Incremental SAR Automatic Target Recognition With Error Correction and High Plasticity

Jiaxin Tang, Deliang Xiang, *Member, IEEE*, Fan Zhang, *Senior Member, IEEE*, Fei Ma, *Member, IEEE*, Yongsheng Zhou, *Member, IEEE*, and HengChao Li, *Senior Member, IEEE*

***Abstract*—Synthetic aperture radar automatic target recognition (SAR ATR) uses computer processing capabilities to infer the classes of the targets without human intervention. For SAR ATR, deep learning gradually emerges as a powerful tool and achieves promising performance. However, it faces serious challenges of how to deal with incremental recognition scenarios. The existing deep learning-based SAR ATR methods usually predefine the total number of recognition classes. In realistic applications, the new tasks/classes will be added continuously. If all old data are stored and mixed with newly added data to update the model, the storage pressure and time consumption make the application infeasible. In this article, the high plastic error correction incremental learning (HPecIL) is proposed to address the model degradation and plasticity decline in the incremental scenario. Multiple optimal models trained on old tasks are used to correct accumulative errors and alleviate model degradation. Moreover, the sharp data distribution shift due to newly added data can also result in the model underperforming. A class-balanced training batch is constructed to deal with the issue of unbalanced data distribution. To make a tradeoff between model stability and model plasticity, low-effect nodes in the model are removed to boost the efficiency of model update. The proposed HPecIL outperforms the other state-of-the-art methods in incremental recognition scenarios. The experimental results demonstrate the effectiveness of the proposed method.**

***Index Terms*—Automatic target recognition (ATR), incremental learning, synthetic aperture radar (SAR).**

I. Introduction

DUE to the superiorities of all-weather, day-and-night, wide-range, and high-resolution imaging, synthetic aperture radar (SAR) has been widely applied in military reconnaissance, geographic information collection, and change detection [1]. Different from optical imaging, the single-polarized SAR image characterizes the target by scattering intensity. The SAR imaging mechanism results in the shortcomings of blurry details and strong anisotropy. Simultaneously, limited resolution and complex background also increase the difficulty of manual interpretation. SAR automatic target recognition (ATR) uses computer processing capabilities to infer the classes of the targets without human intervention [2].

Previous SAR ATR studies mostly exploited template matching [3], [4] and electromagnetic modeling [5], [6]. For the former class, the lack of templates leads to poor recognition performance. In contrast, the latter class requires complicated electromagnetic knowledge and prior information about targets. With the help of machine learning, hand-craft feature [7], [8], and robust trainable classifier [9], [10] are applied in the SAR ATR field. It is different from the template-based method, which needs to select the best match. Most subsequent research is on selecting an appropriate feature or improving the classifier.

The renaissance of neural networks makes more and more deep learning-based SAR ATR methods emerge. Chen *et al.* [11] proposed an all-convolutional network, which is efficient and accurate. Li *et al.* [12] improved recognition accuracy by adding component information. Wang *et al.* [13] extracted multifarious target attributes from the segmentation task to achieve superior recognition performance. Besides, some other research works focus on training deep neural networks with limited labeled data. CHU-Net [14] utilized a convolutional highway unit to improve the model feature extraction capacity for limited labeled data. MCGAN [15] mixed the encoded features with noise and category labels to generate diverse and correct samples. DKTS-N [16] incorporated SAR domain knowledge related to the azimuth angle, the amplitude, and the phase data of vehicles to extract more information from a small amount of labeled data to avoid overfitting problems. However, humans can accumulate the knowledge step by step, instead of completing it all at once. Neural network-based methods should be able to learn knowledge incrementally.

In the real-life application of SAR ATR, new tasks/classes will continue to be added [17]. The conventional methods assume that training data of all tasks/classes are always available in every training phase. As shown in Fig. 1, the conventional methods adopt joint training to retrain the model in the incremental scenario. Reusing old data will increase time consumption. The

Manuscript received October 12, 2021; revised December 8, 2021; accepted December 29, 2021. Date of publication January 10, 2022; date of current version January 31, 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 61871413, Grant 62171015 and Grant 62171016, and in part by the Fundamental Research Funds for the Central Universities under Grant XK2020-03 and Grant buctrc202121. *(Corresponding author: Fei Ma.)*

Jiaxin Tang, Fan Zhang, Fei Ma, and Yongsheng Zhou are with the College of Information Science and Technology, Beijing University of Chemical Technology, Beijing 100029, China (e-mail: tangchiahsin@outlook.com; zhangf@mail.buct.edu.cn; mafei@mail.buct.edu.cn; zhyosh@mail.buct.edu.cn).

Deliang Xiang is with the Beijing Advanced Innovation Center for Soft Matter Science and Engineering, Interdisciplinary Research Center for Artificial Intelligence, Beijing University of Chemical Technology, Beijing 100029, China (e-mail: xiangdeliang@gmail.com).

HengChao Li is with the School of Information Science and Technology, Southwest Jiaotong University, Chengdu 611756, China (e-mail: lihengchao_78@163.com).

Digital Object Identifier 10.1109/JSTARS.2022.3141485

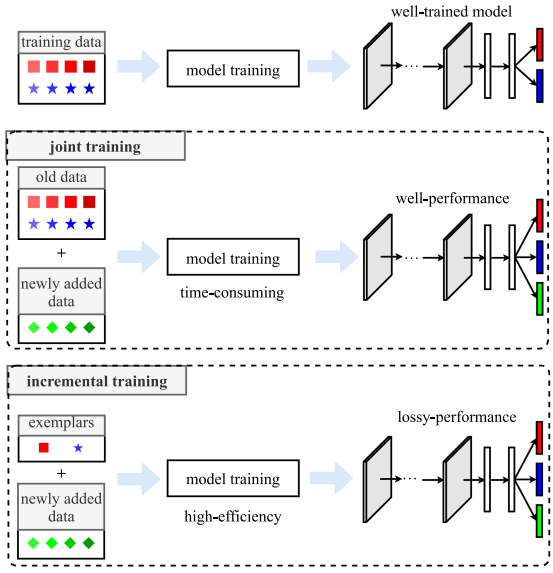

Fig. 1. Illustration for the retraining phase of the conventional methods and incremental learning methods.

amount of newly added data only accounts for a small ratio of the total amount of data, which means that the most resource is used on the old data. However, most well-performance methods exploit the backpropagation algorithm to update parameters. If the old data are unavailable, the knowledge about the old data will be completely lost (i.e., catastrophic forgetting [18], [19]). The data-driven methods suffer from this issue severely. Humans can continuously acquire knowledge when observing new instances. Old concepts may change or be forgotten, but a complete loss is nearly impossible. Incremental learning can continuously acquire knowledge from newly added data and maintain the old knowledge like humans, which is regarded as a more appropriate learning mode to cope with this scenario, where data increase gradually [20].

The up-to-date incremental learning methods can be divided into three main categories: data replay-based methods, weight regularization-based methods, and model growth-based methods. Data replay-based methods [21]–[27] construct a small exemplar set from old data or synthesize samples to keep recognition capacity for old categories. Weight regularization-based methods [28]–[30] penalize the modification of important model parameters to overcome the performance degradation. Model growth-based methods [31]–[36] fix parameters and extend models to cope with old knowledge reservation and new knowledge acquirement. The data replay-based method consumes more training time. The model growth-based method needs extra memory for a growing model. The weight regularization-based method does not require old data and extension models but the performance is unsatisfactory. Applying incremental learning methods directly cannot satisfy the requirements (i.e., high accuracy, low time consumption, and low data loss risk).

Existing incremental SAR ATR methods are based on conventional machine learning [37] and [38] proposed a dynamic feature learning method based on nonnegative matrix factorization (NMF). However, the classifier still needs to be retrained and the recognition performance is not satisfactory. Dang *et al.* [39] and Ma *et al.* [40] improved open set SAR target recognition with incremental learning. Nevertheless, the main focus is on how to cope with unknown categories. CBesIL [17] improved exemplar selection in [39] and synthesizes extra samples. Yet, synthetic data increase computational time. Some common issues in incremental SAR ATR are listed as follows:

1) Error accumulation: As the backpropagation algorithm is used to update the model parameters, the model is only locally optimal for the newly added data. The errors for old classes will be accumulated with continual model updates. None of the existing methods corrects the errors in the subsequent training phase.
2) Stability-plasticity dilemma: Most methods pay much attention to the capacity of preserving knowledge from previous data (i.e., stability) but damage the capacity of acquiring knowledge from newly added data (i.e., plasticity).
3) Class imbalance: The dataset shift is a basic issue in the incremental recognition scenario. The training set distribution changes from old classes to newly added classes. Most methods preserve an exemplar set to solve the issue. However, few methods pay attention to the class imbalance issue degraded from the dataset shift.

