applied sciences

MDPI

Article

SAR Target Incremental Recognition Based on Hybrid Loss Function and Class-Bias Correction

Yongsheng Zhou [1], Shuo Zhang [1], Xiaokun Sun [1,*], Fei Ma [1] and Fan Zhang [1,2]

[1] College of Information Science and Technology, Beijing University of Chemical Technology, Beijing 100029, China; zhyosh@mail.buct.edu.cn (Y.Z.); 2019210502@buct.edu.cn (S.Z.); mafei@mail.buct.edu.cn (F.M.); zhangf@mail.buct.edu.cn (F.Z.)

[2] Interdisciplinary Research Center for Artificial Intelligence, Beijing University of Chemical Technology, Beijing 100029, China

* Correspondence: sunxk@mail.buct.edu.cn

Abstract: The Synthetic Aperture Radar (SAR) target recognition model usually needs to be retrained with all the samples when there are new-coming samples of new targets. Incremental learning emerges to continuously obtain new knowledge from new data while preserving most previously learned knowledge, saving both time and storage. There are still three problems in the existing incremental learning methods: (1) the recognition performance of old target classes degrades significantly during the incremental process; (2) the target classes are easily confused when similar target classes increase; (3) the model is inclined to new target classes due to class imbalance. Regarding the three problems, firstly, the old sample preservation and knowledge distillation were introduced to preserve both old representative knowledge and knowledge structure. Secondly, a class separation loss function was designed to reduce the intra-class distance and increase the inter-class distance, effectively avoiding the confusion between old and new classes. Thirdly, a bias correction layer and a linear model was designed, which enabled the model to treat the old and new target classes more fairly and eliminate the bias. The experimental results on the MSTAR dataset verified the superior performance compared with the other incremental learning methods.

Keywords: incremental learning; SAR target recognition; knowledge distillation; old sample preservation; class imbalance

check for updates

Citation: Zhou, Y.; Zhang, S.; Sun, X.; Ma, F.; Zhang, F. SAR Target Incremental Recognition Based on Hybrid Loss Function and Class-Bias Correction. *Appl. Sci.* **2022**, *12*, 1279. https://doi.org/10.3390/app12031279

Academic Editor: Qizhi Xu

Received: 25 December 2021
Accepted: 23 January 2022
Published: 25 January 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

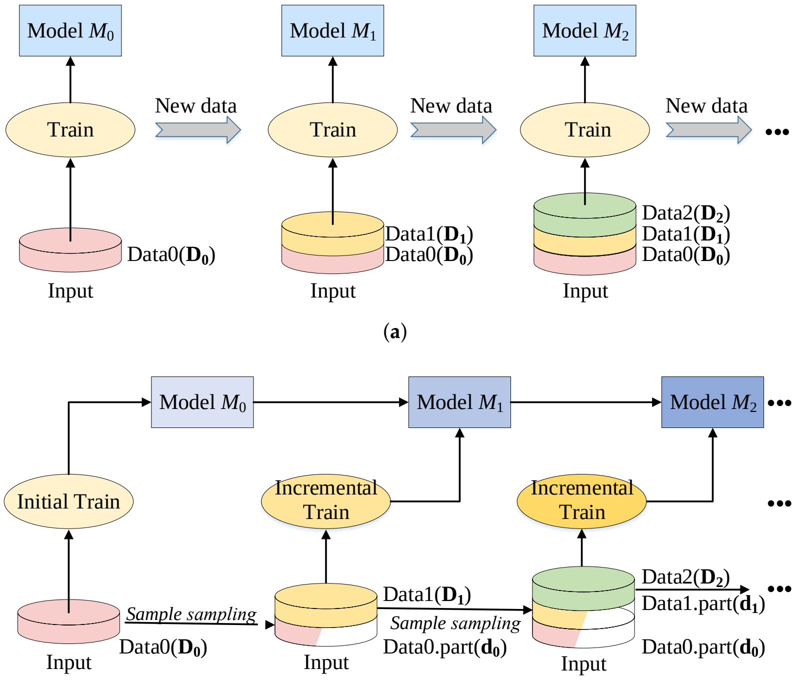
Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Remote sensing is a detection technology that obtains information about objects without direct contact at long distances. It plays a very important role in many studies, and an increasing number of more precise sensors and measurements are also providing researchers with more information [1]. It has a growing impact in a wide variety of areas from business to science to public policy. Synthetic Aperture Radar (SAR) is one of the active remote sensing technologies and has a unique ability to obtain high-resolution microwave images of the Earth's surface targets in almost all weather conditions. Automatic Target Recognition (ATR) based on the SAR images, especially ATR through deep learning, has played an essential role in wide-area monitoring of targets such as vehicles, ships, and aircraft [2–4].

The commonly used training method for the target recognition model is the one-time supervised learning or batch learning mode to process the data in an offline manner [5,6], as shown in Figure 1a. All existing labeled target samples are trained to predict the labels of unknown input data. However, it is usually challenging to collect the labeled samples of all the target classes at once in practice, and the training data of the new target classes are obtained gradually. The storage requirement of training samples increases as well. The trained target recognition model using old target samples needs to be retrained using both old and new target samples. Due to the limitation of computation and storage resources in some particular application situations, this approach cannot be applied.

The incremental learning (also known as lifelong learning, continuous learning) emerges to enable trained target recognition model to continuously learn new tasks [7]. It keeps knowledge gained from previous tasks without the need to preserve large amounts of old data [8], as shown in Figure 1b. Incremental learning focuses the problem of catastrophic forgetting [9–11]. The catastrophic forgetting means that after a target recognition model is trained with a new dataset, the weights necessary for the old tasks are changed to adapt to the new task, and the knowledge of the previously learned task is lost.



(**a**)

(**b**)

Figure 1. Illustration of the non-incremental learning and incremental learning process. (**a**) Non-incremental learning (batch learning) process. (**b**) Incremental learning process.

Although some incremental learning methods have been proposed, there are still three problems needed to be addressed.

(1) The performance of the incremental process still degrades significantly for old target classes, i.e., incremental learning faces the problem of catastrophic forgetting, and there are varying degrees of performance degradation as the classes increase. Regarding the problem of catastrophic forgetting, some methods are proposed, such as preserving a small number of training samples of old classes [8,12], using a sub-network for each incremental phase [13,14], freezing the nodes of the network essential for the old task during the training of the model [15,16]. Among them, the old sample preservation is a straightforward way to prevent the model from forgetting what it has learned. However, it is difficult to ensure that the preserved old samples are representative enough to train a new model with good recognition performance. Although the methods in [17,18] learn to generate pseudo-samples of the old data to reconstruct the data distribution of the old classes, the quality of the pseudo-samples is not able to entirely represent the actual samples. It is worth noting that the knowledge distillation [19], which is initially proposed for transfer learning, is very effective in enabling the current model to simulate the old model. Therefore, it can be introduced into incremental learning. However, the standard knowledge distillation method builds relevant models by minimizing the Kullback–Leibler Divergence (KLD) between probabilistic outputs, and ignores important knowledge structures in the old model [20]. In this paper, considering the advantage of knowledge distillation and old

sample preservation, they are combined to alleviate catastrophic forgetting. Firstly, the samples closer to the class center of the old data are selected and preserved, because these samples are more representative and able to better represent a class of targets. Secondly, the cross-entropy loss function and the knowledge distillation loss function of the old and new samples are integrated. The distillation loss between the old and new classes is used to preserve knowledge, and the cross-entropy loss is used to identify different classes. The advantage of this approach is that not only the old representative knowledge but also the knowledge structure are preserved.

(2) The old and new target classes are easily confused when similar target classes increase. With the arrival of new targets, the total number of classes to be identified by the model increases, and some visually similar classes may appear in each training phase. When the number of old and new target samples is unbalanced, the increase of visually similar classes will further degrade the recognition performance. On the one hand, this is because the boundary between target classes is more susceptible to class imbalance. On the other hand, it is also because the difference of samples within the same classes and the similarity of samples among different classes are more likely to cause the model to confuse the recognition of old and new classes and produce false recognition results. In [21], a new loss function, i.e., rectification loss, is adopted to deal with the confusion among old and new target classes. However, the disadvantage is that it relies more on the model to select the corresponding samples in the training dataset to calculate this loss. In this paper, a class separation loss function is designed that does not rely on directly selecting samples in the training dataset and can be directly added into the incremental learning. It is capable of making the inter-class sample distance larger, and the intra-class sample distance smaller, promoting the separation of old and new classes.

