gao is with the School of Electronic and Information Engineering,Beihang University, Beijing 100191, China, and also with the BeihangHangzhou Innovation Institute Yuhang, Xixi Octagon City, Hangzhou 310023,China (e-mail: feigao2000@163.com).

Lingzhe Kong, Rongling Lang, Jinping Sun, and Jun Wang are with the School of Electronic and Information Engineering, Beihang University, Beijing 100191, China (e-mail: klz1008@buaa.edu.cn; ronglinglang@163.com; sunjinping@buaa.edu.cn; wangj203@buaa.edu.cn).

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

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

and new data outnumbers the old data. When learning on imbalanced data, the model often exhibits a classification bias towards the majority classes. For incremental learning, this bias manifests as a tendency for the model to predict test samples as newly learned classes, which is referred to as ”task-time bias” [13], [14].

To reduce the forgetting of previously learned knowledge, various incremental learning algorithms have employed different methods, which gives rise to a new problem known as the ”stability-plasticity dilemma” [15]. In this dilemma, the ability of a model to preserve learned knowledge corresponds to its stability, and the ability to learn new knowledge corresponds to its plasticity. For incremental learning algorithms, sorely concentrating on the acquisition of new knowledge leads to catastrophic forgetting, while predominantly preserving learned knowledge impedes the learning of new tasks [12]. Thus, in practical algorithm design, the improvement of stability and plasticity often conflicts, making it challenging to strike a balance between the two.

Additionally, according to [10], the inter-task confusion discussed above is a prominent issue in SAR target incremental recognition. The same target appears different at varied azimuths, while targets of different classes tend to be similar at the same azimuth. As a result, samples from the same class exhibit relatively large intra-class differences, whereas samples from different classes show exactly the opposite characteristic. This makes it more challenging for neural networks to extract features that are easy to classify, resulting in the confusion between different classes. Therefore, we directly focus on the separability of features and propose a novel incremental learning method to address the aforementioned problems. The contributions of this paper are as follows:

(1) A feature separability enhancement method based on intra-class scatter and inter-class scatter. By designing a feature separability loss that minimizes the intra-class scatter and maximizes the inter-class scatter, features of each class are brought closer to their corresponding centers while different classes are separated from each other. As a result, the spatial gap between different classes is enlarged, which reduces the feature confusion.

(2) A method for preserving learned knowledge based on intra-class clustering loss. By constraining the old class features with the intra-class clustering loss during model updating, the proposed algorithm keeps all old class features clustered around their corresponding centers, thus maintaining the feature separability between different classes and avoiding catastrophic forgetting of learned knowledge.

(3) A classifier bias correction method based on boundary features. By saving a predetermined number of boundary features of old classes, the classifier parameters are fine-tuned using these features after each new task is learned. This approach enables the classifier to modify the decision boundary of old classes and mitigate the classification bias towards new classes, which improves the overall SAR target recognition performance.

II. Related Works

A. Incremental Learning

In recent years, with the rapid development of deep learning technology, incremental learning has received widespread attention. Unlike traditional models built on static tasks, incremental learning focuses on dynamic tasks, which require dealing with more complex scenarios, such as the continuous arrival of new data, the temporary storage of old data, and the need for efficient computation [12]. Therefore, incremental learning algorithms need to meet additional requirements. According to [16], these requirements are as follows:

(1) The model should be trainable on data streams.

(2) The model should be able to distinguish all previously learned classes at any stage.

(3) The overall computational resources consumed by the model should be controlled within a relatively reasonable range instead of increasing without limitation as the number of learned classes increases.

Currently, researchers have proposed various incremental learning methods from different perspectives to address the aforementioned needs. The existing incremental learning algorithms can be classified into three categories: regularization, data replay, and bias correction [17].

Regularization aims to restrict the direction of model updates so that the updated model maintains good classification performance on old classes [7]. Knowledge distillation is the most widely used regularization method in incremental learning. The model trained on old classes is used as the teacher network to transfer the classification knowledge to the new model. Based on the knowledge distillation, many effective regularization methods have been developed in recent research [18], [19], [20], [21], [22]. The Less Forgetting Learning proposed by Jung et al. [20] preserves previously learned knowledge by freezing the last layer of the model and penalizing the differences between the activations before the classifier. Douillard et al. [21] designed pooling output distillation to preserve previously learned knowledge. The calculation of pooling output distillation loss not only uses the output logits, but also utilizes the pooling results of the feature maps in intermediate layers of the network to further preserve classification knowledge of old classes. Lee et al. [22] presented global distillation, which utilizes external data to distill knowledge from previous tasks. In SAR target incremental recognition tasks, Li et al. [10] proposed an incremental learning method based on anchored class centers to reduce intra-class differences and increase inter-class differences, helping the model distinguish between new and old classes. Tang et al. [15] put forward a knowledge distillation framework based on multiple old task models to reduce cumulative errors in incremental learning and better preserve learned knowledge.

Data replay methods save a small number of samples for each old class, which are used to ”review” the learned knowledge while learning new tasks [12]. Many studies have proven that data replay is a simple and effective way for mitigating the forgetting of learned knowledge [23], [24], [25], [26], [27]. The key to data replay lies in the selection method of exemplars. Rebuffi et al. [16] proposed an exemplar selection method based on the herding algorithm. Dang et al. [26] put forward an exemplar selection method based on the class boundaries for SAR images, with a priority given to samples located in overlapping and marginal regions. In addition, Shin et al. [27] introduced a generative adversarial network to generate old class samples for replay, further reducing the algorithm's dependence on old data to address stricter constraints of memory and data security.

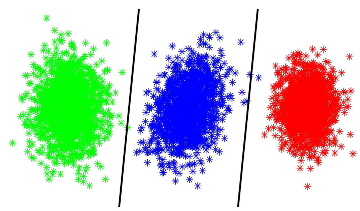
Bias correction aims to improve task-time bias in incremental learning for better recognition performance. Existing research mainly includes methods such as data balancing and weight correction [28], [29], [30], [31]. Belouadah and Popescu [14] proposed to rectify the network predictions based on the saved certainty statistics of previous task predictions to counteract the impact of task-time bias. Castro et al. [29] introduced an additional balanced training phase at the end of each incremental learning phase to prevent task-time bias. Zhao et al. [30] put forward to balance the weights of the classifier parameters to mitigate classification bias. For SAR images, Huang et al. [31] introduced a weight correction method based on a memory-enhanced module to extract typical representations from old class weights and balance the inductive bias between new and old classes.

B. Feature Distribution in Incremental Learning

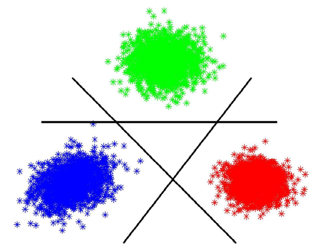
When performing target recognition tasks, neural networks typically use linear classifiers to classify extracted high-dimensional features, which enables the training process to be regarded as learning to map images of different classes to linearly separable features. Similarly, preserving the knowledge of old classes in incremental recognition can also be considered as maintaining the linear separability of old class features. Consequently, the strength of linear separability of features during incremental learning process is an important factor affecting the recognition performance.

Linear separability is a spatial distribution property of point sets. In two-dimensional space, if there exists a line on the plane such that one point set lies on one side of the line while the other point set lies on the other side, then these two sets of points are linearly separable. The linear separability of multiple point sets is more complex. For multiple point sets in a two-dimensional plane, if there exists a set of lines that can separate the different sets of points, then these point sets are linearly separable. This property can be extended to higher dimensions by using hyperplanes instead of lines [32], [33].

Moreover, there are various cases of linear separability. The difficulty of classifying linearly separable point sets belonging to different cases also varies. Fig. 2 shows two common scenarios of linear separability, but there is a significant difference between these two situations. Specifically, the three classes in Fig. 2(a) only satisfy the basic requirement of linear separability: there exists a set of lines that can completely separate different classes. On the other hand, the classes in Fig. 2(b) satisfy an additional condition: any class is linearly separable from all other classes. This means that the discussion about the classification boundary between the red cluster and the green cluster in Fig. 2(a) cannot be separated from the blue cluster, as the blue cluster cannot be linearly separated from both the red and green clusters simultaneously. All three classes must be considered as a whole. In contrast, each class in Fig. 2(b) can be considered individually in terms of a shared classification boundary with all remaining classes, without affecting the classification between the remaining classes. Therefore, the scenario depicted in Fig. 2(a) has only one possible classification, making it a more challenging situation, while the scenario shown in Fig. 2(b) has multiple classification options, making different classes easier to distinguish. By quantitatively measuring the linear separability of high dimensional features extracted by the feature extractor, we can gain insight into the difficulty level of classifying these features and help the model extract features with stronger linear separability. However, we find it difficult to provide a quantitative description of linear separability, both in terms of its definition and classification strategy.