To overcome these problems, a plastic error correction incremental learning method is proposed and the overall framework is shown in Fig. 2. In addition to the basic CNNs model structure, the method includes exemplar management, knowledge inheritance, data distribution balance, and pruning initialization parts. The exemplar management part selects key samples to construct a limited exemplar set. The knowledge inheritance part utilizes multiple old models to guide the current model training and correct the accumulation errors. The data distribution balance part constructs class-balanced training batches to avoid a sharp data distribution shift. The pruning initialization part increases the contribution rate of the old data in the training phase to improve the model plasticity.

To summarize, our contributions are the following:

1) The proposed knowledge inheritance preserves multiple optimal models trained on old data to correct the accumulative errors. The current model accumulates errors due to the continuous reduction of old category data in the training phase, which results in the model performance deteriorates. With the guidance of multiple optimal models, the recognition performance of the current model achieves a significant improvement.
2) The data distribution balance is exploited to cope with the class imbalance in the training batches. Due to the emergence of newly added data in the incremental recognition scenario, the exemplars of old classes only account for a small ratio. The catastrophic forgetting degenerates into an imbalance problem. Class-balanced training batches alleviate the sharp shift in the training data distribution and achieve better performance.

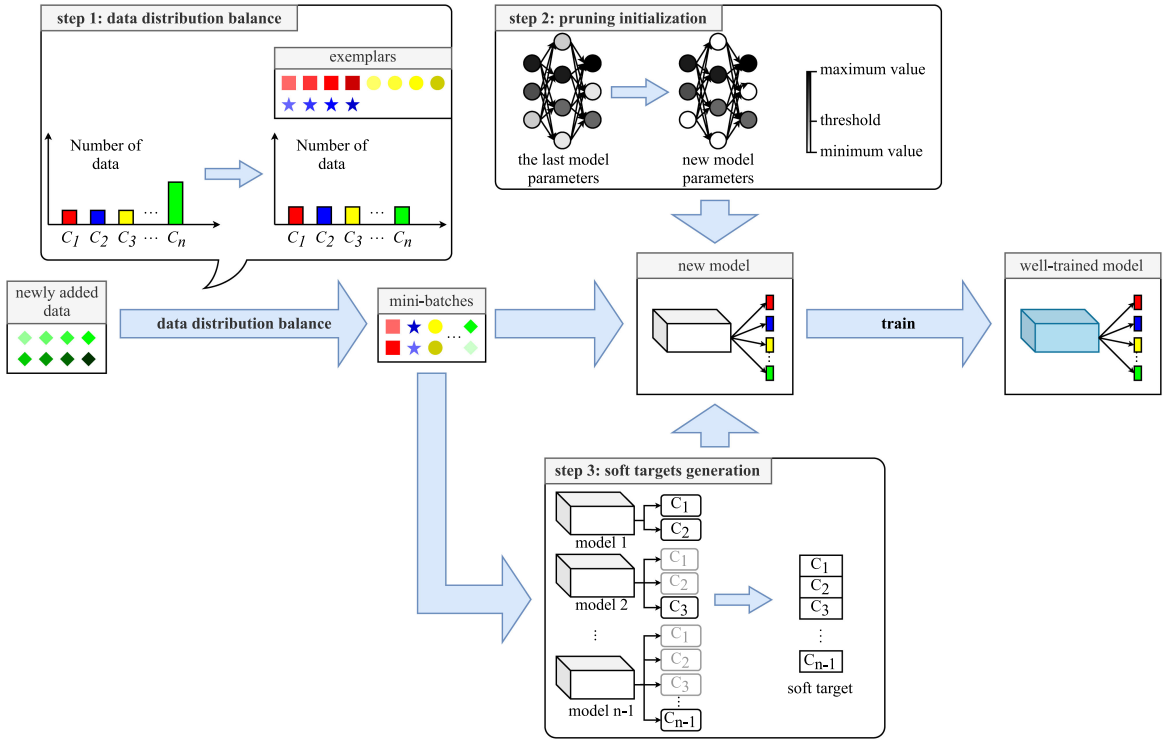

Fig. 2. Proposed incremental recognition framework.

3) The unstructured pruning initialization, instead of the initialization by all parameters of the last model, is used in the proposed HPecIL. The initialization by all parameters of the last model can maintain past recognition performance but reduces the update efficiency of the exemplars to the model. The unstructured pruning initialization removes low-effect nodes, which gives a momentum boost to the model update.

The rest of this article is organized as follows. More related work in incremental learning based on deep learning is discussed in Section II. The details of the proposed method and each part are described in Section III. In Section IV, the experimental setup, experimental results, and analysis are presented. The analysis contains average accuracy comparison, accuracy change of each class per incremental stage, and time consumption. Finally, Section V concludes this article.

II. Related Work

Our work is mainly related to the data replay-based [21]–[23] and weight regularization-based incremental learning [28], [30], [41]. In this section, the common technical terms and basic strategies of incremental learning are described.

A. Definitions of Terms

The common technical terms used in the data replay-based and weight regularization-based incremental learning are described. The proposed HPecIL also follows the same definitions of the terms.

1) *Class-incremental learning* [21]–[23] is a specific incremental learning scenario, which regulates that the new task only includes unseen classes.
2) *Old data* are training data that have been used to update the model parameters.
3) *Newly added data* emerge with the new task. And they will be used to train the new model.
4) *Old models* [41] have been trained on the old data. Their performances are locally optimal for the training classes.
5) *The new/current model* is initialized by the last old model. And it will be updated by the newly added data and exemplars.
6) *Exemplars* [21]–[23] are representative samples selected from old data. They will be added to the training data for the parameter update of the new model.
7) *Soft targets* [30] are old model outputs with the training data as inputs. The soft targets used by the distillation loss are efficient to transfer knowledge to the new/current model.

B. Incremental Learning

In recent years, neural network models have surpassed human-level performance on individual tasks, e.g., object detection, object recognition. While the results of these static models are impressive, they are incapable of adapting the behavior over time [20]. Therefore, these models should be retrained when newly added data emerge. In our realistic world, reusing all training data to update model parameters is intractable for data streams [41]. Moreover, the storage constraints and data loss

risk make the practice unavailable. This calls for a continually adaptive method, which can keep on learning over time.

Incremental learning methods attempt to follow the human cognition system, that is, learn concepts sequentially [32]. According to gradually extending acquired knowledge, incremental learning methods can adapt the data streams from different domains or tasks. The old concepts can be reawakened by observing exemplars, but it is not significant to preserve all old data. The forgetting of partial old knowledge is allowed, but complete loss (i.e., catastrophic forgetting) should not occur.

III. Proposed Method

In this section, the overall framework is described, and the implementation of incremental recognition is explained. Section III-A introduces the data format, basic network architecture, and incremental training phase of the proposed method in an overview perspective. Section III-B presents the main components including exemplars management, knowledge inheritance, data distribution balance, and pruning initialization.

A. Overall Process

1) Data Format: The continuously increasing data streams can be denoted as $X^1, X^2, \ldots$, and the sample set $X^y = \{x_1^y, \ldots, x_{n^y}^y\}$ means that there are n^y samples of class $y \in Y$. Our method selects exemplars from X^y to construct exemplar sets E^y. The total number of exemplars is a fixed number K. The exemplar management algorithm is described in Section III-B1.

2) Architecture: As the underlying support, our method makes use of CNN with residual structure as feature extractor and exploits an adaptive average pooling and linear layer combined with sigmoid function as a classifier. The model is denoted as $\Phi(w, b, \gamma, \beta, x)$, where w includes the weight of the convolutional (conv) layer w_{conv} and the weight of the linear layer w_{fc}, b is the bias of the linear layer, γ is the weight of the batch normalization (BN) layer, β is the bias of the BN layer, and x is the input image. The network architecture is ResNet18. If there are specific requirements, such as model size, inference speed, and feature space, the network architecture used in our method can be replaced by any other architecture.