(3) The target recognition model is more likely to classify the input samples as new target classes because there is a weight bias in the fully connected layer of model. The weight bias occurs due to that there is a class imbalance between the old and new classes during training, where the model is able to obtain all samples of the new targets but only a few samples of the old targets. In [12], a balanced training phase is introduced at the end of each model training to fine-tune the model with a balanced dataset. However, these samples cannot fully represent the data distribution and may over-fit the stored samples. In this paper, a bias correction is performed to solve the weight bias problem in the fully connected layer. Firstly, a bias correction layer is added after the fully connected layer of the model, and a linear model with two parameters is used. Secondly, the preserved representative samples are used to train the bias correction layer. This approach enables the model to treat the old and new target classes more fairly and eliminates the bias.

In all, the three main contributions of this paper are summarized as follows.

- The knowledge distillation method has been introduced into SAR target incremental recognition with the combination of old sample preservation. It is capable of better achieving plasticity-stability balance and preventing catastrophic forgetting of old classes.
- An effective loss function for class separation with attention to class boundaries has been designed, which increases inter-class sample distance and decrease intra-class sample distance, therefore facilitates the separation between new and old classes.
- A bias correction layer has been designed and trained to solve the problem of weight bias in the fully connected layer, which corrects the biased output of the model.

The rest of this paper is organized as follows. Section 2 describes the existing methods of incremental learning. Section 3 introduces the proposed method in detail. In Section 4, experiments results on the dynamic/static target acquisition and recognition (MSTAR) dataset are presented. Section 5 draws the conclusions.

2. Relative Work

Incremental learning is a topic that has long existed in machine learning and has attracted more and more attention with the introduction of deep learning [8,22]. Incremental learning aims to train the model to recognize new target classes while preserving its initially learned knowledge of the old target classes. As shown in Table 1, many approaches have been proposed in recent research to find a good compromise between the stability and plasticity of trained models to address the catastrophic forgetting problem faced by neural networks.

In this section, three types of incremental learning methods are briefly reviewed, including the old sample preservation method, the knowledge distillation method, and the method for resolving class imbalance. The old sample preservation strategy stores a subset of representative samples from previous tasks for replay in the incremental training phase [23]. The knowledge distillation approach introduces the concept of a teacher model (complex, highly learned model) and a student model (simple, small parametric model) [24]. By training the student model to transfer knowledge, the representative knowledge learned in the teacher model can be transferred to the student model [25]. The combination of the two methods is able to effectively improve the performance of the target recognition model and alleviate catastrophic forgetting. Although the methods based on preserving representative samples are effective in incremental learning, the class imbalance problem still exists. The new class has more samples in the training phase than the old class. It affects both the convergence of the model in the training phase and the generalization of the model in the test dataset.

Table 1. Summary of existing incremental learning methods.

Methods	**Year**	**Old Data Preservation**	**Knowledge Distillation**	**Class Balancing**	**Application Field (Dataset Name)**
LWF [15]	2016		✓		Places365-standard, ILSVRC 2012.
iCaRL [8]	2017	✓	✓	✓	CIFAR-100, ILSVRC 2012.
EWC [9]	2017				MNIST.
ICL-GAN [26]	2018		✓		CIFAR-100, Flower-102, MS-Celeb-1M-Base.
EEIL [12]	2018	✓	✓	✓	CIFAR-100, ILSVRC 2012.
LWM [27]	2019		✓		iCIFAR-100, iILSVRC-small.
United [28]	2019	✓	✓	✓	CIFAR-100, ImageNet-Subset.
Bic [29]	2019	✓	✓	✓	ImageNet-1000, MS-Celeb1M.
M2KD [30]	2019		✓		CIFAR-100, iILSVRC-small.
IL2M [31]	2019	✓	✓	✓	ILSVRC, VGGFace2, Landmarks.
ScaIL [32]	2020	✓			ILSVRC, VGGFace2, Landmarks, CIFAR-100.
Mnemonics Training [33]	2020	✓	✓		CIFAR-100, ILSVRC 2012.
CBesIL [6]	2020	✓			MSTAR.
Our-method	2021	✓	✓	✓	MSTAR.

2.1. Old Sample Preservation

Multi-class incremental learning without forgetting (mnemonics training) [33] is a method based on old sample preservation, and it is similar to [8,12,28,29]. The method optimizes the framework in both model-level and sample-level to ensure that the reserved samples are capable of well representing the boundary and mean of each class's distribution. However, it relies on well-preprocessed feature extractors to obtain superior performance of incremental learning. The Class Boundary exemplar selection based Incremental Learning (CBesIL) for automatic target recognition [6] mainly consists of two parts, i.e., class boundary selection and incremental learning. The class boundary selection is based on local geometric information and statistical information to extract class boundaries. It performs distribution reconstruction to update the sample set when new classes are added. However, it only considers protecting class boundary updates during data reconstruction, which will cause inaccuracy of recognition. The overcoming of catastrophic forgetting in neural networks (EWC) [9] evaluates the importance of network synapses by computing the Fisher information matrix, which is used to slow down the learning of weights that highly correlate with the previous task. However, this approach also leads to an imbalance for new tasks. The classifier weights Scaling for class Incremental Learning (ScaIL) [32] reuses the network weights learned from all data to reduce bias. However, it depends on comparing classifier weights for the current state and the initial state.

It is also a feasible way to generate pseudo-samples of old data using Generative Adversarial Networks (GAN), reducing storage consumption. Wu et al. [26] use GAN to generate pseudo images for old classes and combine these images with new class's images and achieve slightly better performance than [8]. However, the quality of the pseudo-samples is not able to entirely represent the actual samples. Therefore the performance will significantly degrade if only relying on the generated pseudo-samples. Moreover, it is not the optimal incremental approach due to its requirement of creating more GAN models and more memory cost.

Based on the above analysis, this paper adopts the old sample preservation strategy so that the model does not forget what it has learned. When a new target arrives, the first k samples closer to the class center of each target are preserved.

2.2. Knowledge Distillation

Rather than preserving old samples, the Learning Without Forgetting (LWF) [15] is a pioneering incremental learning effort that uses knowledge distillation to minimize the representation difference between old and new classes. During the incremental learning process of this method, the network parameters related to the old classes are frozen, only the new parameters are trained, and finally, all network parameters are jointly trained until convergence. However, this method relies highly on the correlation between the new task and the old task, which will cause task confusion when the tasks' difference is significant. The training time increases linearly with the number of tasks. The Learning Without Memorizing (LWM) [27] is also a knowledge distillation approach without the necessity to preserve the old samples. This method employs an information preserving penalty that uses the attention distillation loss to obtain changes in the model's attention graph and preserve old knowledge. Instead of sequentially extracting knowledge only from the model of the penultimate task, the Multi-model and Multi-level Knowledge Distillation for incremental learning (M2KD) [30] directly applies all previous model information. It also proposes an additional distillation term that runs in the middle layer of the network in addition to the fully connected layer.

Based on the above analysis, this paper adopts the knowledge distillation strategy. Each incremental phase uses a well-trained model on the old classes to initialize the model that will learn the new classes. The advantage of this method is that it does not require any change in the model structure, while preserving the knowledge that the model has learned.

2.3. Solution to Class Imbalance

The sample number of new class is always more than each old class. The incremental Classifier and Representation Learning (iCaRL) [8] uses old sample preservation and knowledge distillation to avoid catastrophic forgetting, and uses cross-entropy loss to classify targets. This method introduces a nearest class mean classifier by computing the class mean of the samples in the sample feature representation. Because this process is independent of the weights and biases of the last layer, the class imbalance problem is well solved. A helpful solution strategy is also proposed in the End-to-End Incremental Learning (EEIL) [12] method, where it includes a phase called "balanced training" at the end of each training session. In this phase, the same number of samples from all visible classes are used to perform a small batch of fine-tuning. However, when these samples are not fully representing the class distribution, balancing training is likely to over-fitting the stored samples. Hou et al. [28] propose to learn a unified classifier incrementally via rebalancing (United), which uses a cosine normalization layer to replace the standard Softmax layer, and combines the fine-tuning technique to improve classification performance. Wu et al. [29] use a two-phases training approach to train a new task using distillation loss and cross-entropy loss in the first phase and correct the imbalance problem using a validation set selected in advance in the second phase. Belouadah et al. [31] propose the class Incremental Learning with dual Memory (IL2M) method to modify network predictions. The imbalance is corrected using the deterministic statistics of class predictions saved from previous tasks. However, most of the methods mentioned above require a long offline training time, and each incremental phase takes a lot of time to obtain good performance.