(a)



(b)

Fig. 2. Comparison of two linearly separable scenarios. (a) Linearly separable case with only one possible classification. (b) Linearly separable case with multiple classification options.

Although it is currently difficult to quantitatively measure linear separability, it can be inferred that the linear separability in the case shown in Fig. 2(b) surpasses that in Fig. 2(a). The main difference between the two is the relative position of the cluster centers and the aggregation degree of each cluster towards the corresponding center. In Fig. 2(b), the centers of the point clusters are further apart from each other and the degree of aggregation towards the center is higher. Therefore, by controlling the relative positions of the centers of different classes and the intra-class feature distribution of each class, the feature extractor can extract a set of features with strong linear separability.

III. Methods

The structure of incremental learning based on strong separability features (SSF-IL) is illustrated in Fig. 3. The algorithm mainly consists of three parts: the input data streams, the model updating strategy, and the neural network model. The neural network model comprises a feature extractor composed of convolutional layers and a classifier composed of fully connected layers.

The model is first trained on a base task. The feature extractor maps input images into features of 64 dimensions, and the classifier computes the predictions. To optimize the model, the cross-entropy loss with label smoothing is used as the classification loss. Both intra-class and inter-class scatter of the features are calculated to obtain the feature separability loss,

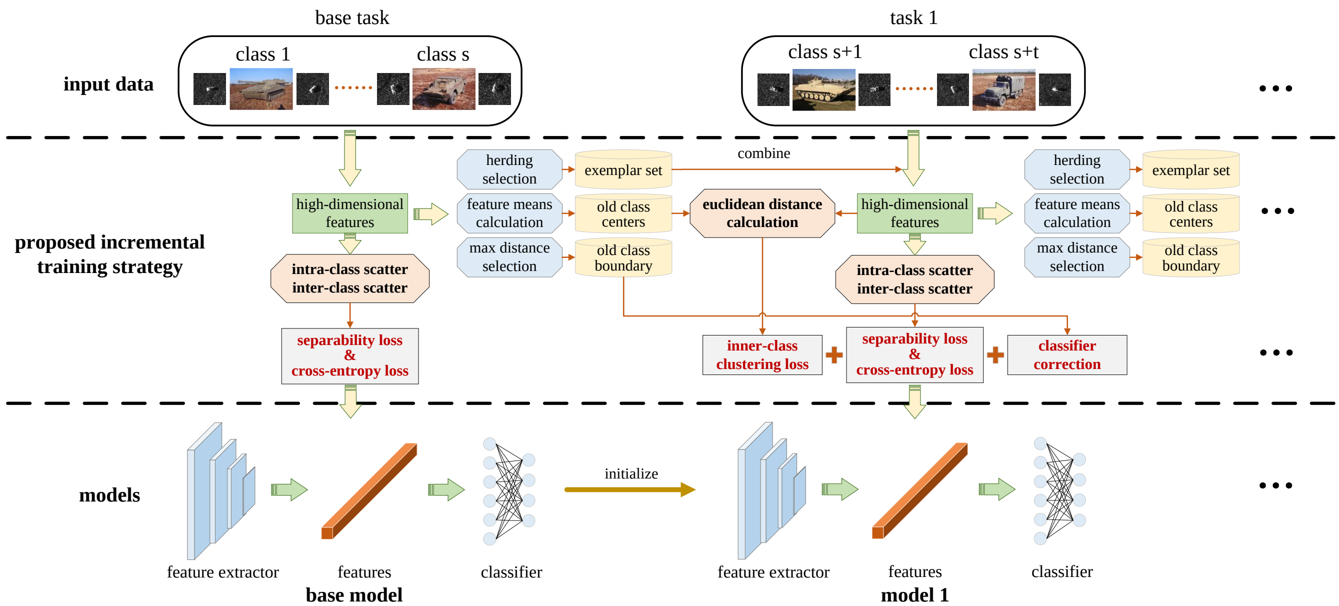


Fig. 3. The structure of the proposed SSF-IL algorithm.

which reinforces the linear separability of different classes. After training, exemplar images are selected with the herding algorithm, and the feature mean of each class is calculated as the class center and saved. The boundary features, i.e., features with the maximum Euclidean distance from the class centers, are also saved. When learning new classes, the neural network takes exemplar images and new class images as input data. In addition to the cross-entropy loss and the feature separability loss, an intra-class clustering loss is designed to preserve the learned knowledge. It constrains the old class features to be around the fixed class centers by reducing the Euclidean distance between them, thus maintaining the linear separability of old class features and avoiding catastrophic forgetting. Finally, the saved boundary features are used to fine-tune the classifier parameters to correct the decision boundary and reduce the classification bias towards new classes. The proposed methods and the overall implementation process of the algorithm are described in detail in the following subsections.

A. Feature Separability Loss

In classification problems, the linear separability of high-dimensional features can significantly affect the results. By leveraging the neural network to map the input images to strongly linearly separable features, the confusion between different classes can be effectively reduced. According to [34], we utilize the eigenvalues of the feature covariance matrix to describe the dispersion of the features.

Taking two-dimensional data as an example, Fig. 4 shows the relationship between the spatial distribution of the features and the eigenvalue-eigenvector pairs of feature covariance matrix. The covariance matrix of two-dimensional data is also two-dimensional and has two sets of eigenvalue-eigenvector pairs. In Fig. 4, the cluster of points is the two-dimensional data, and its two sets of eigenvalue-eigenvector pairs are $\left(\lambda_1, \vec{q}_1\right)$ and $\left(\lambda_2, \vec{q}_2\right)$ respectively. Assuming $\lambda_1 > \lambda_2$ and starting from the center of the two-dimensional data, with the orientation of the eigenvectors as the direction and the magnitude of the eigenvalues as the length, we draw two vectors: $\frac{\lambda_1}{\left|\vec{q}_1\right|}\vec{q}_1$ and $\frac{\lambda_2}{\left|\vec{q}_2\right|}\vec{q}_2$, which correspond to the red and the green arrows. It can be seen that the eigenvectors point in the directions of the spatial dispersion of the data, and the magnitude of the corresponding eigenvalues is positively correlated with the degree of dispersion in those directions. Hence, we employ the magnitude of eigenvalues of the covariance matrix to quantitatively describe the dispersion degree of high-dimensional features.

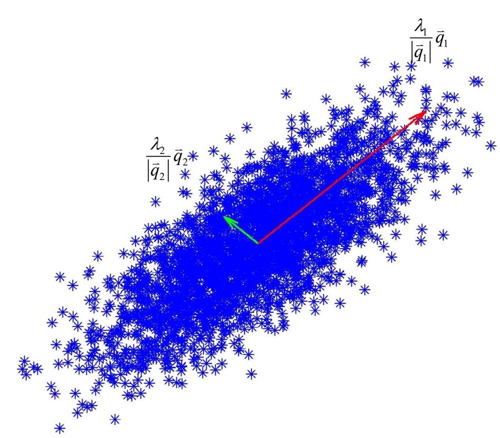


Fig. 4. The relationship between the eigenvalue-eigenvector pairs of the covariance matrix and the data distribution.

(1) intra-class scatter

For high-dimensional features belonging to the same class, the eigenvalues of the covariance matrix represent the dispersion degree of the features in all directions. We only need to focus on the direction with the largest dispersion to control the intra-class distribution. Thus, the largest eigenvalue of the covariance matrix is used as the intra-class scatter.

Assuming that $X_i = \{x_1, x_2, \cdots, x_{N_i}\}$ are samples of class y_i and $f_\theta : \chi \to \mathbb{R}^n$ is the neural network, we first calculate the feature mean μ_i of each class as the class center:

$$\mu_i = \frac{1}{N_i} \sum_{j=1}^{N_i} f_\theta(x_j) \tag{1}$$

Then, the feature covariance matrix S_i for each class is calculated respectively:

$$S_i = \frac{1}{N_i - 1} \sum_{j=1}^{N_i} (f_\theta(x_j) - \mu_i)(f_\theta(x_j) - \mu_i)^T \tag{2}$$

We add all covariance matrices together to obtain the total covariance matrix S_w:

$$S_w = \sum_{i=1}^{C} S_i \tag{3}$$

where C is the current total number of classes. Finally, the maximum eigenvalue λ_w of the total covariance matrix is calculated as the total intra-class scatter.