The feature extractor $\theta(w_{\text{conv}}, \gamma, \beta, x) : \chi \to \mathbb{R}^d$ can learn a feature representation from incremental datasets. The feature map z is obtained by the feature extraction operation defined as:

$$\mathrm{z} = \theta(w_{\text{conv}}, \gamma, \beta, x). \tag{1}$$

Then, the feature map z is reduced to the vector s by the adaptive average pooling operation defined as

$$\mathrm{s} = \mathrm{pool}(\mathrm{z}). \tag{2}$$

The number of known classes is denoted as C, The linear layer $o(w_{fc}, b, \mathrm{s})$ maps s into a vector $\mathrm{v_C}$ of length C. Then, the sigmoid function is used to activate the vector and output the predictions $\hat{y} \in \{1, \ldots, C\}$

$$\mathrm{v_C} = o(w_{fc}, b, \mathrm{s}), \tag{3}$$

$$\hat{y} = \arg\max_C \frac{1}{1 + \exp(-\mathrm{v_C})}. \tag{4}$$

3) Incremental Training Phase: The incremental recognition framework is shown in Fig. 2. The base model architecture has been introduced above. In the first recognition task, the training dataset includes C_1 classes, which determines the output number of the linear layer. Due to the class imbalance, the training set will be preprocessed to balance the data distribution. The balanced training set is inputted into the model to predict the classes. The error between predictions and labels are calculated through the classification loss function, and the model parameters are updated through the backpropagation algorithm. When the training iteration condition is met, the model can predict the first recognition task as follows:

$$\hat{y} = \Phi_1(w, b, \gamma, \beta, x^y), y \in 1, \ldots, C_1, \tag{5}$$

where Φ_1 is the model for the first recognition task. To realize incremental recognition, exemplars are selected from the training dataset to construct an exemplar set based on herding as follows:

$$E^y \leftarrow \mathrm{herd}(X^y). \tag{6}$$

The model after training will be also saved to supervise future model training, the operation can be defined as

$$\mathcal{M} \leftarrow \Phi_1 \tag{7}$$

where $\mathcal{M}$ stores all previous models.

When the tth recognition task brings new data, the model training has changed into the incremental recognition mode. The parameter size of the feature extractor is fixed, while the parameter size of the classifier is variable with the number of known classes. The number of known classes C_t will update by the number of newly added classes C_{new} as follows:

$$C_t = C_{t-1} + C_{\text{new}} \tag{8}$$

where C_{t-1} represents the previous number of know classes. The number of the output of the linear layer of the classifier changes with C_t. The lower weight of the last saved model will be pruned and the rest will be utilized to initialize the tth model. The newly added nodes in the linear layer of the classifier are initialized by (16)–(18). The tth model is initialized as follows:

$$\theta_t = \mathrm{pruning}(\theta_{t-1}) \tag{9}$$

$$o_t = \mathrm{pruning}(o_{t-1}) + \mathrm{initializing}(o_{\text{new}}) \tag{10}$$

where θ_t is the feature extractor of the tth model, θ_{t-1} is the feature extractor of the $(t-1)$th model, o_t is the linear layer of the classifier of the tth model, o_{t-1} is the linear layer of the classifier of the $(t-1)$th model, o_{new} is the newly added nodes of the linear layer.

The training batches consisting of newly added data and selected exemplars will be balanced before it is used to train the new model. In the incremental recognition stage, the training set will be respectively inputted into old models and the new model to produce soft targets $\overline{y}$ and predictions $\hat{y}$. The generation of soft targets is as follows:

$$\overline{y}_{i,j,t} = \Phi_t(x_{i,j}), t = 1, \ldots, M, \tag{11}$$

where $x_{i,j}$ is the sample i of class j in the training set, M is the number of previous models in $\mathcal{M}$. The loss function consists of

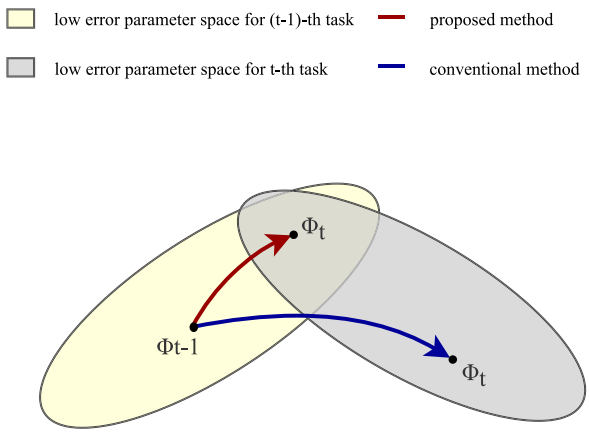

Fig. 3. Geometric illustration of the model change of the proposed method and conventional method in parameter space. Φ_{t-1} is the model parameter trained on the previous data. Φ_t is the model parameter trained on the current data. The red arrow line indicates the proposed method. The blue arrow line indicates the conventional method.

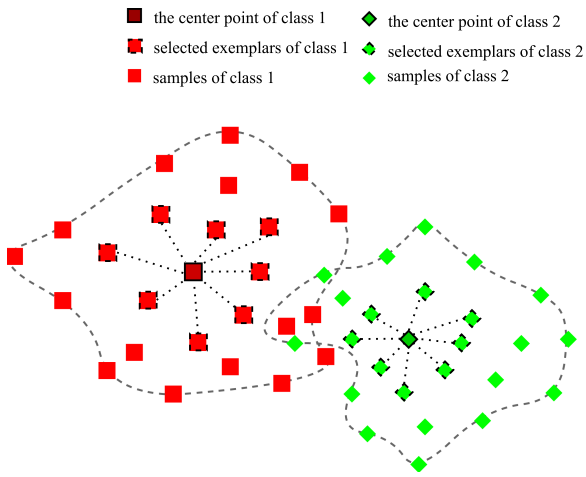

Fig. 4. Illustration for exemplar selection in feature space.

two parts, one is classification loss and the other is distillation loss. The calculation of the classification loss is the same as that in the first recognition task, and the distillation loss calculates the error between predictions and soft targets. Then, the optimizer updates the model parameters. The model and exemplars will be stored like the first recognition task. Finally, the total number of exemplars can not exceed a fixed parameter K. The subsequent incremental recognition task will be processed by the above workflow. The illustration of the model change in the parameter space is shown in Fig. 3.

4) Other Details: In the feature extractor θ, biases are not for conv layers. The parameters that should be initialized include the weights of conv layers w_{conv}, the weights of BN layers γ, and the biases of BN layers β. The parameters of BN layers are initialized by constants, i.e., γ is set to 1, and β is set to 0. The weights w_{conv} follow a normal distribution with a mean of 0 and a standard deviation of $\text{std}_{w_{\text{conv}}}$

$$w_{\text{conv}} \sim N(0, \text{std}_{w_{\text{conv}}}). \tag{12}$$

The standard deviation $\text{std}_{w_{\text{conv}}}$ is related to the selected activation function and fan mode. In our model, all activation functions after conv layers are rectified linear units (ReLU). Therefore, the formula of $\text{std}_{w_{\text{conv}}}$ is shown as follows:

$$std_{w_{\text{conv}}} = \sqrt{\frac{2}{\text{fan}_{\text{out}}}}. \tag{13}$$

There are two fan modes, including fan-in and fan-out. Both modes calculate the size of the receptive field by filter width W and height H. The difference is that the fan-in mode multiplies the number of input channels of the filter by the size of the receptive field to calculate the value, and the fan-out mode uses the number of output channels. The detailed formula is shown as follows:

$$\text{fan}_{\text{in}} = C_{\text{in}} \times W \times H \tag{14}$$

$$\text{fan}_{\text{out}} = C_{\text{out}} \times W \times H. \tag{15}$$

In the classifier o, the weights of linear layer w_{fc}, and the biases of linear layer b follow a uniform distribution. The boundary value of w_{fc} and the boundary value of b are both denoted by bound. The formulas are shown as follows:

$$w_{fc} \sim U(-\text{bound}, \text{bound}) \tag{16}$$

$$b \sim U(-\text{bound}, \text{bound}) \tag{17}$$

$$\text{bound} = \sqrt{\frac{1}{\text{fan}_{\text{in}}}}. \tag{18}$$

B. Main Components

The key parts in the proposed method, i.e., exemplars management, knowledge inheritance, data distribution balance, and pruning initialization, are detailed presented as following.

1) Exemplar Management: Catastrophic forgetting makes the model unusable in continual growth data. To address this problem, an intuitive method is selecting a few key samples from each class as exemplars. In [41], each class of samples is randomly selected with a fixed number to form an exemplar set. iCaRL [21] exploits a prioritized exemplar selection based on herding [42] and a hyperparameter for the total number of exemplars is used to replace the fixed number per class. After each training epoch, the mean of samples per class is calculated as the center and the exemplar is selected by the distance from the center as shown in Fig. 4. Exemplars that are further from the center should be removed when the new class is added.