Based on the above research, class imbalance will cause bias, namely the last fully connected layer of the model has bias in target recognition. The classification logits of the new targets are larger than the old targets, and the model is easier to recognize the sample as the new targets. To solve this problem, a bias correction method for class imbalance is introduced in this paper. A bias correction layer is added after the last fully connected layer of the model, and a linear function with two parameters is used to correct the bias. It solves the problem of bias in the fully connected layer due to class imbalance and achieves superior performance in incremental learning.

3. Methodology

The proposed SAR target incremental recognition method based on hybrid loss function and class-bias correction is shown in Figure 2. Firstly, a small number of old target class samples are preserved and input into the model along with the new target class samples. Secondly, the model is trained using knowledge distillation loss, cross-entropy loss, and class separation loss, where the class separation loss is to reduce the confusion between target classes. Thirdly, a bias correction layer is introduced to correct the bias in the fully connected layer since the imbalance between the number of old and new samples will lead to bias in the fully connected layer of the model.

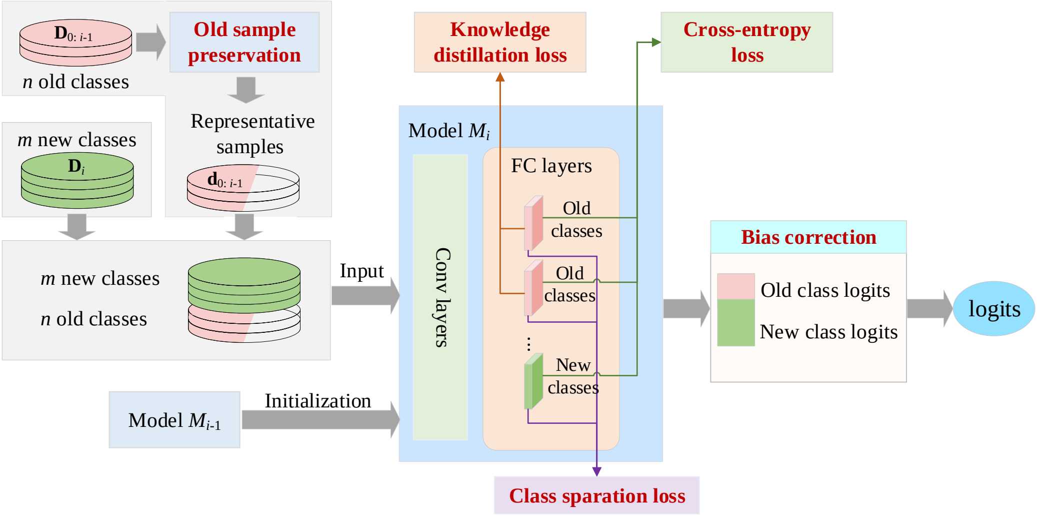


Figure 2. Illustration of one incremental phase of the proposed SAR target incremental recognition method based on hybrid loss function and class-bias correction.

For the convenience of presenting the proposed method, some notations of incremental learning are defined in the following. Incremental learning assumes that one or a batch of new class samples arrives at a time, the model learns multiple tasks in turn, and each task contains several new classes. It is assumed that there are totally $N + 1$ phases of incremental learning (1 initial phase and N incremental phases). In the initial phase, cross-entropy classification loss is used to learn model $\mathbf{M}_0$ by the old classes of data $\mathbf{D}_0$, and then preserve model M_0 to memory. Due to the memory limitation, the whole $\mathbf{D}_0$ is difficult to be preserved, so a small number of samples $\mathbf{d}_0$ is preserved instead of $\mathbf{D}_0$. $\mathbf{d}_0$ is a tiny subset of the old classes data $\mathbf{D}_0$ and $\mathbf{d}_0 \subset \mathbf{D}_0$ and $|\mathbf{d}_0| \ll |\mathbf{D}_0|$. In i-th incremental phase having n old classes and m new classes, a new model M_i is trained to classify the $n + m$ classes. The previous $\mathbf{d}_0 \sim \mathbf{d}_{i-1}$ is abbreviated as $\mathbf{d}_{0:i-1}$. M_{i-1} and $\mathbf{d}_{0:i-1}$ are loaded from memory and $\mathbf{d}_{0:i-1}$ is combined with the new classes data $\mathbf{D}_i$ to train M_i initialized by M_{i-1}. In the incremental phase, the model M_i is trained using the hybrid loss functions and bias correction method. The goal is to train a model that performs well on all currently observed classes without catastrophic forgetting.

3.1. Old Sample Preservation by Herding Selection

The old sample preservation approach refers to retaining a small portion of the old classes data during incremental learning. According to existing studies, keeping a tiny subset of old classes is able to significantly improve the recognition rate of old classes by the model. Algorithm 1 describes the selection process of representative samples. When one or a group of new target classes are added in the training phase, the most representative subset of samples is selected and stored in memory. The number of representative samples for each class, denoted by k, is set the same (k = 20 in this paper).

The old sample preservation strategy in this paper is based on herding selection [34]. According to the distance from each sample to the mean value of the sample, a ranked list of samples within the class is generated. For each class, select the samples closest to the center of the current class, namely the top k samples among them are selected based on the list of samples after sorting. The advantage of herding selection is that the final sample mean is closest to the actual class mean. In terms of the mean, these samples we selected are most representative and able to represent the class better. Therefore, this paper chooses this method to reserve representative samples for each target class.

Algorithm 1: Old Sample Preservation Process by Herding Selection.

```
Input: Image set $\boldsymbol{x} = \{x_1, x_2, ..., x_n\}$ of class label $y$; Samples size $k$;
Input: Trained model $M$;
Use model $M$ to obtain the feature map $F_{\boldsymbol{x}}$ and feature function $\phi$;
Let $\mathbf{d}$ be an empty set;
Compute the class mean $\mu$ through $F_{\boldsymbol{x}}$;
  for 1, 2, 3, ..., $k$ do
    $d_k \leftarrow \arg\min_{x \in \boldsymbol{x}} \left\| \mu - \frac{1}{k}[\phi(x) + \sum_{j=1}^{k-1} \phi(d_j)] \right\|$
  end for
  $\mathbf{d} \leftarrow (d_1, ..., d_k)$;
Output: Representative samples set $\mathbf{d}$;
```

3.2. Hybrid Loss Function for Knowledge Distillation and Class Separation

In this paper, the knowledge distillation loss function L_{KD}, the classified cross-entropy loss function L_{CE}, and the class separation loss function L_{SP} are combined to enable the model to achieve better plasticity-stability balance and forget less about the previously learned knowledge. The knowledge distillation loss is applied to the classification layer of the old target, and the cross-entropy loss function is applied to the classification layer of all targets. The class separation loss is used to separate the old and new target samples better. The hybrid loss function L is denoted as:

$$L = \lambda L_{KD}(x) + (1 - \lambda) L_{CE}(x) + L_{SP}(x) \quad (1)$$

$$\lambda = \sqrt{\frac{n}{n+m}} \quad (2)$$

where λ is a dynamically changing weight coefficient defined by (2), n and m denote the number of old and new classes at each phase, respectively. As the incremental learning phases increase (m is fixed and n increases), the value of λ gradually becomes larger, and the model increasingly tends to retain the existing knowledge.