(2) inter-class scatter

During the process of minimizing intra-class scatter, the features belonging to the same class gradually converge towards the class center. By simply keeping class centers away from each other, the overall distance between different classes can be increased. Therefore, the inter-class scatter is calculated using the center of each class.

Assuming that the set of feature means calculated using Eq.1 is denoted as $\{\mu_i\}$, the covariance matrix S_b of $\{\mu_i\}$ is calculated as:

$$S_b = \frac{1}{C - 1} \sum_{i=1}^{C} (\mu_i - \bar{\mu})(\mu_i - \bar{\mu})^T \tag{4}$$

where C is the total number of classes and $\bar{\mu}$ is the mean of $\{\mu_i\}$:

$$\bar{\mu} = \frac{1}{C} \sum_{i=1}^{C} \mu_i \tag{5}$$

Similarly, the magnitude of each eigenvalue of S_b represents the spatial dispersion of all class centers in the direction of corresponding eigenvector. According to [34], for the C classification problem, the $(C-1)th$ largest eigenvalue λ_b is used as the inter-class scatter.

By combining the intra-class scatter and the inter-class scatter, a loss function is constructed to enhance the feature separability:

$$L_{separability} = \sqrt{\lambda_w} + 1 \Big/ \sqrt{\lambda_b} \tag{6}$$

The objective of this loss function is to minimize intra-class scatter λ_w and maximize inter-class scatter λ_b, hence λ_b is placed in the denominator. Since λ_w and λ_b have the same unit as the square of the features, they are square rooted to avoid the influence of squaring on optimization. Optimizing the feature separability loss ensures that centers of different classes are far apart from each other and the features belonging to the same class are gathered towards the center. As a result, high-dimensional features with strong separability are obtained.

B. Intra-class Clustering Loss

Currently, most incremental learning algorithms use conventional knowledge distillation to transfer classification knowledge of the old model to the new model. The logits output by the old model are used as soft labels to calculate the distillation loss together with the predictions of the new model. Optimizing the distillation loss function can narrow the gap between the outputs of the old and new models for old class samples, thereby preserving the classification ability for old classes on the new model. However, simply constraining the logits can pose a challenge in balancing the model's plasticity and stability. Strong constraints may effectively preserve the learned knowledge, but also result in the new model's output excessively resembling that of the old model, making it difficult to adapt to new classes, which is unacceptable for incremental learning algorithms. Loosening the constraints can facilitate better adaptation to the new knowledge, but may also cause significant forgetting of the previously learned knowledge. This contradiction makes conventional knowledge distillation unable to meet the needs of incremental learning models.

To achieve a better balance between the model's plasticity and stability in incremental learning, we eschew conventional knowledge distillation and instead devise an objective function that forces the features of the old classes to be around their saved class centers to preserve the model's classification ability for the old classes. With the feature separability loss as a constraint, the saved centers of old classes exhibit strong linear separability, enabling the features constrained around them to be distinguished from features of other classes. Such constraint does not necessitate consistency in the output of the old and new models for old class samples, but rather restricts the deviation range of old class features. This not only provides strong constraints for old classes to reduce the forgetting of learned knowledge, but also prevents the outputs of the new model from converging with those of the old model. As a result, the stability and plasticity of the model are well balanced.

We refer to the designed objective function as intra-class clustering loss. During a certain incremental learning phase, the new class is denoted as y_i, with training samples $X_i = \{x_1, x_2, \cdots, x_{N_i}\}$. The neural network model $f_\theta : \chi \to \mathbb{R}^n$ trained in this phase is used to extract features of the training samples, and the feature mean μ_i of class y_i is calculated and saved as the class center with Eq.1. Then the average distance between the features and the class center is calculated as:

$$d_i^{avg} = \frac{1}{N_i} \sum_{j=1}^{N_i} \| f_\theta(x_j) - \mu_i \|_2^2 \tag{7}$$

In the next incremental learning phase, the old class centers saved in all previous learning phases are denoted as

$\{\mu_1, \mu_2, \cdots, \mu_t\}$. The average distance of each old class between the features and their corresponding class center is $\{d_1^{avg}, d_2^{avg}, \cdots, d_t^{avg}\}$. Samples from old classes are denoted as $\phi_i = \{e_1, e_2, \cdots, e_k\}, i = 1, 2, \cdots, t$, and the current neural network model is $f'_\theta : \chi \to \mathbb{R}^n$. Around each old class center, a target area for feature aggregation is set with a radius of d_i^{avg}/γ. γ is a hyperparameter, where the larger it is, the smaller the target area, and the stronger the constraint on features. The distance between the features of old class samples and the target area is calculated as:

$$d_i = \sum_{j=1}^{k} \left(\| \mu_i - f'_\theta (e_j) \|_2^2 - \frac{d_i^{avg}}{\gamma} \right), i = 1, 2, \cdots, t \quad (8)$$

As the sum of the squared Euclidean distances between all features of class y_i and the target area, d_i equally reflects the proximity of each feature belonging to class y_i to its center. Therefore, it can be regarded as a measure of the overall aggregation level of class y_i. We use the sum of d_i as the intra-class clustering loss of all old classes:

$$L_{cluster} = \sqrt{\sum\nolimits_{i=1}^{t} d_i} \quad (9)$$

The sum of all d_i equally reflects the proximity of all old class features to their corresponding centers. Since the unit of d_i is the same as that of the square of the features, the square root is taken after the summation to avoid the influence of squaring on optimization.

Fig. 5 illustrates the process of the intra-class clustering. As the model learns new classes, parameters deviating from the optimal solution for old classes leads to the feature confusion between different classes. The intra-class clustering loss pulls old class features towards the saved class centers and ultimately constrains the features to be around the class centers. In this way, the linear separability of old class features is well maintained during model updating, which avoids the feature confusion between different classes.

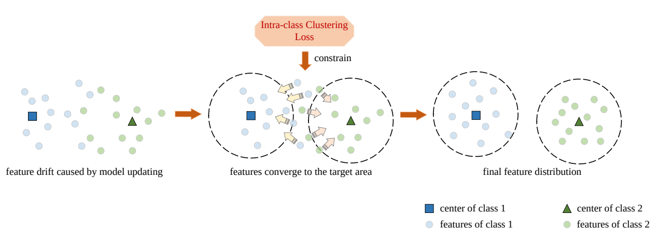


Fig. 5. Intra-class clustering process.

C. Classifier Bias Correction based on Boundary features

Bias correction aims to mitigate the classification bias of the model towards new classes. Existing incremental learning algorithms have proposed several solutions to this problem. For example, samples are taken from new classes and combined with old class exemplars to create a balanced training set for parameter fine-tuning after model updating [13], [29]. The relative importance of old and new classes is balanced by weighting the classifier parameters related to the new classes [30]. However, reusing exemplar samples during the fine-tuning stage after training cannot supplement the model with additional knowledge of old classes, and weighting the parameters related to new classes may affect the model's classification performance for them. These methods do not address the needs of both old and new classes.

We propose a classifier bias correction method based on boundary features to address these problems. Our algorithm saves a fixed number of boundary features to correct the classifier parameters. The bias towards new classes is essentially the confusion between the classification boundaries of old and new classes. By fine-tuning the classifier parameters with boundary features from both old and new classes, the decision boundary of the classifier is strengthened for both types of classes. Additionally, our algorithm constrains old class features around fixed class centers through the intra-class clustering loss, minimizing the feature drift. As a result, the saved boundary features of old classes continue to describe the boundary of the corresponding class even after the model is updated. Furthermore, features extracted by the neural network are one-dimensional tensors, so saving a small number of boundary features hardly increases the storage cost.

After each incremental learning phase, the trained neural network model $f_\theta : \chi \to \mathbb{R}^n$ is used to extract features from the training set $X_i = \{x_1, x_2, \cdots, x_{N_i}\}$ of class y_i, and the feature mean μ_i is calculated as the class center with Eq.1. Assuming that the total number of boundary features to be saved is M, and the total number of learned classes is C, we select m features with the maximum Euclidean distance from the class center as the boundary features of class y_i, where $m = M/C$. When the model needs to save the boundary features of a new class, the number of saved boundary features for each class is updated to $m' = M/C'$ based on the current total number C' of learned classes, and the closest $m - m'$ boundary features to the class center are removed.