Most data replay-based incremental learning methods [22], [43], [44] follow the iCaRL experiment benchmark protocol to arrange classes and select exemplars. In ScaIL [45], the experimental results show that exemplar selection based on herding can improve performance.

The exemplar management includes two steps, i.e., exemplar selection and reduction. After the model parameter update is completed, the algorithm selects a limited number of exemplars from the training set. The total number of exemplars that can be stored is set to K and C classes have been observed, $m = K/C$ exemplars will be stored for each class. In order to keep exemplar size constant, other exemplars will be removed.

The exemplar management algorithm selects data that are closer to the average value of feature vectors of a certain class in Euclidean metric space as exemplars of the class. For example, there is an image set $X^y = \{x_1, \ldots, x_n\}$ of class y, and the current feature extractor is $\theta : \chi \to \mathbb{R}^d$. The mean value μ^y of features of class y is calculated as follows:

$$\mu^y = \frac{1}{n} \sum_{x \in X^y} \theta(x). \tag{19}$$

If there are m exemplars that should be selected to construct an exemplar set $E^y = \{e_1, \ldots, e_m\}$ of class y, the operation is shown as follows:

$$e_q = \arg\min \left\| \mu^y - \frac{1}{q} \left[\theta(x) + \sum_{p=1}^{q-1} \theta(e_p) \right] \right\|, q = 1, \ldots, m, \tag{20}$$

where e_q is the qth selected exemplar of class y, μ^y is the mean value of features of class y. θ is the feature extractor and x is a sample of class y. As the new data continuously emerge, m becomes smaller. A few exemplars should be removed from the exemplar set. In the exemplar set E^y, the sequential position of exemplars represents the approximation degree. Namely, the earlier exemplars in E^y are more approximate to the mean value than the latter ones. The procedure for exemplar reduction removes the latter exemplars to keep only m exemplars in the exemplar set E^y.

2) Knowledge Inheritance: In addition to retraining with the exemplars, the old model implicitly involving the knowledge acquired from old data can be used to guide the training of the new models [46]–[48]. The knowledge distillation is originally proposed for network compression. Hinton *et al.* [46] transfer the dark knowledge in the cumbersome model to a tiny model by using posterior probability produced by the cumbersome model as soft targets. In [47], the inputs of the final softmax called logits are used to train a student model to mimic the deep neural networks via L2 regression. Compared with the hard targets, the soft targets and logits have much of the information in nodes with small value.

The soft targets blur the decision boundary and make training easy with limited training data. Inspired by this work, some incremental learning methods transfer knowledge from the old model to the current model based on knowledge distillation. In LwF [30], the soft targets generated by the old model and new data implicitly contain the knowledge in the old model. The joint loss function of LwF is composed of knowledge distillation loss function and cross-entropy loss function.

The distillation loss L_D is given by

$$L_D = -\frac{1}{N \times C} \sum_{i=1}^{N} \sum_{j=1}^{C} (\overline{y}_{i,j}^{\frac{1}{T}} - \hat{y}_{i,j}^{\frac{1}{T}})^2, \tag{21}$$

where $\overline{y}_{i,j}$ is the soft target generated by the old model for the sample i of class j and $\hat{y}_{i,j}$ is the output of the current model for the sample i of class j, N is the number of samples, C is the number of classes, T is the temperature scaling parameter.

The experiment of the effect of distillation in IL2M [23] shows that distillation loss is beneficial when no exemplar is supported. The use of distillation loss is detrimental if exemplars per old class are allowed. A similar result is presented in ScaIL [45], and more experiments about the effect of distillation are analyzed. The authors made a hypothesis that the detriment is caused during the incremental learning process, and these errors are caused by class imbalance. In the supplementary materials, the hypothesis is verified.

Like some other incremental learning methods for neural networks, our method also uses classification loss to learn knowledge from newly added data and combines the distillation loss to retain the knowledge from old classes. However, there are three differences in our method: 1) using the sigmoid as activation function instead of the softmax; 2) using mean square error (MSE) as classification loss instead of cross-entropy loss; and 3) using multiple old models instead of just the previous one. The softmax makes the output nodes sum to be 1. Nevertheless, the soft target is formed by concatenating the part of output nodes of multiple old models. The outputs of each model are activated by a softmax function, respectively. This operation results in the soft target neither correctly representing the probabilities nor correctly indicating the class of the sample. The other reason is that the softmax makes the output nodes influence each other, which is not conducive to knowledge preservation. The sigmoid independently processes each node of the output. The one-hot encoding makes the only nonzero contribution to the cross-entropy loss come from the truth node, while this issue does not happen in MSE loss. In [49], the MSE loss achieves a better result in class-incremental recognition. The reason for using multiple old models is that the classification accuracy of the model is always the best when the data are the most. If only the last model is used, the error will become larger as it accumulates. The comparison of single-model distillation and multimodel distillation is presented in Fig. 5. The combined loss function is defined as follows:

$$L(\Phi) = L_C(\Phi) + \lambda \sum_{t=1}^{M} L_{D_t}(\Phi) \tag{22}$$

where $L_C(\Phi)$ is the MSE loss for the stored exemplars from old classes and samples from new classes, $L_{D_t}(\Phi)$ is the distillation loss of the tth model Φ_t, M is the number of previous models in $\mathcal{M}$, and λ is a weighted parameter for distillation loss.

The MSE loss $L_C(\Phi)$ is defined as

$$L_C(\Phi) = -\frac{1}{N \times C} \sum_{i=1}^{N} \sum_{j=1}^{C} (y_{i,j} - \hat{y}_{i,j})^2 \tag{23}$$

where $\hat{y}_{i,j}$ is the output of the current model for sample i of class j, $y_{i,j}$ is the ground truth for sample i of class j, and N and C denote the number of samples and classes, respectively.

The distillation loss $L_{D_t}(\Phi)$ is given by

$$L_{D_t}(\Phi) = -\frac{1}{N_{\text{new}} \times C_{\text{new}}} \sum_{i=1}^{N_{\text{new}}} \sum_{j=1}^{C_{\text{new}}} (\overline{y}_{i,j}^{\frac{1}{T}} - \hat{y}_{i,j}^{\frac{1}{T}})^2 \tag{24}$$

where $\overline{y}_{i,j}$ is the soft target generated by the tth model for the sample i of class j and $\hat{y}_{i,j}$ is the output of the current model for the sample i of class j. C_{new} and N_{new} are the number of newly

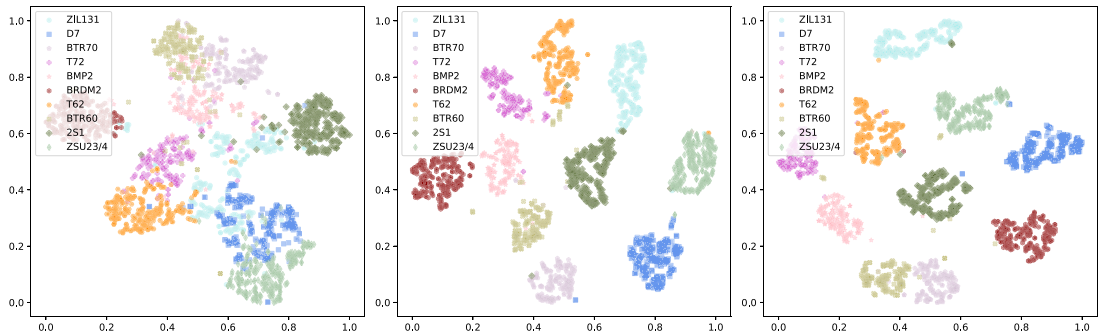

Fig. 5. Visualize the features of the MSTAR dataset using different incremental training. From left to right are results of fine tuning, single-model distillation, and multimodel distillation.

added classes and the sample number of newly added classes in the tth task, respectively. T is the temperature scaling parameter. T can adjust the influence of the smaller values in soft targets on the model update. In the high temperature, the distillation pays more attention to matching soft targets, and the smaller values in soft targets have greater influence. Like [21], [30], T is set to 2 in our experiments.

3) Data Distribution Balance: The sample amount of newly added classes is much greater than the amount of the stored exemplars of old classes. The newly added data are selected more frequently to form batches for the model training. This phenomenon is often referred to as the class imbalance problem. Since the model training uses the backpropagation algorithm, the data class imbalance will cause the model to be biased. Ultimately, the model can only discriminate the newly added classes very well, but the performance of the old classes degrades.