3.2.1. Loss Function for Knowledge Distillation

In this paper, the knowledge distillation loss L_{KD} is used to retain the knowledge of old targets. It is capable of making the current model simulate the old model trained on the old classes. The output value of the old model is set as an additional supervised signal to assist the training of the new model [35].

$$L_{KD}(x) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{c} d(p_{ij}) log[d(q_{ij})] \tag{3}$$

where q_{ij} denotes the probability of i-th sample belonging to j-th class predicted by the model, p_{ij} is the one-hot class label vector of j-th class corresponding to i-th sample. $d(p_{ij})$ and $d(q_{ij})$ are modified versions of p_{ij} and q_{ij} [12]:

$$d(p_{ij}) = \frac{exp(\widehat{o}_i(x)/T)}{\sum\limits_{j=1}^{n} exp(\widehat{o}_j(x)/T)} \tag{4}$$

$$d(q_{ij}) = \frac{exp(o_i(x)/T)}{\sum\limits_{j=1}^{n} exp(o_j(x)/T)} \tag{5}$$

T is the distillation parameter. $o(x)$ represents the output logits of the current model, $\widehat{o}(x)$ represents the output logits of the old model in the previous increment process. $T = 1$ denotes the common Softmax output probability. However, when $T > 1$, the classification probability distribution is more moderate, making the remaining classes have a greater impact. Their higher loss function values must be reduced to a minimum. The model needs to learn more fine-grained classification. Therefore, the model learns a more distinctive representation of the class to retain the knowledge of the old classes. T is empirically set to 2 for all experiments in this paper.

In this paper, the cross-entropy loss is used to learn to recognize new targets. The cross-entropy loss is used as classification loss L_{CE} to predict the difference or closeness of the output to the true sample labels.

$$L_{CE}(x) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{c} p_{ij} log[q_{ij}] \tag{6}$$

where n and c denote the number of samples and the number of classes, respectively. When the difference between the expected probability and the true sample label increases, their cross-entropy increases.

3.2.2. Loss Function for Class Separation

As mentioned above, the second problem of incremental learning is that the old and new target classes are easily confused when similar target classes increase. With the arrival of new target classes in the training phase, on the one hand, the total number of target classes that the model needs to recognize continues to increase; on the other hand, the visually similar target classes in each training phase will also increase. The boundary between target classes is very sensitive. The increase of visually similar target classes will reduce the recognition accuracy. Moreover, the similarity between different target classes and the difference between the same target classes will indirectly lead to the confusion of the new and old classes in the model, and hence it will produce a series of adverse effects.

Regarding this problem, an class separation loss L_{SP} is designed in this paper to better separate the old and new target samples.

Generally, calculating L_{SP} requires a three-part triplet: benchmark, positive samples, and negative samples [28]. In incremental learning, considering that the number of trainable samples in the old classes is small, it is wise to make full use of these preserved samples to reduce the confusion in the model learning process effectively. In this paper, each sample of old classes is used as a benchmark. The class vector corresponding to the benchmark belonging to the old class (each column of the weight vector in the last layer of the model can be considered as the class vector of each class) is used as the positive samples. The negative samples are the class vector corresponding to the new class that produces a higher response to the benchmark. Benchmark and positive samples belong to the same classes, while negative samples come from different classes. The purpose of using this loss is to maximize the distance between the old and new classes. The distance between samples from the same classes will be as small as possible than the distance between samples from different classes, thus solving the confusion problem between the old and new tasks. The designed loss function L_{SP} is described as

$$L_{SP}(x) = \frac{1}{k}\sum_{i=1}^{n} max[(r + D(f(x_a^i), x_p^i) - D(f(x_a^i), x_n^i)), 0] \tag{7}$$

where r is a hyperparameter, k is the number of preserved samples in the old classes included in each incremental phase, and $f(x)$ denotes the feature vector about sample x obtained after the current model. x_p, x_n are the class vectors corresponding to positive and negative samples. $D(*,*)$ denotes the cosine distance between the two vectors f_1, f_2,

$$D(f_1, f_2) = \frac{f_1 \cdot f_2}{||f_1|| \times ||f_2||} \tag{8}$$

As r is added to the loss function, the value of the loss function is equal to 0 only when the distance between the benchmark and the negative samples exceeds the distance r between the benchmark and the positive samples. Therefore, during the incremental learning process by gradient updating and network training, the distance between different classes will eventually increase to r. Another point to note is the benchmark x, whose corresponding positive and negative samples are not directly from the training sets, but the corresponding class vectors. This allows the class separation loss L_{SP} to be added directly to the incremental learning process without changing the sampling of the data sets.

3.3. Bias Correction Layer for Class Imbalance

Many recent studies [12,21,29,31,32,36] have found that the imbalance between old and new classes will lead to bias in the weights of the fully connected layer. The bias refers to the incremental model being more inclined to the new classes. This is because the model is trained on unbalanced dataset, where the model has accessed to many samples from the new task. However, there are few samples from the older classes from earlier times. Although the forgetting prevention mechanisms, such as old sample preservation and knowledge distillation, have been employed, the model prefers newer classes. A direct result found by Hou et al. [28] is that the classifier norms for new classes are larger than old classes. The classifiers are biased towards new classes.

This paper adopts a simple and practical method to solve the problem of class imbalance, namely a simple bias correction layer is added after the last fully connected layer of the model. It applies a linear function model with two parameters to achieve this function. The linear function is chosen because it is a simple function that does not introduce more hyperparameters to the model and is fast to train. Keep the logits of the output of the old classes $(1, 2, \ldots, n)$ and use this linear function model to correct the bias of the output logits of the new classes $(n+1, \ldots, n+m)$:

$$q_k = as(x) + b \tag{9}$$

$$s(x) = logit(x) = W^T f(x) \tag{10}$$

where a and b are the parameters of the complementary task deviations and $s(x)$ is the output logits, represented by (10). The bias parameters a and b are shared by all new classes, with a = 1, b = 0 under the initial non-incremental phase.

For the parameter a, inspired by [37], in the incremental phase, it is computed by normalizing the weight vector in the fully connected layer. Firstly, the weights W are modified in the fully connected layer in the following form:

$$W = (W_0,\ W_n) \tag{11}$$

$$W_0 = (w_1,\ w_2,\ \ldots,\ w_n) \in \mathbb{R}^{d \times n} \tag{12}$$

$$W_n = (w_{n+1},\ w_{n+2},\ \ldots,\ w_{n+m}) \in \mathbb{R}^{d \times n} \tag{13}$$

Then the weight vector parametrization for the old classes and the new classes are calculated separately:

$$Norm_0 = (||w_1||, \ldots, ||w_n||) \tag{14}$$

$$Norm_n = (||w_{n+1}||, \ldots, ||w_{n+m}||) \tag{15}$$

Based on the above representation, the parameter a is calculated by:

$$a = \frac{Mean(Norm_0)}{Mean(Norm_n)} \tag{16}$$

where $Mean(\cdot)$ denotes the mean value of the elements in the vector. This will make the average norm of new and old classes weight vectors equal. It is worth noting that the relative magnitudes of the weight vector vanes do not change in the new or old classes. Because we only change the average parametrization and make it equal. The critical point of this design is to enable better separation of data within the old and new classes.

For the parameter b, it is optimized on the transformed logits q_k by the representative samples preserved in the old sample preservation step. The representative samples set is a balanced dataset with an equal number of samples from the old and new classes. Because of its small size, it is more appropriate to choose a linear function model with fewer and simpler parameters to correct for bias. More importantly, a similar approach to ours, BIC [29], introduces a separate validation set for this linear function model, while we only use the representative samples set for the knowledge retrospection process. Furthermore, the BIC approach is highly dependent on the validation set, which is only effective for the performance of large-scale dataset. When the validation set is small, it leads to a significant performance degradation. In this paper, the parameter b is optimized by cross-entropy loss, as follows:

$$L_b = -\sum_k d_{y=k} log[softmax(q_k)] \tag{17}$$

where y is the label of the ground-truth class, and $d_{y=k}$ is the indicator function of $y = k$.

3.4. Algorithm Implementation

The method in this paper addresses the catastrophic forgetting problem, recognition confusion problem, and bias problem caused by class imbalance in SAR target incremental learning. Algorithm 2 describes the detailed implementation process of the proposed method.

Algorithm 2: Detailed Implementation Process of the Proposed Method.

```
Input: Train dataset $\mathbf{D}_i$; Test dataset $\mathbf{E}_i$;
Output: New representative sample set $\mathbf{d}_i \subsetneq \mathbf{D}_i$; Model $M_i$;
Output: Test results;
  for $i$ in $(0, 1, \ldots, N)$ do {//1 initial phase and $N$ incremental phases}
    Get $\mathbf{D}_i$;
    if $i = 0$ then {//initial phase}
      Train $M_0$ by $\mathbf{D}_0$;
      Select samples $\mathbf{d}_0 \subsetneq \mathbf{D}_0$ by Algorithm 1;
    else {//incremental phases}
      Load $\mathbf{d}_{0:i-1}$ from memory;
      Initial $M_i$ with $M_{i-1}$;
      Train $M_i$ on $\mathbf{d}_{0:i-1} \cup \mathbf{D}_i$ by (1);
      Correct the bias of the fully connected layer by (9);
    end if
    Run a test with $\mathbf{E}_i$ and record the results;
    Update samples $\mathbf{d}_i$ by Algorithm 1;
  end for
```

4. Experiments and Discussions

To demonstrate the effectiveness of the proposed method, experiments on the MSTAR dataset are carried out. The dataset and experiment settings are introduced. The three comparative methods, including iCaRL, United, and CBesIL, are selected. The analysis of experiment results, including comparison with existing methods, time consumption, ablation experiments, preserved sample size, and confusion matrix comparison, are presented to verify the proposed methods from different aspects.