After the updating of boundary features, the current feature extractor parameters are frozen, while the classifier parameters remain trainable. The saved boundary features are used to fine-tune the classifier parameters. We use a fully connected layer as the classifier. Boundary features of each class are directly input into the classifier to calculate the output logits and the cross-entropy loss. After multiple iterations, a modified classifier is obtained.

D. Selection and Management of Exemplars

In incremental learning, saving a small number of samples for each class as exemplars is a simple and effective way to overcome the catastrophic forgetting problem. To implement this, we adopt the exemplar selection and management method proposed by iCaRL [16], which has proven successful in many other algorithms.

The selection of exemplars is mainly divided into two steps. The first step is to calculate the number of samples to be saved for each class based on the current number of learned classes. Assuming that the total number of exemplar samples is fixed at K and the number of current classes is C, the number of samples saved for each class is $m = K/C$. The second step is to calculate the feature mean for each class and select samples based on the Euclidean distance between the extracted features and the feature mean. The training set for class y_i is $X_i = \{x_1, x_2, \cdots, x_{N_i}\}$, and the feature extractor is $f_\theta : \chi \to \mathbb{R}^n$. The feature mean μ_i for class y_i is calculated with Eq.1.

Exemplar samples of class y_i are denoted as $E_i = \{e_1, e_2, \cdots, e_m\}$. The operation to select m samples is as follows:

$$e_q = \arg\min \left\| \mu_i - \frac{1}{q}\left[f_\theta(x) + \sum_{p=1}^{q-1} e_p \right] \right\| \tag{10}$$

where e_q is the qth selected sample, and x represents samples in X_i. The above process iteratively selects samples that make the feature mean of the current exemplar closest to that of the corresponding class. Samples selected earlier have features that are closer to the class feature mean, which makes them more representative of the class. Therefore, the importance of different samples is reflected from the selection order. When the exemplars of new classes need to be established, the number of samples for each class is updated to $m' = K/C'$, where C' is the current total number of classes. All exemplars of old classes remove the last selected $m - m'$ samples to maintain a fixed number of exemplar samples.

E. Algorithm Implementation

The incremental learning algorithm proposed in this paper jointly addresses the catastrophic forgetting of the model from three aspects. It enhances feature separability by minimizing the intra-class scatter and maximizing the inter-class scatter. The intra-class clustering maintains the separability of old class features during model updating. The classifier bias correction based on boundary features mitigates the model's classification bias. Algorithm 1 describes the overall implementation process of the proposed algorithm.

It is worth noting that the proposed algorithm randomly selects samples from each new class to construct a balanced dataset along with the exemplars in each iteration. The sampling process ensures that each class contributes relatively equally when computing the feature separability loss. It also effectively avoids situations where certain batches contain only samples from the new classes, which would prevent the calculation of the intra-class clustering loss. The parameters α and β in the total loss function are scale factors to control the strength of feature separability loss and intra-class clustering loss, respectively.

Algorithm 1: Implementation Process of SSF-IL.

Input: $\chi = (X_{s+1}, \cdots, X_{s+t})$, $\varepsilon = (E_1, \cdots, E_s)$ // new class data, exemplar sets
Input: Θ^s, $\mathrm{M} = (\mu_1, \cdots, \mu_s)$ // current model, old class centers
Input: $\Omega = (\omega_1, \cdots, \omega_s)$ // boundary features of old classes
Output: Θ^{s+t} // model trained on t new classes
for (x, y) ***in*** χ **do**
  for *epoch* ***in*** *epochs:* **do**
    $\varphi \leftarrow X_{s+i} + \varepsilon$ // randomly select a subset from X_{s+i} to form a balanced train set with ε;
    $f, o = \Theta^{s+t}(\varphi)$ // features $f = (f_{old}, f_{new})$, predictions o;
    compute cross-entropy loss with (o, y) as the classification loss L_C;
    compute separability loss $L_{separability}$ with f;
    compute intra-class clustering loss $L_{cluster}$ with (f_{old}, M);
    $L_{total} = L_C + \alpha L_{separability} + \beta L_{cluster}$
  end
  select exemplars from X_{s+i} and update ε;
  select boundary features of X_{s+i} and update Ω;
  fine-tune the classifier with the updated Ω
end

IV. Experiment

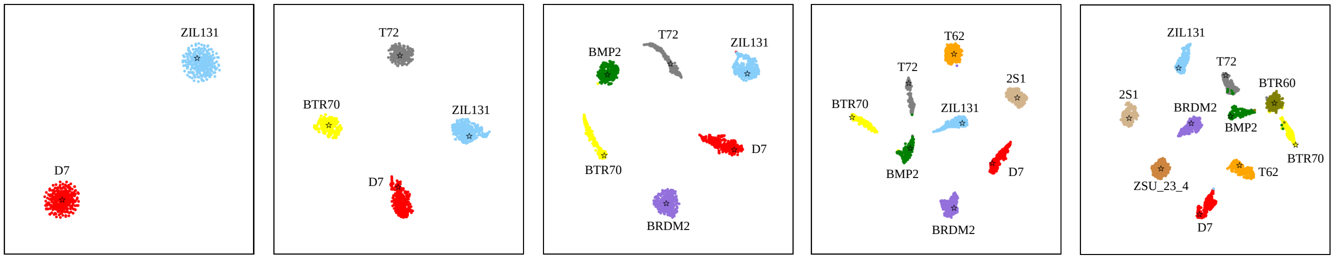
To demonstrate the effectiveness of the proposed algorithm, we carried out class incremental learning experiments on the MSTAR dataset. The dataset and the experimental settings are described in Section IV-A and IV-B. The analysis of experimental results includes the comparison with existing algorithms, ablation experiments, replay data analysis and time consumption.

A. Data Set

The MSTAR dataset is derived from the Motion and Stationary Target Acquisition and Recognition Program, jointly sponsored by the Defense Advanced Research Projects Agency and the Air Force Research Laboratory. The dataset was collected by high-resolution synthetic aperture radar with X-band frequency and HH polarization [35]. It contains ten classes of ground mobile targets, with each class containing SAR images of azimuth angles from 0 degrees to 360 degrees at two different elevation angles of 15 degrees and 17 degrees. In practice, the images at 17-degree elevation angle are usually used as the training set and the images at 15-degree elevation angle are used as the testing set. The number of samples in each class is summarized in Table I.

TABLE I
SAMPLE STATISTICS OF THE MSTAR DATASET

Target Type	Training Set(17°)	Testing Set(15°)
ZSU234	299	274
2S1	299	274
BRDM2	298	274
D7	299	274
BMP2	233	195
ZIL131	299	274
BTR60	256	195
BTR70	233	196
T62	299	273
T72	299	274
Total	**2747**	**2425**



(a) phase 1 (b) phase 2 (c) phase 3 (d) phase 4 (e) phase 5

Fig. 6. Visualization of SSF-IL's feature representation during incremental learning.

B. Experimental Setup

ResNet-18 [36] is utilized as the backbone for the experiments, and the algorithm is implemented in the Pytorch framework [37]. The model is trained using the Adam optimizer for 80 epochs, with an initial learning rate of 0.0005 which is halved every 25 epochs. During the classifier correction phase, the model is iterated 20 times with a learning rate of 0.00025. The feature extractor extracts high-dimensional features of 64 dimensions. Both the total number of exemplar samples and the total number of boundary features are set to 200. In the intra-class clustering loss function, γ is set to 5. In the total loss function, α and β are set to 0.8 and 0.75.

The size of the images in the MSTAR dataset is uniformly set to 128*128, and the batch size of training is set to 128. The images are standardized and then input into the neural network model. The model is trained in three incremental learning scenarios: 10-phase, 5-phase, and 2-phase, with the number of new classes learned by the model in each phase set to 1, 2, and 5, respectively. After training, the model is tested on all previously learned classes, and the testing results of all models are displayed through classification accuracy curves and tables.