To cope with the problem, the sampling probabilities for each class data before incremental training will be weighted by the data amount. The probability of one class being selected is proportional to the data amount of that class in the case of random sampling. There are a set of sampling probability weights to ensure that the sampling probabilities for each class are the same. The class y sampling probability weight σ_y is defined as

$$\sigma_y = \frac{\sum_{j=1}^{C} N_j}{C \times N_y} \tag{25}$$

where C is the number of known classes, N_j and N_y are the sample number of the class j and class y respectively.

The random sampling probabilities are multiplied by the weights to obtain class-balanced sampling probabilities. After the weight operation, although the data diversity of old classes cannot be changed, the classes in each batch are balanced. The data distribution balance suppresses the model parameter bias caused by the class imbalance when the model is updated.

4) Pruning Initialization: The conventional method initializes the current model parameters by the last model parameters directly. However, the model parameters are usually redundancy. The overfull parameters make the model perform well on the training dataset, but when the data distribution changes, the model performance drops sharply. In the incremental recognition scenario, it is common that the data distribution changes

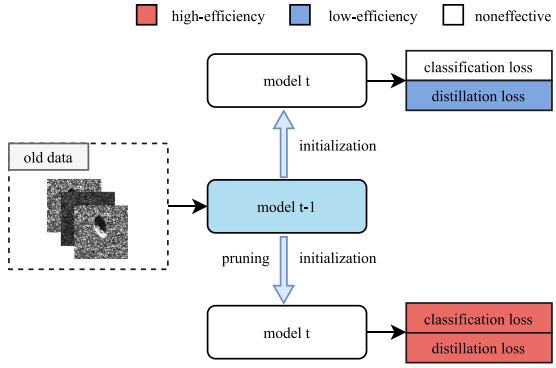

Fig. 6. Difference between pruning initialization and non-pruning initialization.

from old classes to new classes. To fit newly added data distribution efficiently, unstructured pruning initialization is used to initialize the current model

The data distribution balance component smoothens the data distribution change, the newly added data only account for a small part of the training batches. As shown in Fig. 6, the model is completely updated in the $(t-1)$th recognition task. Both the classification loss and distillation loss are tiny. If the $(t-1)-th$ model is used to initialize the tth model directly, the old data make almost no contributions to the model update. Pruning initialization could give a momentum boost to the model update. It means that the initialization without pruning makes the model plasticity worse and the initialization with pruning improves the update effectiveness of the model. Both old data and newly added data act on the model update to improve efficiency. Thus, model pruning is adopted to improve the model plasticity.

Before the current model is initialized, the importance of the model parameters is evaluated. Neurons with large absolute values play a major role in model prediction. Using unstructured pruning can remove neurons with small absolute values and preserve the model structure. The pruning operation is shown as follows:

$$w_l = \begin{cases} w_l & |w_l| \geq T_l \\ 0 & |w_l| < T_l \end{cases} \tag{26}$$

Fig. 7. Optical images (Top) and SAR images (Bottom) of MSTAR dataset. The class order is: ZIL131, D7, BTR70, T72, BMP2, BRDM2, T62, BTR60, 2S1, ZSU23/4.

where $|w_l|$ is the absolute value of each neuron in layer l, T_l is the pruning threshold of layer l. The neurons in each layer are sorted by the absolute value. Then, the threshold T_l of layer l is calculated by the pruning ratio. The pruning ratio is set to 20%, that is, neurons with the smaller 20% absolute values will be removed.

IV. Experimental Evaluation

In this section, the experiments are carried out to evaluate the effectiveness of the proposed method. The datasets moving and stationary target acquisition and recognition (MSTAR) and OpenSARShip [50] are used for experimental evaluation.

1) Dataset: The MSTAR dataset, that supported by the defense advanced research project agency (DARPA) and the air force research laboratory (AFRL), collects high-resolution spotlight-SAR imaging of ten classes of military vehicles, i.e., tank: T-72 and T-62; armored personal carrier: BMP-2, BRDM-2, BTR-60, and BTR-70; rocket launcher: 2S1; air defense unit: ZSU-234; truck: ZIL-131; bulldozer: D7. SAR images and optical images of 10 class vehicles in the MSTAR dataset are shown in Fig. 7. Most works for ATR use the MSTAR dataset to develop and evaluate their methods. The images in the MSTAR dataset are obtained by X-band SAR radar with HH polarization. The spatial resolution of these images is 1 ft $\times$ 1 ft. The dataset captures images continuously from 0° to 360° in azimuth with an interval of 5° or 6 ° at two depression angles 15° and 17°. In the experiments, images with the depression angle of 17° and images with the depression angle of 15° are used as the training set and the test set, respectively, whose configuration of the dataset is shown in Table I. The original image size ranges from 128 to 193. The larger size image contains more scenes of background clutter, and the target size is almost the same. Thus, the images are cropped to 88 $\times$ 88 to unify the size.

The OpenSARShip dataset comes from the Sentinel-1 satellite, which contains many medium-resolution ship chips under VV and VH polarization modes. Although the number of ship categories is large, the data are extremely unbalanced. As shown in Fig. 8, three types of ships under the VV polarization mode are selected to evaluate the proposed method. Each type has 300 samples. After dividing the data into 80% training and 20% test data, the final split results are shown in Table II. The images use the same preprocessing method as the MSTAR dataset.

The dataset is manually divided into multiple tasks for testing the incremental recognition capability of the proposed method and other compared methods. The incremental recognition scenario setup is similar to [17] and is shown in Table III. Three incremental recognition scenarios are designed, respectively, for which the training set is split into tasks with different increment numbers of classes. Each task does not include classes observed in other tasks.

TABLE I
Configuration of MSTAR Dataset

Target Class	Number of Images (17°)	Number of Images (15°)	Labels
ZIL131	299	274	0
D7	299	274	1
BTR70	233	196	2
T72	232	196	3
BMP2	233	195	4
BRDM2	298	274	5
T62	299	273	6
BTR60	256	195	7
2S1	299	274	8
ZSU23/4	299	274	9

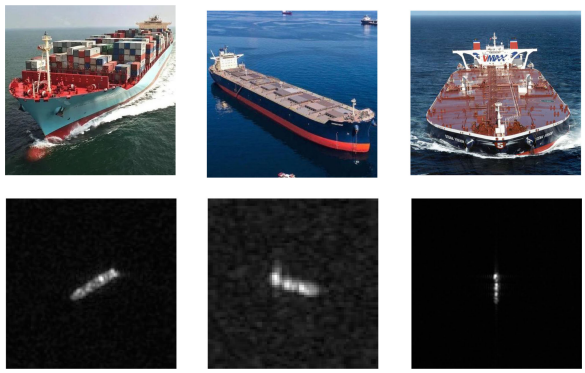

Fig. 8. Optical images (Top) and SAR images (Bottom) of selected data from OpenSARShip dataset. The class order is: Container Ship, Bulk Carrier, Tanker.

TABLE II
Configuration of OpenSARShip Dataset

Target Class	Number of training data	Number of test data	Labels
Container Ship	240	60	10
Bulk Carrier	240	60	11
Tanker	240	60	12

TABLE III
INCREMENTAL RECOGNITION SCENARIOS

Scenario \ Class	ZIL131	D7	BTR70	T72	BMP2	BRDM2	T62	BTR60	2S1	ZSU23/4	Container Ship	Bulk Carrier	Tanker
Scenario 1	*TASK 1*		*TASK 2*	*TASK 3*	*TASK 4*	*TASK 5*	*TASK 6*	*TASK 7*	*TASK 8*	*TASK 9*	*TASK 10*	*TASK 11*	*TASK 12*
Scenario 2	*TASK 1*		*TASK 2*		*TASK 3*		*TASK 4*		*TASK 5*		*TASK 6*		*TASK 7*
Scenario 3	*TASK 1*						*TASK 2*				*TASK 3*		

B. Experimental Setup

1) Evaluation Metrics: Each method evaluated in this article is trained in the incremental recognition scenario. The average value of all class accuracies per incremental recognition, called average incremental accuracy, can be one of the evaluation metrics. This metric reflects the overall accuracy of newly added classes and old classes.

As known that the average incremental accuracy can only represent the overall accuracy of each task, but the changes in the accuracy of each class are also essential for evaluating the stability and plasticity of these methods. The stability and plasticity of these methods can be compared by observing the accuracy of old classes and newly added classes, respectively. Therefore, the accuracy of each class per incremental batch can be one of the metrics.

For the evaluation of incremental recognition methods, only accuracy metrics are not enough. One advantage of incremental learning is that there is no need to reuse old training data. Such that, the time consumption of the training phase should be considered an essential metric.