4.1. Experiment Dataset

The dataset samples used in the experiment of this paper are from the MSTAR co-database. The dataset is obtained from a cluster-beam SAR with a resolution of 0.3 m × 0.3 m, which operates in X-band and HH polarization. The dataset contains 10 classes of military targets, such as armored vehicles, tanks, rocket launchers, etc. Figure 3 shows the SAR images of 10 classes of targets and their corresponding optical image training examples. According to the official recommendations given by the MSTAR dataset, the samples with a depression angle of 17° are selected as the training dataset, and the samples with a depression angle of 15° are selected as the test dataset in this experiment.

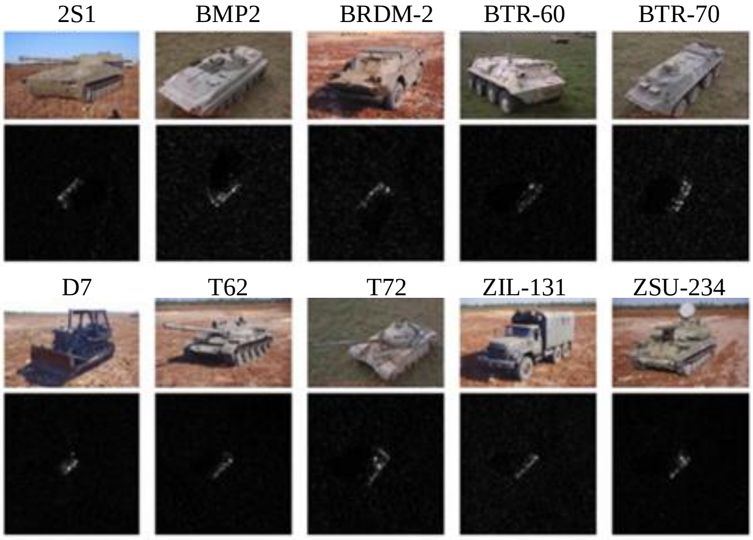


Figure 3. Optical image (**top**) and SAR image (**bottom**) of 10 types of targets.

4.2. Experiment Setting

The ResNet-18 [38] is used as the base network for experiments. Each old class is saved with k = 20 samples in a representative memory of old sample preservation. The procedure is implemented based on the deep learning framework PyTorch [35]. The training contains 120 Epochs, and the initial learning rate is set to 0.01. After the 40th and 80th Epochs, the learning rate is reduced to 0.001 and 0.0001, respectively. The model is trained by stochastic gradient descent throughout the incremental learning phase.

The MSTAR dataset consists of 10 target classes, and the image size is 128 × 128 pixels. The number of data for each training batch is set to 64. After random flipping and cropping, the dataset is used as input of the model. No additional data preprocessing is performed beyond that. The number of new classes for each experiment is set to be 1, 2, and 5 training classes, so 10, 5, and 2 incremental training phases will be performed, respectively. In each different incremental training phase, the 10 classes are arranged in a fixed random order. Experimentally all methods are trained in class increments on the available dataset at each phase. The generated models are evaluated based on test dataset consisting of all old and new classes. This paper follows a protocol for assessing incremental learning [8,39], and the results of the experiments are reported as classification accuracy curves for each incremental batch. In addition, this paper creates three different random classification orders to run three experiments, and reports the average incremental accuracy and standard deviation.

4.3. Comparison Methods

Some existing incremental learning methods are very similar because they use representative sample memory with a fixed capacity, so the performance of the proposed method is compared with the following typical incremental learning methods.

- iCaRL [8]: iCaRL is a representative method in incremental learning. It preserves the samples of old target classes, applies distillation loss to preserve weights, and uses the nearest class mean classifier in the feature space. This experiment reviews iCaRL's approach of using network output to classify and the approach of using nearest-mean-of-exemplars classification (which requires samples to compute past class means and thus works only when old samples are retained). They are called iCaRL-CNN and iCaRL-NME, respectively.
- Unified [28]: the Unified approach introduces cosine normalization and fine-tuning methods. It uses model output for classification, and introduces a less forgetting constraint that preserves the geometric structure of the old classes. Unified also uses old sample preservation and knowledge distillation loss. Two versions of this method, called Unified-CNN and Unified-NME, are also experimented in this experiment.
- CBesIL [6]: The CBesIL method is based on the selection of class boundary samples. This method saves the recognition ability of old samples in the form of class boundary samples and proposes a class boundary selection method based on local geometric and statistical information. A class boundary-based data reconstruction method is used to update the sample set during the incremental process continuously.

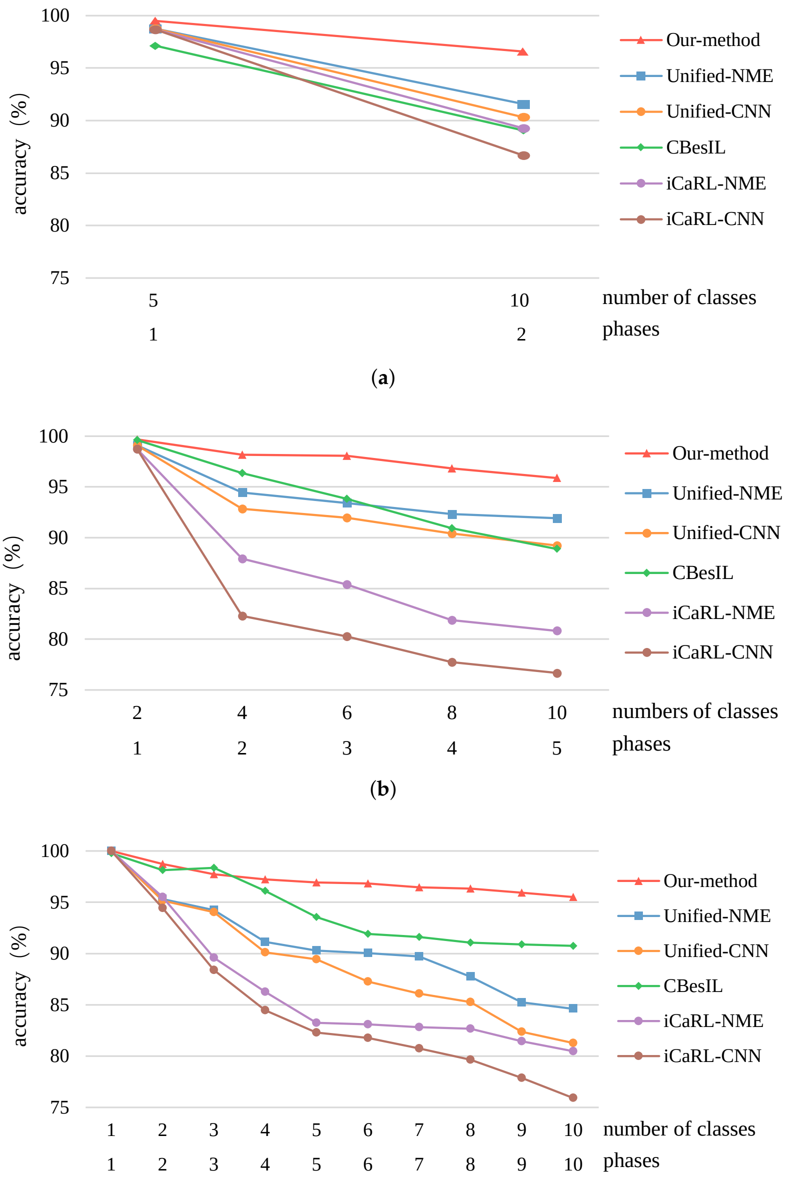
The class order of all experimental comparison approaches is the same to ensure comparability of results.