C. Comparison Methods

We compare the proposed SSF-IL with some existing incremental learning algorithms to demonstrate the classification performance. The comparative algorithms are as follows:

- JointTraining trains the model on all samples of the old and new classes in each learning phase, and can therefore be considered as the upper limit of performance of all incremental learning algorithms.
- LUCIR [13] applies cosine normalization to the features and classifier weights, and proposes less forget loss and rank margin loss to help distinguish different classes.
- iCaRL [16] is a classical replay-based incremental learning algorithm, which utilizes distillation loss and exemplars to preserve the learned knowledge.
- EEIL [29] samples data from each new class to create a balanced dataset with exemplars. The model is modified at a small learning rate to correct its bias towards new classes.
- PODNet [21] proposes to perform pooling operations on intermediate feature maps in different dimensions and calculate distillation loss using the pooled feature maps of the old and new models.
- DER [38] proposes a new feature representation called ”super feature” that allows freezing the dimensions related to old classes and increasing new dimensions to adapt to new classes.
- FOSTER [39] divides incremental learning process into two stages: boosting and compression. In the boosting stage, new feature extractors are expanded and merged with the old ones to preserve old class knowledge while learning new knowledge. In the compression stage, the expanded model is compressed with knowledge distillation to limit the model size.
- AFC [40] proposes a novel knowledge distillation method that uses the changes of old class feature maps at each layer to calculate the loss function.

D. Experimental Results

Following the settings in Section IV-B, we conducted incremental learning experiments on the MSTAR dataset. According to the different numbers of new classes learned by the model in each phase, the experimental scenarios are divided into 10-phase, 5-phase, and 2-phase incremental learning. In the 5-phase incremental learning scenario, the model initially learns two classes and then proceeds to learn two new classes in each subsequent phase. The experimental process for the other two scenarios is similar to that of the 5-phase incremental learning.

Fig. 6 shows the t-SNE [41] visualization of the feature representation as the SSF-IL sequentially learns new classes. The feature distribution at each phase under the 5-phase scenario is respectively demonstrated in (a)-(e), with different colored clusters representing the features of different classes, and the class centers are marked with stars. The visualized feature distribution demonstrates that, under the constraint of feature separability loss, the features of different classes are clustered around their respective class centers while the class centers are far apart from each other, leading to a large spatial gap between different feature clusters and strong separability. When the model updates parameters to learn new classes, the intra-class clustering loss ensures that the features of old classes are still distributed around the corresponding class centers, despite some deviation. Since the features of new classes are far from old classes due to the feature separability loss, old class features clustered around the fixed class centers can maintain good separability from other classes. Furthermore, the classifier correction at each phase provides more reasonable initial parameters for the next incremental learning phase, which helps the model achieve better performance. In summary, the proposed algorithm avoids significant feature confusion between different classes during model updating, and achieves slow forgetting of previously learned knowledge in the incremental learning process.

1) Recognition Accuracy: As shown in Fig. 7, subfigures (a), (b), and (c) illustrate the target recognition accuracy of the proposed SSF-IL and several comparative algorithms in the 10-phase, 5-phase, and 2-phase incremental learning scenarios, respectively. JointTraining uses all the training data of all classes in each phase, representing the performance upper bound for incremental recognition. Other algorithms experience a gradual decrease in recognition accuracy as they cannot use the complete training set of old classes and forget the learned knowledge.

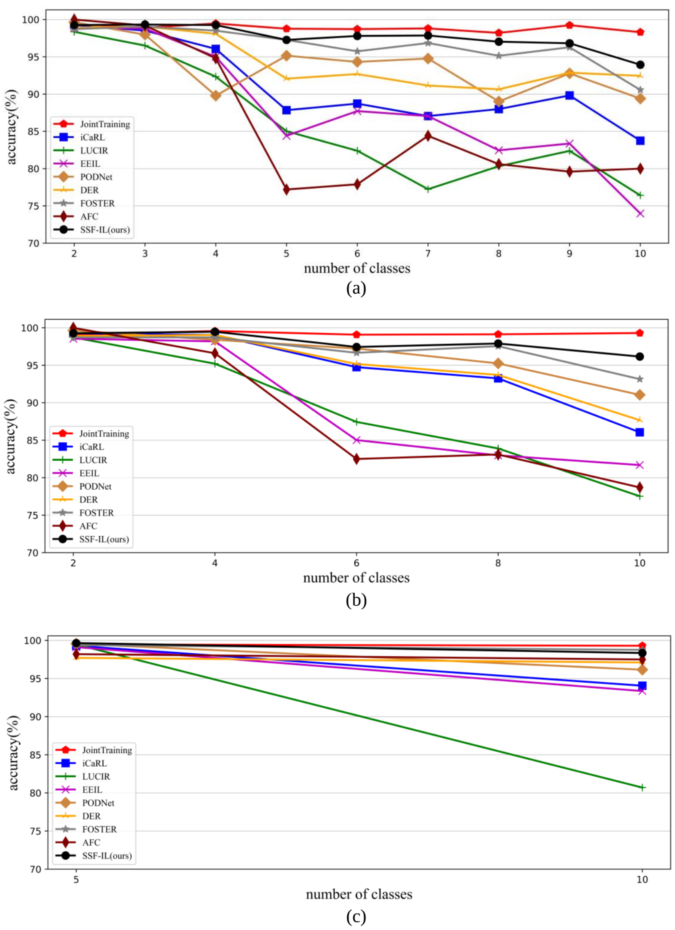


Fig. 7. Comparison of classification accuracy among different incremental learning algorithms. (a) 10-phase. (b) 5-phase. (c) 2-phase.

In all three scenarios, all algorithms achieve almost the same recognition accuracy in the base task. In the 2-phase scenario, FOSTER exhibits the closest recognition accuracy to JointTraining, while SSF-IL is only slightly lower than FOSTER and outperforms other algorithms excluding FOSTER. In the 5-phase and 10-phase scenarios, SSF-IL has the slowest decline in recognition accuracy and achieves the best recognition results. Among the comparative algorithms, iCaRL, LUCIR, and EEIL do not impose additional constraints on the features, resulting in a smaller feature space gap between similar classes, which makes it difficult to distinguish them. PODNet uses pooling output distillation loss, which better preserves the learned knowledge through reinforced constraints, but also decreases the model's adaptability to new classes. FOSTER compresses the expanded model using the knowledge distillation. When there are fewer incremental learning phases, the expanded model effectively adapts to new knowledge, and the loss of old knowledge through distillation is not significant. However, with more incremental learning phases, the gradual accumulation of forgotten old knowledge due to model compression eventually leads to a decline in recognition performance. These methods have not solved the stability-plasticity dilemma of the model in scenarios with more incremental learning phases. SSF-IL uses the feature separability loss to enhance the linear separability of features of all classes, reducing the confusion between old and new classes. Then, an intra-class clustering loss is proposed to replace the knowledge distillation in order to preserve the learned knowledge and limit the offset of old class features within an acceptable range, which better balances the model's stability and plasticity. Finally, a classifier bias correction method based on boundary features is designed to reinforce the decision boundary of the classifier. The combination of the three mitigates the forgetting of the learned knowledge in scenarios with more incremental learning phases.

After learning 10 classes, the target recognition accuracy of SSF-IL reaches 98.35% in the 2-phase scenario, 96.16% in the 5-phase scenario, and 93.94% in the 10-phase scenario, with a smaller decline compared with the base task. These results indicate that SSF-IL effectively avoids the catastrophic forgetting of the model and achieves good performance in SAR target incremental recognition tasks.

TABLE II
MEAN AND STANDARD DEVIATION OF CLASSIFICATION ACCURACY FOR THREE SCENARIOS.