2) Implementation Details: The $PyTorch$ deep learning framework is used to implement the experiments, which use a personal computer with Intel Core i7-8700 CPU of 3.2 GHz and NVIDIA RTX 2070 on Windows 10 system. The maximal number of exemplars K is 200. Each training phase consists of 50 epochs. In our method, the stochastic gradient descent is applied as an optimizer, where the learning rate is 0.01, the momentum factor is 0.9, and the weight decay $= 0.0001$. The training batch size is 32, and the distillation loss weighted parameter $\lambda = 0.2$.

C. Incremental Recognition Performance

The experiment mainly studies the classification accuracy and time consumption of different methods under the class-incremental condition. The proposed HPecIL and comparison methods are described as following.

1) *Joint training* repeatedly uses all training datasets in each training phase. It can be regarded as the upper boundary of the average accuracy of all incremental recognition methods, and the time consumption in the training phase is also the longest.
2) *iCaRL* [21] is a classical incremental recognition method based on data replay, which uses distillation loss to transfer old knowledge to the current model.
3) *CBesIL* [17] saves the previous recognition capabilities in the form of the class boundary exemplars. It uses 1NN as the classifier.
4) *ECIL* is an incomplete version of the proposed method that only uses exemplar management and knowledge inheritance. Compared with *iCaRL*, the big difference is the use of multimodel distillation instead of single-model distillation.
5) *ECIL+* adds data distribution balance part on the basis of *ECIL*.
6) *HPecIL* is the complete proposed method, including exemplar management, knowledge inheritance, data distribution balance, and pruning initialization parts.

According to the experiment results, the effectiveness of each part in the proposed method can be proved and analyzed.

The above methods were evaluated on three incremental recognition scenarios, respectively, and the average incremental accuracies of these methods are shown in Fig. 9. It can be seen from Fig. 9 that in different incremental recognition scenarios, the performances of all methods degrade with the emergence of new tasks. Due to the sufficient training data, the recognition accuracy of joint learning decreases the least. After completing the training of the last task data, the accuracy of HPecIL is second only to the joint learning. Because the feature representation capacity of NMF is lower than that of CNN, the recognition performance of CBesIL declined sharply. The final performance of HPecIL in the three scenarios exceeds CBesIL by 4.19%, 4.30%, and 3.54%, respectively. Especially for the last three types of ship data from OpenSARShip, the complexity and diversity of the data make identification much more difficult than the MSTAR dataset. In all three scenarios, a total of 13 types of targets were identified. However, the greater the number of tasks, the lower the final accuracy.

Tables IV–VII give the detailed change of accuracy per class of iCaRL, ECIL, ECIL+, and HPecIL in incremental recognition scenario 2. According to the analysis of these results, the effect of parts in the proposed method can be verified.

As shown in Table IV, the highest recognition accuracy of a category is when it first appears, that is, when the data of that category are the most. However, with the addition of new data, the accuracy of old data is gradually lower than that of first recognition. In addition, the earlier the category is trained, the more the accuracy drops. After the final incremental training, the recognition accuracy of ZIL131 and BMP2 compared to those of the first training drop 12.04% and 12.8%. ECIL adds the knowledge inheritance part to iCaRL. It can be seen from Table V, the decline in the accuracy of ZIL131 and T72 is significantly alleviated, which proves the effectiveness of the knowledge inheritance part. Using multiple models instead of just the former one effectively avoids the accumulation of misclassification. Nevertheless, the accuracy of container ships

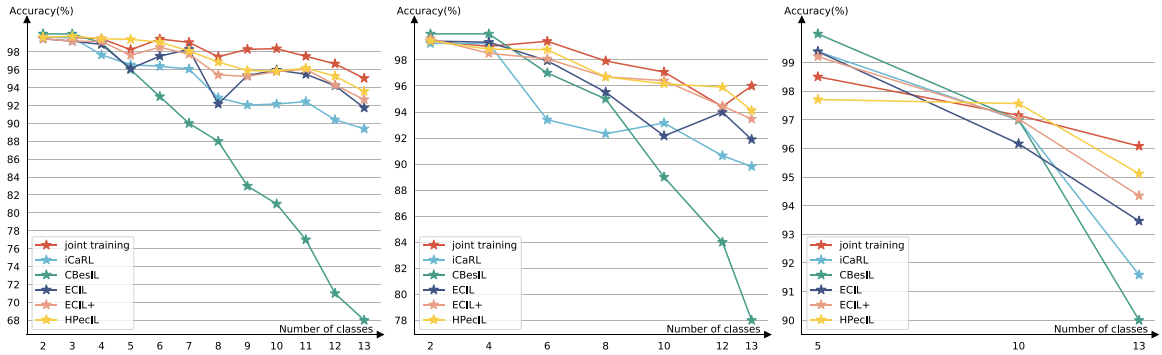

Fig. 9. Incremental recognition accuracies of scenario 1 (Left), 2 (Middle), and 3 (Right).

TABLE IV
DETAILED RECOGNITION ACCURACY OF ICARL IN INCREMENTAL RECOGNITION SCENARIO 2(%)

Training Set	ZIL131	D7	BTR70	T72	BMP2	BRDM2	T62	BTR60	2S1	ZSU23/4	Container Ship	Bulk Carrier	Tanker
Task 1	100.00	98.54	-	-	-	-	-	-	-	-	-	-	-
Task 2	98.91	98.91	100.00	99.49	-	-	-	-	-	-	-	-	-
Task 3	95.26	99.27	100.00	82.14	89.74	91.61	-	-	-	-	-	-	-
Task 4	91.24	99.27	97.45	85.20	85.13	95.26	91.21	90.77	-	-	-	-	-
Task 5	94.53	98.54	95.41	89.29	88.21	96.72	84.25	89.23	97.81	94.53	-	-	-
Task 6	87.59	98.18	97.45	88.78	80.51	96.35	91.21	85.13	98.18	89.42	76.27	61.02	-
Task 7	87.96	98.18	96.43	89.29	76.92	94.89	91.21	89.74	97.81	91.97	64.40	52.54	67.80

TABLE V
DETAILED RECOGNITION ACCURACY OF ECIL IN INCREMENTAL RECOGNITION SCENARIO 2(%)

Training Set	ZIL131	D7	BTR70	T72	BMP2	BRDM2	T62	BTR60	2S1	ZSU23/4	Container Ship	Bulk Carrier	Tanker
Task 1	100.00	98.91	-	-	-	-	-	-	-	-	-	-	-
Task 2	97.81	98.54	100.00	99.49	-	-	-	-	-	-	-	-	-
Task 3	99.64	98.18	98.47	98.47	96.41	98.91	-	-	-	-	-	-	-
Task 4	99.64	99.27	90.82	91.33	82.05	99.64	97.80	95.90	-	-	-	-	-
Task 5	91.61	98.18	96.94	94.90	85.13	98.91	79.85	88.72	99.64	100.00	-	-	-
Task 6	97.45	98.52	94.39	98.98	91.79	96.72	90.84	89.23	98.18	94.16	81.36	66.10	-
Task 7	95.62	97.81	95.92	94.90	85.13	99.27	91.58	89.74	98.91	96.35	11.86	1.69	100.00

TABLE VI
DETAILED RECOGNITION ACCURACY OF ECIL+ IN INCREMENTAL RECOGNITION SCENARIO 2(%)

Training Set	ZIL131	D7	BTR70	T72	BMP2	BRDM2	T62	BTR60	2S1	ZSU23/4	Container Ship	Bulk Carrier	Tanker
Task 1	100.00	99.27	-	-	-	-	-	-	-	-	-	-	-
Task 2	98.54	96.72	100.00	99.49	-	-	-	-	-	-	-	-	-
Task 3	99.64	97.81	100.00	94.90	95.90	99.27	-	-	-	-	-	-	-
Task 4	99.64	96.72	98.98	96.43	93.85	99.27	97.44	88.72	-	-	-	-	-
Task 5	95.62	97.45	97.45	95.41	92.82	98.91	95.60	91.28	100.00	97.08	-	-	-
Task 6	94.53	98.18	98.98	93.88	94.87	97.81	94.87	90.26	99.27	92.34	72.88	67.80	-
Task 7	96.35	98.91	97.45	96.43	92.82	98.91	95.24	89.74	98.91	95.99	55.93	77.97	28.81

TABLE VII
DETAILED RECOGNITION ACCURACY OF HPECIL IN INCREMENTAL RECOGNITION SCENARIO 2(%)