4.4. Experimental Results

4.4.1. Target Recognition Accuracy

As described in Section 4.2, the incremental learning experiments on the MSTAR dataset start with a pre-trained model on a portion of the old classes data. The incremental process can be divided into 2-phases, 5-phases, and 10-phases increments depending on the number of new target classes each time. 5-phases increment means the process of initial training on 2 classes of old data, followed by increments of 2 classes of new data each time. The rest of the incremental phases are similar to the above process.

Table 2 shows the comparison of our method with existing methods. All tables show the average incremental accuracy and standard deviation of each method at different incremental stages. It can be seen that our method has a smaller standard deviation and less forgetting of old samples as new samples arrive. Figure 4 shows the accuracy differences between the proposed method and the comparison methods. The proposed method achieves superior performance at different incremental phases. The accuracy improvement is about 6% to 8%. It solves well the imbalance problem between old and new target classes, the bias of the fully connected layer for the old and new tasks.



(**a**)

(**b**)

(**c**)

Figure 4. Target incremental recognition accuracy at different incremental phases and different classes per phase. (**a**) 2 phases, 5 classes per phase. (**b**) 5 phases, 2 classes per phase. (**c**) 10 phases, 1 class per phase.

It can be also observed from Figure 4 and Table 2 that the more incremental phases there are, the more significant forgetting and accuracy decreasing. Namely, more incremental phases imply more forgetting. The proposed method significantly reduces forgetting compared with other methods. In the 10-phase increments with 1 class per phase, as shown in Figure 4c), the model by the proposed methods shows more significant differences from the other models, degrading slowly throughout the run against forgetting ("gradual forgetting"). In contrast, the other methods show a significant performance degradation at the beginning of the run ("catastrophic forgetting"). It is worth noting that the accuracy becomes better due to the mixture of bias correction methods and class separation loss. They solve the problem of biased output of the model and identification of confusion very well.

Table 2. Average incremental accuracy for 0ur-method vs. state of the art (average incremental accuracy ± standard deviations).

Methods	Average Incremental Accuracy ± Standard Deviations 2 Phases	5 Phases	10 Phases
iCaRL-CNN	92.68 ± 8.48	83.13 ± 8.98	84.55 ± 7.61
iCaRL-NME	93.97 ± 6.65	86.95 ± 7.14	86.51 ± 6.54
United-CNN	94.54 ± 5.96	92.71 ± 3.82	89.10 ± 5.91
United-NME	95.17 ± 5.07	94.23 ± 2.88	90.83 ± 4.08
CBesIL	93.10 ± 5.71	93.92 ± 4.26	94.21 ± 3.53
Our-method	98.04 ± 2.06	97.73 ± 1.44	97.17 ± 1.35

4.4.2. Time Consumption

The advantage of incremental learning is that it is able to significantly reduce the time consumption required for model training. The purpose of using incremental learning for SAR ATR is to occupy smaller memory space, obtain faster training speed, and consume less time. Non-incremental learning (one-time supervised learning or batch learning) means that the model training phase uses all data from each target class, which is the common joint training approach. The experiment is run under windows environment, the computer cpu is intel core i5-8500, the gpu is NVIDA GeForce RTX 2070, the RAM is 16 g. We first start training on data containing 5 target classes, and then joint training obtains all data from each class each time. Incremental training adds 1 new class of data at a time, while only representative samples of the old data from the past are preserved. As shown in Figure 5, during the total of 6 phases, the incremental learning approach significantly reduces the time consumption. The time required for incremental training remains essentially in the same dimension as the new target arrives. However, the time consumption of non-incremental training continues to increase and exceeds the incremental methods. This shows that the incremental learning method proposed in this paper is able to obtain a better recognition rate and less time consumption.

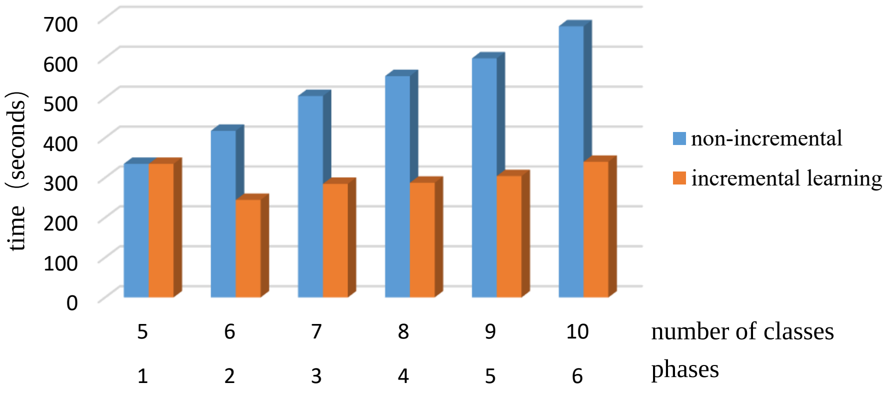


Figure 5. Comparison of non-incremental and incremental learning time consumption.

4.4.3. Ablation Experiments

The method proposed in this study combines knowledge distillation loss, cross-entropy loss, class separation loss, and bias correction to deal with catastrophic forgetting, old and new class confusion, class imbalance. To analyze the impact of different components on incremental learning performance, each of the following three combinations of methods are compared.

- Hybrid 1: training the model using cross-entropy loss (L_{CE}) and knowledge distillation loss (L_{KD}).
- Hybrid 2: training the model using cross-entropy loss (L_{CE}), knowledge distillation loss (L_{KD}), and class separation loss (L_{SP}).
- Our-method: the model is trained using cross-entropy loss (L_{CE}), knowledge distillation loss (L_{KD}), class separation loss (L_{SP}), and bias correction (Bic correction) methods.

The experiments are conducted under a 5-phases incremental process, with 2 new classes incremented each time. Table 3 reports the average incremental accuracy and standard deviation with different components. As shown in Figure 6 and Table 3, Hybrid 1 uses knowledge distillation loss to preserve the knowledge of the old classes, the confusion between the old and new classes gradually increased with the incremental phases, and the model performance degradation is significant. Hybrid 2 is better than Hybrid 1 because the class separation loss focusing on class boundaries increases the distance between samples of different classes and reduces the distance within samples of the same class, which solves the classification confusion problem well. However, due to class imbalance between old and new classes, the recognition accuracy further decreases after multiple phases of incremental training. When the bias correction method is added to the training, the classification preference of the fully connected layer for the new class is corrected. The model's performance improves significantly and shows a slow forgetting during incremental learning. These results suggest that the bias problem caused by a class imbalance in incremental learning with exemplars significantly affects the incremental learning performance. Our proposed method is able to further improve the model recognition ability and achieve a perfect stability-plasticity compromise: it maintains the model's ability to discriminate between old tasks and learn new tasks well.

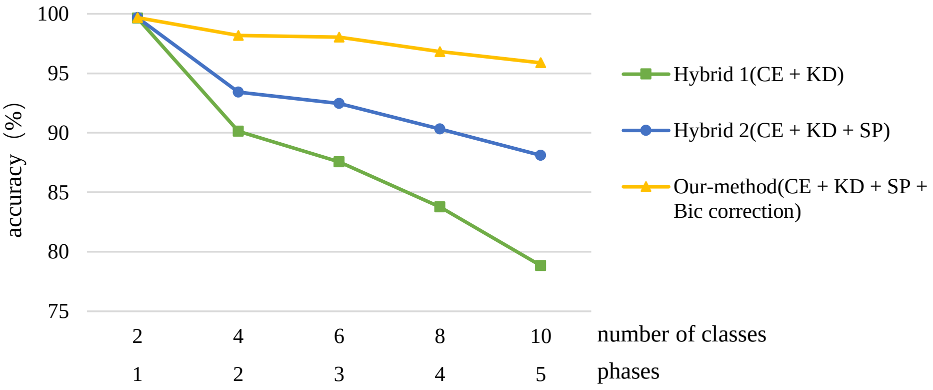


Figure 6. The effect of different loss functions on incremental learning performance.

Table 3. The impact of different components on incremental learning performance (average incremental accuracy ± standard deviation).