Method	Average Accuracy ± Standard Deviations(%)		
	10-phase	5-phase	2-phase
JointTraining	98.86±0.40	99.23±0.19	99.38±0.09
iCaRL [16]	91.00±5.21	94.47±4.82	96.68±2.62
LUCIR [13]	85.66±7.68	88.56±7.64	90.09±9.38
EEIL [29]	87.96±7.80	89.28±7.49	96.24±2.88
PODNet [21]	93.65±3.56	96.31±3.00	97.86±1.70
DER [38]	94.21±3.24	94.90±4.17	97.41±0.30
FOSTER [39]	96.46±2.45	96.96±2.06	**99.07±0.31**
AFC [40]	85.97±8.82	88.18±8.47	97.85±0.35
SSF-IL(ours)	**97.62±1.60**	**98.05±1.22**	99.00±0.65

Table II presents the mean and standard deviation of recognition accuracy for all models in the three incremental learning scenarios. In the 2-phase scenario, FOSTER achieves the best recognition performance. The mean and standard deviation of recognition accuracy of SSF-IL are very close to FOSTER's and superior to other comparative algorithms outside of FOSTER. In the 5-phase and 10-phase scenarios, SSF-IL has a higher average recognition accuracy and a smaller

TABLE III
MODELS USED FOR THE ABLATION EXPERIMENTS

Model	Cross-Entropy Loss	Knowledge Distillation	Feature Separability Loss	Intra-class Clustering Loss	Classifier Bias Correction
Hybrid1	✓	✓			
Hybrid2	✓			✓	
Hybrid3	✓	✓	✓		
Hybrid4	✓		✓	✓	
SSF-IL	✓		✓	✓	✓

standard deviation, which indicates less significant fluctuations in accuracy. The significant fluctuations in accuracy mean that the newly learned knowledge causes a great impact on the knowledge of old classes, which reflects the lack of model stability in the incremental recognition task. The comparative models all have larger fluctuations in recognition accuracy than SSF-IL in adjacent phases, while SSF-IL effectively avoids mutual interference between the features of old and new classes by reinforcing and maintaining the separability of features during incremental learning. Thus SSF-IL achieves better model stability.

Furthermore, as shown in Table II, it can be seen that with an increase in incremental learning phases, there is a decrease in the average recognition accuracy and an increase in the accuracy standard deviation, indicating more forgetting of the learned knowledge. For the same algorithm and dataset, the model performs best in the 2-phase incremental learning scenario, followed by the 5-phase scenario, with the recognition accuracy dropping fastest in the 10-phase scenario. Compared with the 2-phase incremental learning scenario, comparative algorithms, such as iCaRL, EEIL, PODNet, and DER, exhibit a significant recognition performance decrease in the 5-phase and 10-phase scenarios, with the accuracy dropping over 7%. In contrast, the proposed SSF-IL has less degradation in target recognition performance when switching scenarios. After learning 10 classes, the recognition accuracy difference between the 2-phase and 5-phase scenarios is approximately 2%, and the difference between the 5-phase and 10-phase scenarios is also 2%. SSF-IL exhibits a more stable target recognition performance compared with other algorithms and can effectively handle various incremental learning scenarios.

2) Ablation Experiments: The algorithm proposed in this paper combines feature separability loss, intra-class clustering loss, and classifier bias correction to mitigate the neural network model's forgetting of previously learned knowledge. To analyze the contribution of the proposed methods to incremental recognition tasks, five models are compared in this section. The differences between these models are shown in Table III.

The ablation experiment is conducted in the 5-phase incremental learning scenario, and Fig. 8 displays the recognition accuracy of the comparative models. All comparative models use cross-entropy loss as the classification loss. Hybrid1 employs knowledge distillation without imposing any additional constraints on features. It suffers significant forgetting of previously learned knowledge during model updates, resulting in the lowest recognition accuracy. Hybrid2, on the other hand, uses intra-class clustering loss instead of knowledge distillation to preserve learned knowledge. It achieves better recognition performance compared to Hybrid1, demonstrating that the proposed intra-class clustering loss can replace traditional knowledge distillation and play a better role in SAR target incremental recognition tasks. To illustrate the impact of intra-class clustering loss on model stability and plasticity, the misclassification counts of new and old classes for Hybrid1 and Hybrid2 are presented in Table IV. Hybrid1 exhibits fewer misclassifications for new classes, but a considerable number of classification errors on old classes. In comparison, Hybrid2 shows a slight increase in misidentifying new classes but significantly reduces errors in recognizing old classes, resulting in an overall reduction in misclassifications. Obviously, the intra-class clustering loss better balances model stability and plasticity compared to knowledge distillation. In addition, it should be noted that Hybrid2 does not reinforce feature separability, therefore the γ in Eq.8 needs to be sufficiently large to ensure strong constraints on the features of old classes.

Hybrid3 adds feature separability loss based on Hybrid1. After learning all 10 target classes, its recognition accuracy surpasses Hybrid1 by more than 3%. Thus, the introduction of feature separability loss effectively improves the model's incremental recognition performance. Hybrid4 simultaneously utilizes feature separability loss and intra-class clustering loss, further increasing recognition accuracy compared to Hybrid2 and Hybrid3. This indicates that the proposed feature separability loss and intra-class clustering loss can collaborate effectively in improving incremental recognition performance. SSF-IL, built upon Hybrid4 by incorporating classifier bias correction, ultimately achieves the highest recognition accuracy. This result validates the positive contribution of the proposed classifier bias correction method based on boundary features to incremental recognition.

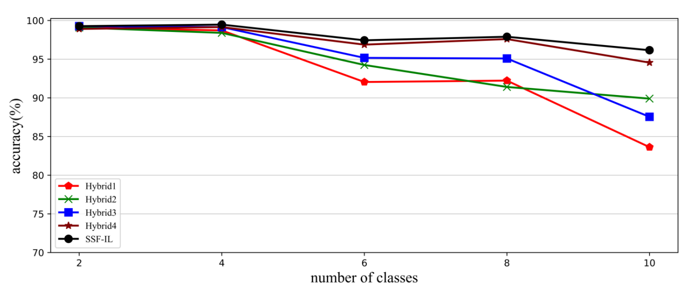


Fig. 8. Comparison of classification accuracy in ablation experiments.

To further demonstrate the constraining effects of the proposed feature separability loss and intra-class clustering loss on the features, changes in feature distribution during the incremental learning for Hybrid1, Hybrid3, and Hybrid4 are shown in Fig. 9. Subfigures (a)-(e) correspond to Hybrid1, (f)-

TABLE IV
COMPARISON OF MODEL STABILITY AND PLASTICITY UNDER SEPARATE CONSTRAINTS OF INTRA-CLASS CLUSTERING LOSS AND KNOWLEDGE DISTILLATION.

Model	Number of misclassifications in new classes					Number of misclassifications in old classes				
	phase 1	phase 2	phase 3	phase 4	phase 5	phase 1	phase 2	phase 3	phase 4	phase 5
Hybrid1	5	1	1	0	11	-	11	111	143	386
Hybrid2	5	3	12	22	49	-	12	69	146	196

(j) to Hybrid3, and (k)-(o) to Hybrid4. The features extracted by Hybrid1 are only highly separable in phase 1 and 2. Starting from phase 3, the features extracted by Hybrid1 become increasingly confused, and this confusion continues to worsen in subsequent phases, indicating that Hybrid1 forgets a lot of the learned knowledge during model updating. After learning all classes, there is a serious overlap in the features of BTR70, BTR60, and BMP2, making it difficult for the classifier to distinguish these classes. Additionally, there is some confusion between the features of T62 and BRDM2. As a result, the recognition accuracy of Hybrid1 in phase 5 decreases significantly compared with phase 1.

Hybrid3 incorporates the feature separability loss on the basis of Hybrid1. As can be seen from the figures, the features of the same class extracted by Hybrid1 have relatively loose distribution and the feature clusters of different classes are close in space. Hybrid3 reinforces the linear separability of features, resulting in closer distribution of same-class features around the feature center. Moreover, the feature clusters of different classes are more evenly distributed in space, and the distance between class centers is larger, which makes it easier for the classifier to differentiate. Compared with Hybrid1, Hybrid3 significantly reduces the confusion between features of different class in phase 3, 4, and 5, leading to effective improvement of classification accuracy.

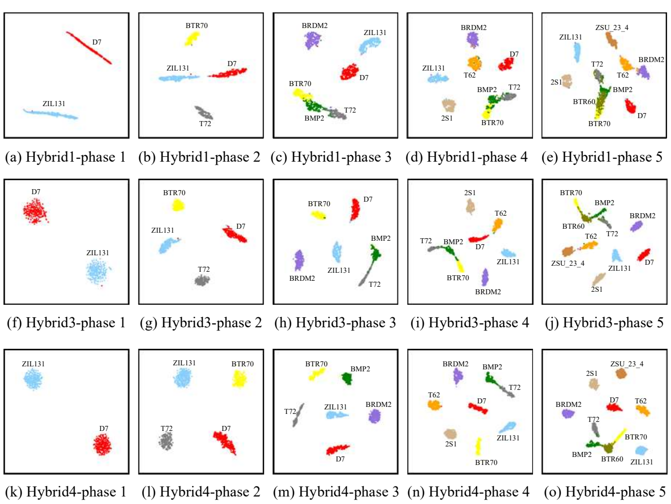


Fig. 9. The visualized feature distribution of Hybrid1, Hybrid3 and Hybrid4.