Training Set	ZIL131	D7	BTR70	T72	BMP2	BRDM2	T62	BTR60	2S1	ZSU23/4	Container Ship	Bulk Carrier	Tanker
Task 1	100.00	98.91	-	-	-	-	-	-	-	-	-	-	-
Task 2	99.64	98.18	100.00	97.45	-	-	-	-	-	-	-	-	-
Task 3	99.64	98.54	99.49	98.98	98.98	97.45	-	-	-	-	-	-	-
Task 4	99.64	98.54	100.00	94.90	94.87	100.00	95.97	86.67	-	-	-	-	-
Task 5	96.72	98.18	99.49	97.96	97.44	98.18	89.74	90.77	98.54	94.89	-	-	-
Task 6	97.81	98.91	98.47	98.47	95.38	97.45	96.33	91.28	98.91	96.35	84.75	59.32	-
Task 7	97.08	99.27	98.47	98.47	93.33	98.91	94.87	91.79	99.27	94.16	64.41	55.93	55.93

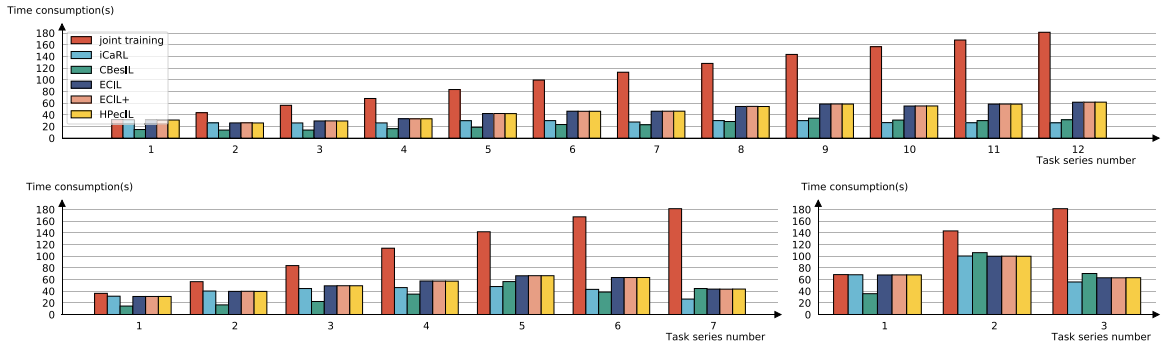

Fig. 10. Time consumption of scenario 1 (Top), 2 (Bottom left), and 3 (Bottom right).

and bulk carriers drops abruptly. This is most likely caused by data imbalance. The MSTAR data is not significantly affected due to the high resolution and little change in the depression angles. Even with the same class in the OpenSARShip, the targets are diverse. Furthermore, the size of targets in MSTAR data is similar, but that in the OpenSARShip differs greatly. sharply data distribution shift makes multiple models contradict each other, which harms the recognition accuracy. Table VI lists the detailed change of accuracy per class of ECIL+. Compared with ECIL, ECIL+ uses the class-balanced training batches. In Tables IV and V, the recognition accuracy of some categories (e.g., ZIL131, T72, and BMP2) fluctuates greatly during the incremental training phase. The recognition accuracy suddenly drops when it is high and then slowly rises back. After adding the data distribution balance part, the recognition accuracy of these categories keeps a smoothy change. Moreover, the unsatisfactory performance on the OpenSARShip data in Table V has been also improved.

The complete proposed method (i.e., HPecIL) adds the pruning initialization part to ECIL+, and the recognition accuracy of HPecIL is listed in Table VII. Compared with the recognition accuracy of ECIL+ in Table VI, the overall accuracy is slightly improved and the accuracy changes more smoothly during incremental training. The pruning initialization part gives a momentum boost to the optimizer when the model is trained on the mixed data. The efficient use of old data accelerates the convergence of model parameters, which improves the plasticity of the model.

Fig. 10 illustrates the time consumption of the different methods. As new data emerge, the training data will accumulate. If the network structure is the same, the time consumption in the training phase is positively correlated with the amount of training data. Joint training applies all training data in the training phase, so it consumes the most time. CBesIL consumes the least time in the beginning because its computational complexity is lower compared with deep learning methods. However, due to the need to synthesize old data, the time consumption of CBesIL will exceed that of iCaRL as recognition categories increases. As shown in Fig. 10, the time consumption of CBesIL in task 7 of scenario 2 even exceeds ECIL, ECIL+, and HPecIL. The same result also occurs in tasks 2 and 3 in scenario 3. After much incremental training, iCaRL is the shortest time-consuming. ECIL, ECIL+, and HPecIL always have similar time consumption. An interesting issue is that the amount of data of task 12 in scenario 1 and task 7 in scenario 2 is the same (i.e., 200 exemplars and 240 newly added samples), but the time consumption of the former is about 60 s and that of the latter is about 40 s. The reason is that the former requires five more models to generate soft targets than the latter. In general, the incremental learning method takes much less time than joint training.

V. Conclusion

In this article, we present a novel approach based on neural networks for class-incremental SAR ATR. The proposed method maintains outstanding recognition accuracy and efficient training in incremental recognition tasks. This work can be extended to be a scheme for other remote sensing class-incremental ATR problems. The improvements in the proposed method are given as follows:

1) The CNNs-based recognition framework incrementally learn to recognize target without reusing all old data in the training phase. Therefore, the training time consumption is reduced and storage pressure is alleviated.
2) The knowledge inheritance can exploit all old models. The soft targets of old classes are generated by local optimal models, i.e., the model obtained when the training data of the classes is the most.
3) Data distribution balance and pruning initialization can suppress the model bias to old or new classes due to data distribution changes and model initialization. The experimental results demonstrate that the data distribution balance and pruning initialization improve the stability and plasticity of the method, respectively.

The experiments compare the proposed method with joint training, CBesIL, and iCaRL. On the one hand, the results show that the incremental recognition methods that do not use all the training data of the old classes have excellent performance on the training time consumption. On the other hand, the accuracy of the incremental recognition methods can approach joint training by increasing the stability and plasticity of the model.

Note that the plasticity and stability of the method are usually in conflict. While strengthening one, it usually weakens the other. Therefore, a tradeoff should be made to achieve the best overall performance.

The future works of this article are listed as follows:

1) The exemplar management part is a lossy sampling of old training data. The unselected samples are not useless for recognition. In the proposed method, the original images are directly selected as exemplars, and the information contained in those unselected samples is lost. Therefore, how to use the least amount of data to retain the most information should be researched.
2) As mentioned earlier, the existing incremental recognition methods have to compromise accuracy, time consumption, and memory storage. The proposed method needs extra memory to store the old models. The relationship between old models should be drilled deep to compress and refine them.

REFERENCES

[1] A. Moreira, P. Prats-Iraola, M. Younis, G. Krieger, I. Hajnsek, and K. P. Papathanassiou, "A tutorial on synthetic aperture radar," *IEEE Geosci. Remote Sens. Mag.*, vol. 1, no. 1, pp. 6–43, Mar. 2013.

[2] K. El-Darymli, E. W. Gill, P. Mcguire, D. Power, and C. Moloney, "Automatic target recognition in synthetic aperture radar imagery: A state-of-the-art review," *IEEE Access*, vol. 4, pp. 6014–6058, 2016.

[3] L. M. Novak, G. J. Owirka, W. S. Brower, and A. L. Weaver, "The automatic target-recognition system in SAIP," *Lincoln Lab. J.*, vol. 10, no. 2, pp. 187–202, 1997.

[4] L. Novak, "State-of-the-art of SAR automatic target recognition," in *Proc. IEEE Int. Radar Conf.*, 2000, pp. 836–843.

[5] E. R. Keydel, S. W. Lee, and J. T. Moore, "MSTAR extended operating conditions: A tutorial," in *Proc. Int. Symp. Opt. Sci. Eng.*, 1996, pp. 228–242.

[6] L. C. Potter and R. L. Moses, "Attributed scattering centers for SAR ATR," *IEEE Trans. Image Process.*, vol. 6, no. 1, pp. 79–91, Jan. 1997.

[7] S. Song, B. Xu, and J. Yang, "SAR target recognition via supervised discriminative dictionary learning and sparse representation of the SAR-HOG feature," *Remote Sens.*, vol. 8, no. 8, 2016, Art. no. 683.

[8] F. Dellinger, J. Delon, Y. Gousseau, J. Michel, and F. Tupin, "SAR-SIFT: A SIFT-Like algorithm for SAR images," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 1, pp. 453–466, Jan. 2015.

[9] Q. Zhao and J. C. Principe, "Support vector machines for SAR automatic target recognition," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 37, no. 2, pp. 643–654, Apr. 2001.