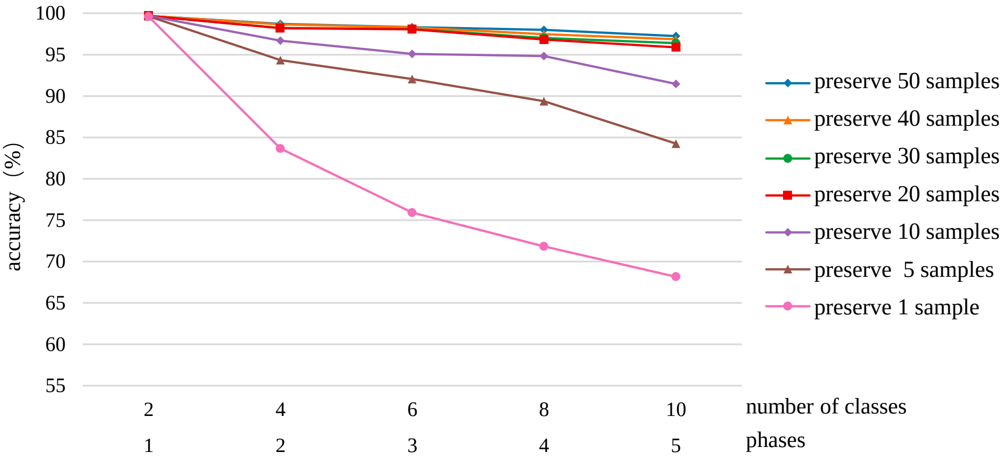
Methods	Average Incremental Accuracy ± Standard Deviations 5 Phases
hybrid 1(CE+KD)	87.99 ± 7.78
hybrid 2(CE+KD+SP)	92.80 ± 4.35
Our-method(CE+KD+SP+Bic correction)	97.72 ± 1.44

4.4.4. Preserved Sample Size Analysis

The proposed method preserves a small number of old samples, which leads to a significant increase in the accuracy of incremental learning. The number of preserved samples is important since different numbers of old samples will have different effects on the performance of incremental learning methods. Table 4 shows the average incremental accuracy and standard deviation for preserving different sample sizes. The larger the number of preserved samples, the smaller the standard deviation, i.e., the smaller the data fluctuation. As shown in Figure 7 and Table 4, under the incremental training of 5 and 10 phases, the model recognition performance degrades significantly when the number of preserved old samples is 1, 5, and 10 samples per class. At the same time, the improvement is not significant when 30, 40, and 50 samples are preserved compared with 20 samples (the curves of k = 20, 30, 40, 50 almost overlap). In all, considering the memory space occupation, this paper sets the number of samples preserved for each old class to be 20, i.e., k = 20.

Table 4. Average incremental accuracy for different preserved sample sizes (average incremental accuracy ± standard deviations).

Number of Preserved Samples	Average Incremental Accuracy ± Standard Deviations	
	5 Phases	10 Phases
1	79.83 ± 12.45	76.61 ± 15.24
5	91.92 ± 5.72	87.97 ± 8.16
10	95.54 ± 2.99	93.48 ± 3.93
20	97.72 ± 1.44	97.17 ± 1.34
30	97.87 ± 1.25	97.58 ± 1.39
40	98.18 ± 1.09	97.87 ± 1.23
50	98.39 ± 0.90	98.21 ± 1.03



(**a**)

Figure 7. *Cont.*

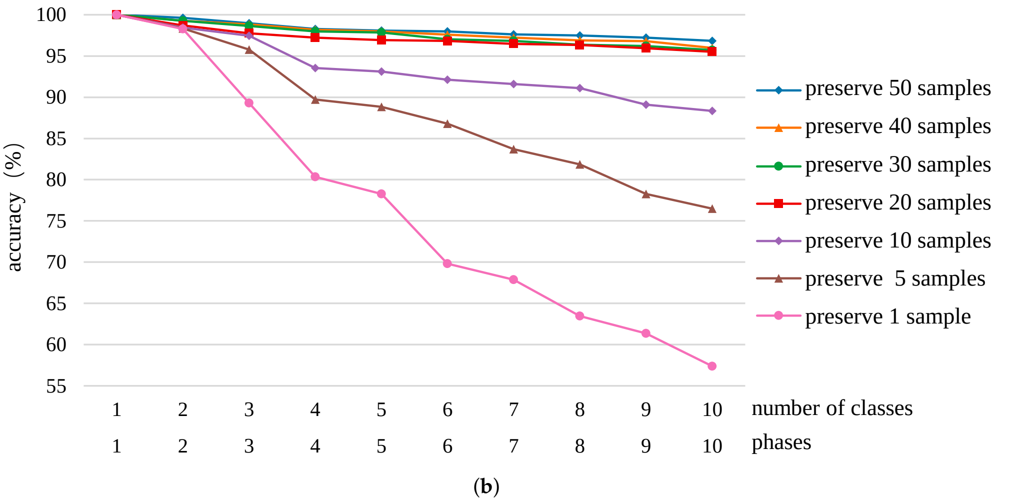


Figure 7. The effect of different preserved sample sizes on incremental learning performance. (**a**) 5 phases, 2 classes per phase. (**b**) 10 phases, 1 class per phase.

4.4.5. Confusion Matrix Comparison

Figure 8 shows the comparison of the proposed incremental learning method with the United-NME and iCaRL-NME methods confusion matrix on the test dataset, and this experiment is performed under a 5-phases incremental process. As shown in Figure 8, the iCaRL-NME method is greatly affected by the imbalance between the old and new classes, and the classification bias is more serious. It is more likely to classify the input test samples as the new classes, and the overall accuracy is poor. The United-NME method is able to better alleviate the class imbalance due to its use of cosine normalization, which solves this problem to some extent. In general, this paper uses knowledge distillation loss and class separation loss to reduce forgetting and confusion. It also corrects the bias of the fully connected layer, which better solves the bias problem caused by classes imbalance. As shown in Figure 8c, the proposed method identifies the old classes better, avoids the problems of catastrophic forgetting and classification confusion well, and solves the problem of class imbalance better.

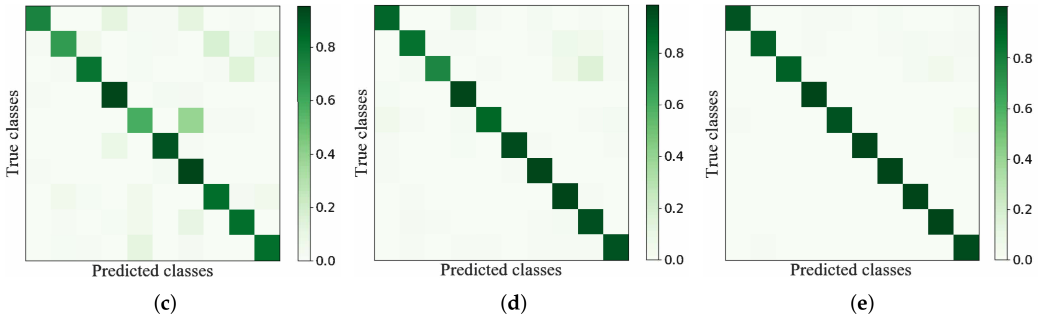


Figure 8. Comparison of confusion matrices under three methods while the experiment is carried out under the condition of 5-phase incremental learning. (**a**) iCaRL-NME. (**b**) United-NME. (**c**) Our-method.

5. Conclusions

This paper presents a new approach to tackle the problem of incremental learning-based SAR target recognition. Through research, it is found that the bias caused by the imbalance between old and new classes is the primary factor that influences the performance of incremental learning. Our incremental approach deals with these problems at different levels and achieves a good balance between the retention of old knowledge and the learning of new knowledge. First, this paper combines the knowledge distillation method and old sample preservation method that preserve representative samples to reduce the forgetting of old classes. Then an effective loss function for class separation is used to reduce the confusion between old and new classes. Finally, we find that the fully connected layer of the deep learning model has a stronger tendency to favor new classes in classification. Therefore, a linear model is used to correct the bias problem of the fully connected layer. Experiments on the MSTAR dataset show that the proposed approach in this paper has superior performance to existing methods.