Hybrid4 replaces the knowledge distillation with the intra-class clustering loss on the basis of Hybrid3. Compared with Hybrid3, Hybrid4 further mitigates the feature confusion among old classes and between new and old classes. After learning new classes, all features still distribute uniformly in space, with only a few features from different classes having slight overlap at the boundary. Thus, the linear separability of features is well maintained during model updating, resulting in less forgetting of learned knowledge. As a result, the incremental recognition accuracy of Hybrid4 is significantly better than that of Hybrid3 and Hybrid1. In conclusion, the feature separability loss can effectively optimize the spatial distribution of features and enhance separability, while the intra-class clustering loss can maintain the linear separability of features and avoid confusion among different classes. The combination of the two yields high classification performance in incremental recognition tasks.

Fig. 10 shows the confusion matrices of the recognition results of Hybrid4 and SSF-IL in each incremental learning phase, where (a)-(e) correspond to Hybrid4 and (f)-(j) correspond to SSF-IL. It can be seen that Hybrid4 exhibits a stronger classification bias towards new classes compared with SSF-IL in each incremental learning phase, with more old class samples being misclassified as new classes. SSF-IL builds on Hybrid4 by incorporating a classifier bias correction method based on boundary features. By fine-tuning the classifier parameters, the proposed bias correction method strengthens the decision boundaries between the old and new classes, enabling the classifier to better differentiate samples that are prone to confusion with other classes. In the recognition results of SSF-IL, there is a significant reduction in the number of old class samples mistakenly classified as new classes, while the classification performance of new classes remains relatively unchanged. This overall improvement in classification performance validates the effectiveness of the classifier correction method based on boundary features.

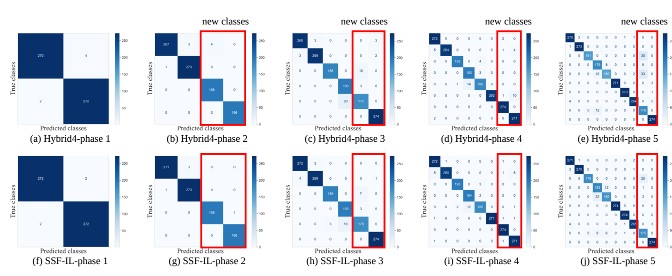


Fig. 10. Comparison between confusion matrices with and without classifier correction.

3) Replay Data Analysis: Fig. 11 illustrates the impact of the total number of exemplar samples K and the number of preserved boundary features M on incremental recognition performance. Experimental results indicate that the more exemplar samples preserved, the better the performance of incremental recognition. However, in comparison to preserving

200 exemplar samples, saving 400 exemplar samples leads to a slight improvement in incremental recognition accuracy while significantly increasing storage resource consumption. Moreover, when K is set to 150 or 100, there is a significant decrease in recognition accuracy. Thus, setting the total number of exemplar samples to 200 can strike a balance between higher recognition accuracy and lower storage consumption. Similar to the exemplar, the more boundary features preserved, the higher the recognition accuracy. Considering the contribution to recognition performance and computational cost, we set the total number of preserved boundary features to 200.

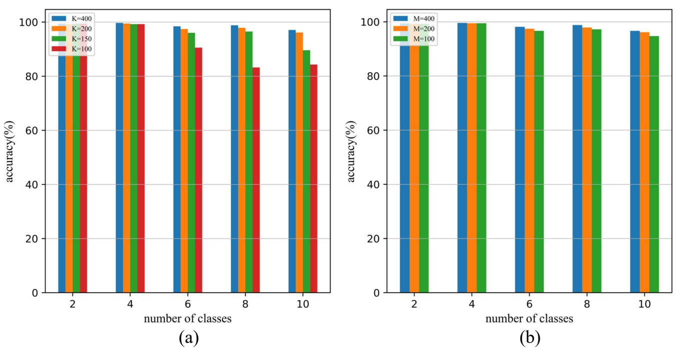


Fig. 11. The impact of the number of replay data on recognition performance. (a) Exemplar. (b) Boundary features.

4) Time Consumption: As mentioned earlier, one of the requirements for incremental learning algorithms is that the computational resources consumed by the algorithm do not increase linearly with the increasing amount of classes. Under identical training conditions, the computational costs of different algorithms can be reflected by the training time. Fig. 12 illustrates the training time of all algorithms in three incremental learning scenarios. As shown in Fig. 12, iCaRL, due to its simple structure, has the shortest training time. The training time of SSF-IL is close to that of iCaRL and significantly lower than other models such as PODNet, FOSTER, and DER. Additionally, when SSF-IL learns new classes, it randomly selects samples from different new classes to form a balanced dataset with exemplars for each iteration. Therefore, the training time of SSF-IL when learning new classes is reduced compared to the basic training that uses all samples in each iteration, and remains almost unchanged in the subsequent phases. It can be concluded that the SSF-IL proposed in this paper can effectively control the computational resources consumed during model updating and limit them to a relatively reasonable range.

V. Conclusion

In this paper, we present a novel SAR target incremental recognition approach based on strong separability features. The proposed method enhances the linear separability of features by minimizing the intra-class scatter and maximizing the inter-class scatter, avoiding confusion between different classes. The intra-class clustering loss better balances the stability and plasticity of the model and maintains the strong separability of old class features during model updating to

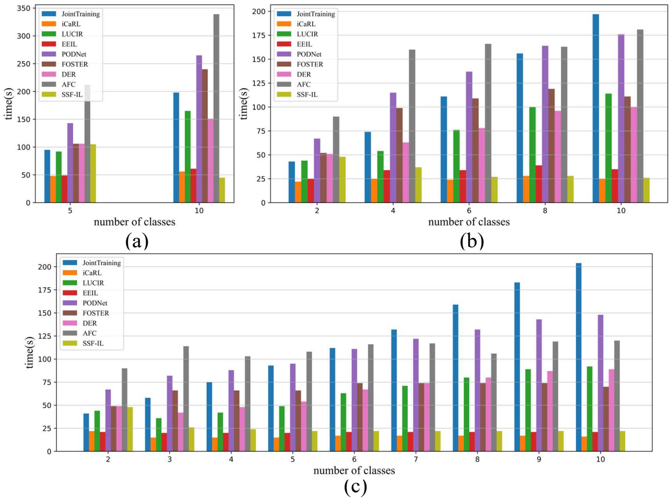


Fig. 12. Comparison of time consumption. (a) 2-phase. (b) 5-phase. (c) 10-phase.

preserve the learned knowledge. Finally, a classifier bias correction method based on boundary features is designed to modify the classifier's decision boundary and reduce the classifier's bias towards new classes.

The effectiveness of the proposed algorithm is verified through target recognition experiments on the MSTAR dataset. SSF-IL not only achieves higher recognition accuracy on the testing set but also effectively controls the computational cost of model updating, meeting practical needs for incremental recognition tasks.

References

[1] Y. Zhou, H. Liu, F. Ma, Z. Pan, and F. Zhang, "A sidelobe-aware small ship detection network for synthetic aperture radar imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–16, 2023.

[2] Z. Geng, H. Yan, J. Zhang, and D. Zhu, "Deep-learning for radar: A survey," *IEEE Access*, vol. 9, pp. 141 800–141 818, 2021.

[3] F. Ma, X. Sun, F. Zhang, Y. Zhou, and H.-C. Li, "What catch your attention in sar images: Saliency detection based on soft-superpixel lacunarity cue," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–17, 2023.

[4] Z. Yue, F. Gao, Q. Xiong, J. Wang, T. Huang, E. Yang, and H. Zhou, "A novel semi-supervised convolutional neural network method for synthetic aperture radar image recognition," *Cognitive Computation*, vol. 13, no. 4, pp. 795–806, 2019.

[5] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A survey of convolutional neural networks: Analysis, applications, and prospects," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 12, pp. 6999–7019, 2022.

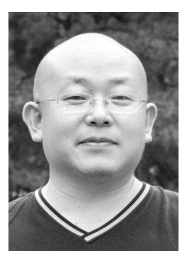
[6] Q. Yang, Y. Gu, and D. Wu, "Survey of incremental learning," in *2019 chinese control and decision conference (ccdc)*. IEEE, 2019, pp. 399–404.

[7] Z. Mai, R. Li, J. Jeong, D. Quispe, H. Kim, and S. Sanner, "Online continual learning in image classification: An empirical survey," *Neurocomputing*, vol. 469, pp. 28–51, 2022.