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

Funding: This research was funded in part by the National Natural Science Foundation of China under Grant 62171016 and 61871413, and in part by the Fundamental Research Funds for the Central Universities under Grant buctrc202001.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

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

Abbreviations

The following abbreviations are used in this manuscript:

SAR	Synthetic aperture radar
ATR	Automatic target recognition
iCaRL	incremental classifier and representation learning
CBesIL	Class boundary exemplar selection based incremental learning

References

1. Cracknell, A.P.; Varotsos, C.A. New aspects of global climate-dynamics research and remote sensing. *Int. J. Remote Sens.* **2011**, *32*, 579–600. [CrossRef]
2. Zhang, F.; Wang, Y.; Ni, J.; Zhou, Y.; Hu, W. SAR Target Small Sample Recognition Based on CNN Cascaded Features and AdaBoost Rotation Forest. *IEEE Geosci. Remote Sens. Lett.* **2020**, *17*, 1008–1012. [CrossRef]
3. Zhang, F.; Liu, Y.; Zhou, Y.; Yin, Q.; Li, H.C. A lossless lightweight CNN design for SAR target recognition. *Remote Sens. Lett.* **2020**, *11*, 485–494. [CrossRef]
4. Chen, S.; Wang, H.; Xu, F.; Jin, Y.Q. Target Classification Using the Deep Convolutional Networks for SAR Images. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 4806–4817. [CrossRef]
5. Wang, Z.; Wang, C.; Pei, J.; Huang, Y.; Zhang, Y.; Yang, H.; Xing, Z. Multi-View SAR Automatic Target Recognition Based on Deformable Convolutional Network. In Proceedings of the 2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS, Brussels, Belgium, 11–16 July 2021; pp. 3585–3588. [CrossRef]
6. Dang, S.; Cao, Z.; Cui, Z.; Pi, Y.; Liu, N. Class boundary exemplar selection based incremental learning for automatic target recognition. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 5782–5792. [CrossRef]
7. Lu, X.; Sun, X.; Diao, W.; Feng, Y.; Wang, P.; Fu, K. LIL: Lightweight Incremental Learning Approach Through Feature Transfer for Remote Sensing Image Scene Classification. *IEEE Trans. Geosci. Remote Sens.* **2021**, 1–20. [CrossRef]
8. Rebuffi, S.A.; Kolesnikov, A.; Sperl, G.; Lampert, C.H. iCaRL: Incremental classifier and representation Learning. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 5533–5542. [CrossRef]
9. Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A.A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; et al. Overcoming catastrophic forgetting in neural networks. *Proc. Natl. Acad. Sci. USA* **2017**, *114*, 3521–3526. [CrossRef] [PubMed]
10. Peng, J.; Tang, B.; Jiang, H.; Li, Z.; Lei, Y.; Lin, T.; Li, H. Overcoming Long-Term Catastrophic Forgetting Through Adversarial Neural Pruning and Synaptic Consolidation. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, 1–14. [CrossRef] [PubMed]
11. Lee, J.J.; Il Lee, S.; Kim, H. Continual Learning for Instance Segmentation to Mitigate Catastrophic Forgetting. In Proceedings of the 2021 18th International SoC Design Conference (ISOCC), Jeju Island, Korea, 6–9 October 2021; pp. 85–86. [CrossRef]
12. Castro, F.M.; Marín-Jiménez, M.J.; Guil, N.; Schmid, C.; Alahari, K. End-to-end incremental learning. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 233–248. [CrossRef]

13. Fernando, C.; Banarse, D.; Blundell, C.; Zwols, Y.; Ha, D.; Rusu, A.A.; Pritzel, A.; Wierstra, D. Pathnet: Evolution channels gradient descent in super neural networks. *arXiv* **2017**, arXiv:1701.08734.
14. Golkar, S.; Kagan, M.; Cho, K. Continual learning via neural pruning. *arXiv* **2019**, arXiv:1903.04476.
15. Li, Z.; Hoiem, D. Learning without forgetting. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 2935–2947. [CrossRef] [PubMed]
16. Lopez-Paz, D.; Ranzato, M. Gradient episodic memory for continual learning. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Volume 30, pp. 6467–6476.
17. Shin, H.; Lee, J.K.; Kim, J.; Kim, J. Continual learning with deep generative replay. *arXiv* **2017**, arXiv:1705.08690.
18. Kemker, R.; Kanan, C. FearNet: Brain-inspired model for incremental learning. In Proceedings of the 6th International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
19. Hinton, G.; Vinyals, O.; Dean, J. Distilling the knowledge in a neural network. *Comput. Sci.* **2015**, *14*, 38–39.
20. Dastidar, S.G.; Dutta, K.; Das, N.; Kundu, M.; Nasipuri, M. Exploring knowledge distillation of a deep neural network for multi-script identification. In Proceedings of the International Conference on Computational Intelligence in Communications and Business Analytics, Santiniketan, India, 7–8 January 2021; Springer: Berlin/Heidelberg, Germany, 2021; pp. 150–162.
21. Dong, Q.; Gong, S.; Zhu, X. Class rectification hard mining for imbalanced deep learning. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 27–29 October 2017; pp. 1869–1878. [CrossRef]
22. Aljundi, R.; Chakravarty, P.; Tuytelaars, T. Expert gate: Lifelong learning with a network of experts. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 7120–7129. [CrossRef]
23. Mai, Z.; Li, R.; Jeong, J.; Quispe, D.; Kim, H.; Sanner, S. Online continual learning in image classification: An empirical survey. *Neurocomputing* **2022**, *469*, 28–51. [CrossRef]
24. Bhat, S.D.; Banerjee, B.; Chaudhuri, S.; Bhattacharya, A. CILEA-NET: Curriculum-Based Incremental Learning Framework for Remote Sensing Image Classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 5879–5890. [CrossRef]
25. Park, W.; Kim, D.; Lu, Y.; Cho, M. Relational knowledge distillation. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 3962–3971. [CrossRef]
26. Wu, Y.; Chen, Y.; Wang, L.; Ye, Y.; Liu, Z.; Guo, Y.; Zhang, Z.; Fu, Y. Incremental classifier learning with generative adversarial networks. *arXiv* **2018**, arXiv:1802.00853.
27. Dhar, P.; Singh, R.V.; Peng, K.C.; Wu, Z.; Chellappa, R. Learning without memorizing. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 5133–5141. [CrossRef]
28. Hou, S.; Pan, X.; Loy, C.C.; Wang, Z.; Lin, D. Learning a unified classifier incrementally via rebalancing. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 831–839. [CrossRef]
29. Wu, Y.; Chen, Y.; Wang, L.; Ye, Y.; Liu, Z.; Guo, Y.; Fu, Y. Large scale incremental learning. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 374–382. [CrossRef]
30. Zhou, P.; Mai, L.; Zhang, J.; Xu, N.; Wu, Z.; Davis, L.S. M2kd: Multi-model and multi-level knowledge distillation for incremental learning. *arXiv* **2019**, arXiv:1904.01769.
31. Belouadah, E.; Popescu, A. IL2M: Class incremental learning with dual memory. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27–28 October 2019; pp. 583–592. [CrossRef]
32. Belouadah, E.; Popescu, A. ScaIL: Classifier weights scaling for class incremental learning. In Proceedings of the 2020 IEEE Winter Conference on Applications of Computer Vision (WACV), Snowmass Village, CO, USA, 1–5 March 2020; pp. 1255–1264. [CrossRef]
33. Liu, Y.; Su, Y.; Liu, A.A.; Schiele, B.; Sun, Q. Mnemonics training: Multi-class incremental learning without forgetting. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 12242–12251. [CrossRef]
34. Welling, M. Herding dynamical weights to learn. In Proceedings of the 26th Annual International Conference on Machine Learning, Montreal, QC, Canada, 14–18 June 2009; pp. 1121–1128.
35. Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; Lerer, A. Automatic differentiation in pytorch. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017.
36. Aydin, M.A. Using Generative Adversarial Networks for Handling Class Imbalance Problem. In Proceedings of the 2021 29th Signal Processing and Communications Applications Conference (SIU), Istanbul, Turkey, 9–11 June 2021; pp. 1–4. [CrossRef]
37. Zhao, B.; Xiao, X.; Gan, G.; Zhang, B.; Xia, S.T. Maintaining discrimination and fairness in class incremental learning. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 13205–13214. [CrossRef]
38. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [CrossRef]
39. Liu, X.; Masana, M.; Herranz, L.; Van de Weijer, J.; López, A.M.; Bagdanov, A.D. Rotate your networks: Better weight consolidation and less catastrophic forgetting. In Proceedings of the 2018 24th International Conference on Pattern Recognition (ICPR), Beijing, China, 20–24 August 2018; pp. 2262–2268. [CrossRef]