[8] Z. Zheng, X. Nie, and B. Zhang, "Fine-grained continual learning for sar target recognition," in *IGARSS 2022-2022 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2022, pp. 2207–2210.

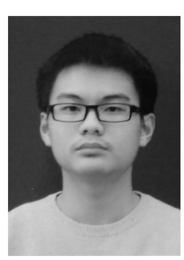
[9] Y. Zhou, S. Zhang, X. Sun, F. Ma, and F. Zhang, "Sar target incremental recognition based on hybrid loss function and class-bias correction," *Applied Sciences*, vol. 12, no. 3, p. 1279, 2022.

[10] B. Li, Z. Cui, Z. Cao, and J. Yang, "Incremental learning based on anchored class centers for sar automatic target recognition," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–13, 2022.
[11] M. De Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars, "A continual learning survey: Defying forgetting in classification tasks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 7, pp. 3366–3385, 2021.
[12] M. Masana, X. Liu, B. Twardowski, M. Menta, A. D. Bagdanov, and J. van de Weijer, "Class-incremental learning: survey and performance evaluation on image classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.
[13] S. Hou, X. Pan, C. C. Loy, Z. Wang, and D. Lin, "Learning a unified classifier incrementally via rebalancing," in *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, 2019, pp. 831–839.
[14] E. Belouadah and A. Popescu, "Il2m: Class incremental learning with dual memory," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 583–592.
[15] J. Tang, D. Xiang, F. Zhang, F. Ma, Y. Zhou, and H. Li, "Incremental sar automatic target recognition with error correction and high plasticity," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 15, pp. 1327–1339, 2022.
[16] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "icarl: Incremental classifier and representation learning," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017, pp. 2001–2010.
[17] S. Mittal, S. Galesso, and T. Brox, "Essentials for class incremental learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3513–3522.
[18] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 12, pp. 2935–2947, 2017.
[19] A. Ashok, K. Joseph, and V. N. Balasubramanian, "Class-incremental learning with cross-space clustering and controlled transfer," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVII*. Springer, 2022, pp. 105–122.
[20] H. Jung, J. Ju, M. Jung, and J. Kim, "Less-forgetting learning in deep neural networks," *arXiv preprint arXiv:1607.00122*, 2016.
[21] A. Douillard, M. Cord, C. Ollion, T. Robert, and E. Valle, "Podnet: Pooled outputs distillation for small-tasks incremental learning," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16*. Springer, 2020, pp. 86–102.
[22] K. Lee, K. Lee, J. Shin, and H. Lee, "Overcoming catastrophic forgetting with unlabeled data in the wild," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 312–321.
[23] F. Zhu, X.-Y. Zhang, C. Wang, F. Yin, and C.-L. Liu, "Prototype augmentation and self-supervision for incremental learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 5871–5880.
[24] Y. Liu, B. Schiele, and Q. Sun, "Adaptive aggregation networks for class-incremental learning," in *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, 2021, pp. 2544–2553.
[25] Q. Lao, X. Jiang, M. Havaei, and Y. Bengio, "A two-stream continual learning system with variational domain-agnostic feature replay," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 9, pp. 4466–4478, 2021.
[26] S. Dang, Z. Cao, Z. Cui, Y. Pi, and N. Liu, "Class boundary exemplar selection based incremental learning for automatic target recognition," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 8, pp. 5782–5792, 2020.
[27] H. Shin, J. K. Lee, J. Kim, and J. Kim, "Continual learning with deep generative replay," *Advances in neural information processing systems*, vol. 30, 2017.
[28] Y. Wu, Y. Chen, L. Wang, Y. Ye, Z. Liu, Y. Guo, and Y. Fu, "Large scale incremental learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 374–382.
[29] F. M. Castro, M. J. Marín-Jiménez, N. Guil, C. Schmid, and K. Alahari, "End-to-end incremental learning," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 233–248.
[30] B. Zhao, X. Xiao, G. Gan, B. Zhang, and S.-T. Xia, "Maintaining discrimination and fairness in class incremental learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 13 208–13 217.
[31] H. Huang, F. Gao, J. Wang, A. Hussain, and H. Zhou, "An incremental sar target recognition framework via memory-augmented weight alignment and enhancement discrimination," *IEEE Geoscience and Remote Sensing Letters*, vol. 20, pp. 1–5, 2023.
[32] D. Elizondo, "The linear separability problem: Some testing methods," *IEEE Transactions on neural networks*, vol. 17, no. 2, pp. 330–344, 2006.
[33] D. A. Elizondo, R. Birkenhead, M. Gamez, N. Garcia, and E. Alfaro, "Linear separability and classification complexity," *Expert Systems with Applications*, vol. 39, no. 9, pp. 7796–7807, 2012.
[34] F. Gao, M. Wang, J. Wang, E. Yang, and H. Zhou, "A novel separability objective function in cnn for feature extraction of sar images," *Chinese Journal of Electronics*, vol. 28, no. 2, pp. 423–429, 2019.
[35] E. R. Keydel, S. W. Lee, and J. T. Moore, "Mstar extended operating conditions: A tutorial," *Algorithms for Synthetic Aperture Radar Imagery III*, vol. 2757, pp. 228–242, 1996.
[36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
[37] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.
[38] S. Yan, J. Xie, and X. He, "Der: Dynamically expandable representation for class incremental learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3014–3023.
[39] F.-Y. Wang, D.-W. Zhou, H.-J. Ye, and D.-C. Zhan, "Foster: Feature boosting and compression for class-incremental learning," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXV*. Springer, 2022, pp. 398–414.
[40] M. Kang, J. Park, and B. Han, "Class-incremental learning by knowledge distillation with adaptive feature consolidation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 16 071–16 080.
[41] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.



Fei Gao received the B.S. degree in industrial electrical automation, and the M.S. degree in electromagnetic measurement technology and instrument from the Xi'an Petroleum Institute, Xi'an, China, in 1996 and 1999, respectively, and the Ph.D. degree in signal and information processing from the 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, deep learning for applications in remote sensing.

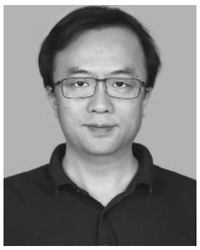


Lingzhe Kong received the B.S. degree in electronic and information engineering from Beihang University, Beijing, China, in 2021. He is currently pursuing the Ph.D. degree in information and communication engineering at Beihang University.

His current research activities are in target recognition and synthetic aperture radar image processing.

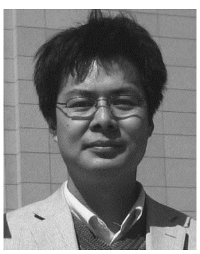
Rongling Lang was born in 1975. She received her Ph.D. degree in automatic control from Northwestern Polytechnical University in 2005. She is with School of Electronic and Information Engineering in Beihang University.

Her current research interests include satellite navigation and fault diagnosis of avionics system.



Jinping Sun (Member, IEEE) received the M.Sc. and Ph.D. degrees from Beihang University (BUAA), Beijing, China, in 1998 and 2001, respectively.

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



Jun Wang received the B.S. degree in communication engineering from the North western Polytechnical University, Xi'an, China, in 1995 and the M.S.and Ph.D. degrees in signal and information processing from the Beijing University of Aeronautics and Astronautics (BUAA), Beijing, China, in 1998 and2001, respectively.

He is currently a Professor with the School of Electronic and Information Engineering, BUAA. He is interested in signal processing, DSP/FPGA real time architecture, target recognition and tracking, and so on. His research has resulted in over 40 papers in journals, books, and conference proceedings.

Amir Hussain received the B.Eng. degree and the Ph.D. degree in Electronic & Electrical Engineering from University of Strathclyde, Scotland, UK, in1992 and 1997, respectively.

Following postdoctoral and senior academic positions at West of Scotland (1996-98), Dundee (1998-2000) and Stirling Universities (2000-18) respectively, he joined Edinburgh Napier University as founding Head of the Cognitive Big Data and Cybersecurity (CogBiD) Research Lab and the Centre for AI 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 Huazhong University of Science and Technology, Wuhan, China, the M.S. degree in biomedical engineering from University of Dundee,Dundee, U.K., and the Doctor of Philosophy degree in 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.He has published widely in medical image processing, computer vision, intelligent systems, and data mining.