[10] Y. Sun, Z. Liu, S. Todorovic, and J. Li, "Adaptive boosting for SAR automatic target recognition," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 43, no. 1, pp. 112–125, Jan. 2007.

[11] S. Chen, H. Wang, F. Xu, and Y.-Q. Jin, "Target classification using the deep convolutional networks for SAR images," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 8, pp. 4806–4817, Aug. 2016.

[12] Y. Li, L. Du, and D. Wei, "Multiscale CNN based on component analysis for SAR ATR," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–12, 2022, Art no. 5211212, doi: 10.1109/TGRS.2021.3100137.

[13] C. Wang *et al.*, "When deep learning meets multi-task learning in SAR ATR: Simultaneous target recognition and segmentation," *Remote Sens.*, vol. 12, no. 23, 2020, Art. no. 3863.

[14] Z. Lin, K. Ji, M. Kang, X. Leng, and H. Zou, "Deep convolutional highway unit network for SAR target classification with limited labeled training data," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 7, pp. 1091–1095, Jul. 2017.

[15] S. Du, J. Hong, Y. Wang, and Y. Qi, "A high-quality multicategory SAR images generation method with multiconstraint GAN for ATR," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2021, Art. no. 4011005, doi: 10.1109/LGRS.2021.3065682.

[16] L. Zhang *et al.*, "Domain knowledge powered two-stream deep network for few-shot SAR vehicle recognition," *IEEE Trans. Geosci. Remote Sens.*, to be published, doi: 10.1109/TGRS.2021.3116349.

[17] S. Dang, Z. Cao, Z. Cui, Y. Pi, and N. Liu, "Class boundary exemplar selection based incremental learning for automatic target recognition," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 8, pp. 5782–5792, Aug. 2020.

[18] J. L. McClelland, B. L. McNaughton, and R. C. O'Reilly, "Why there are complementary learning systems in the hippocampus and neocortex: Insights from the successes and failures of connectionist models of learning and memory," *Psychol. Rev.*, vol. 102, no. 3, 1995, Art. no. 419.

[19] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," *Psychol. Learn. Motivation*, vol. 24, pp. 109–165, 1989.

[20] M. Delange *et al.*, "A continual learning survey: Defying forgetting in classification tasks," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published, doi: 10.1109/TPAMI.2021.3057446.

[21] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "iCaRL: Incremental classifier and representation learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2001–2010.

[22] F. M. Castro, M. J. Marín-Jiménez, N. Guil, C. Schmid, and K. Alahari, "End-to-end incremental learning," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 233–248.

[23] E. Belouadah and A. Popescu, "IL2M: Class incremental learning with dual memory," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 583–592.

[24] H. Shin, J. K. Lee, J. Kim, and J. Kim, "Continual learning with deep generative replay," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 2990–2999.

[25] R. Kemker and C. Kanan, "FearNet: Brain-inspired model for incremental learning," 2017, *arXiv:1711.10563*.

[26] C. Wu *et al.*, "Memory replay GANs: Learning to generate new categories without forgetting," in *Proc. 32nd Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 5962–5972.

[27] C. He, R. Wang, S. Shan, and X. Chen, "Exemplar-supported generative reproduction for class incremental learning," in *Proc. Brit. Mach. Vis. Conf.*, 2018, pp. 98–110.

[28] J. Kirkpatrick *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proc. the Nat. Acad. Sci.*, vol. 114, no. 13, pp. 3521–3526, 2017.

[29] A. Chaudhry, P. K. Dokania, T. Ajanthan, and P. H. Torr, "Riemannian walk for incremental learning: Understanding forgetting and intransigence," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 532–547.

[30] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 2935–2947, Dec. 2018.

[31] J. Yoon, E. Yang, J. Lee, and S. J. Hwang, "Lifelong learning with dynamically expandable networks," 2017, *arXiv:1708.01547*.

[32] E. Belouadah and A. Popescu, "DeeSIL: Deep-shallow incremental learning," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 151–157.

[33] J. Rajasegaran, M. Hayat, S. H. Khan, F. S. Khan, and L. Shao, "Random path selection for continual learning," in *Proc. 33rd Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 12669–12679.

[34] R. Aljundi, P. Chakravarty, and T. Tuytelaars, "Expert gate: Lifelong learning with a network of experts," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3366–3375.

[35] A. Mallya and S. Lazebnik, "PackNet: Adding multiple tasks to a single network by iterative pruning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7765–7773.

[36] A. Mallya, D. Davis, and S. Lazebnik, "Piggyback: Adapting a single network to multiple tasks by learning to mask weights," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 67–82.

[37] S. Dang, Z. Cui, Z. Cao, and N. Liu, "SAR target recognition via incremental nonnegative matrix factorization," *Remote Sens.*, vol. 10, no. 3, 2018, Art. no. 374.

[38] C. Cao, Z. Cao, Z. Cui, and L. Wang, "Incremental robust non-negative matrix factorization for SAR image recognition," in *Proc. 6th Asia-Pacific Conf. Synthetic Aperture Radar*, 2019, pp. 1–5.

[39] S. Dang, Z. Cao, Z. Cui, Y. Pi, and N. Liu, "Open set incremental learning for automatic target recognition," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 7, pp. 4445–4456, Jul. 2019.

[40] X. Ma, K. Ji, L. Zhang, S. Feng, B. Xiong, and G. Kuang, "An open set recognition method for SAR targets based on multitask learning," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2021, Art. no. 4014005, doi: 10.1109/LGRS.2021.3079418.

[41] S. Hou, X. Pan, C. C. Loy, Z. Wang, and D. Lin, "Lifelong learning via progressive distillation and retrospection," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 437–452.

[42] M. Welling, "Herding dynamical weights to learn," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, 2009, pp. 1121–1128.

[43] Y. Wu *et al.*, "Large scale incremental learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 374–382.

[44] J. He, R. Mao, Z. Shao, and F. Zhu, "Incremental learning in online scenario," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 13926–13935.

[45] E. Belouadah and A. Popescu, "ScaIL: Classifier weights scaling for class incremental learning," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2020, pp. 1266–1275.

[46] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*.
[47] G. Urban *et al.*, "Do deep convolutional nets really need to be deep and convolutional," 2016, *arXiv:1603.05691*.
[48] J. Li, R. Zhao, J.-T. Huang, and Y. Gong, "Learning small-size DNN with output-distribution-based criteria," in *Proc. 15th Annu. Conf. Int. Speech Commun. Assoc.*, 2014, pp. 1910–1914.
[49] J. Zhang *et al.*, "Class-incremental learning via deep model consolidation," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2020, pp. 1131–1140.
[50] L. Huang *et al.*, "OpenSARShip: A dataset dedicated to Sentinel-1 ship interpretation," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 1, pp. 195–208, Jan. 2018.

Jiaxin Tang received the B.S. degrees in communication engineering from the Beijing University of Chemical Technology, Beijing, China, in 2017. He is currently working toward the Ph.D. degree in control science and engineering with the College of Information Science and Technology, Beijing University of Chemical Technology, Beijing, China.

His research interests include synthetic aperture radar (SAR) image processing, machine learning, automatic target recognition (ATR), and target detection.

Deliang Xiang (Member, IEEE) received the B.S. degree in remote sensing science and technology from Wuhan University, Wuhan, China, in 2010, the M.S. degree in photogrammetry and remote sensing from the National University of Defense Technology, Changsha, China, in 2012, and the Ph.D. degree in geoinformatics from the KTH Royal Institute of Technology, Stockholm, Sweden, in 2016.

In 2019, he was awarded a Humboldt Research Fellowship. Since 2020, he has been a Full Professor with Interdisciplinary Research Center for Artificial Intelligence, Beijing University of Chemical Technology, Beijing, China. His research interests include urban remote sensing, synthetic aperture radar (SAR)/polarimetric SAR image processing, artificial intelligence, and pattern recognition.

Dr. Xiang serves as a Reviewer for the Remote Sensing of Environment, the ISPRS Journal of Photogrammetry and Remote Sensing, the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, and several other international journals in the remote sensing field.

Fan Zhang (Senior Member, IEEE) received the B.E. degree in communication engineering from the Civil Aviation University of China, Tianjin, China, in 2002, the M.S. degree in signal and information processing from Beihang University, Beijing, China, in 2005, and the Ph.D. degree in signal and information processing from Institute of Electronics, Chinese Academy of Sciences, Beijing, China, in 2008.

He is currently a Full Professor in electronic and information engineering with the Beijing University of Chemical Technology, Beijing, China. His research interests are remote sensing image processing, high performance computing and artificial intelligence.

Dr. Zhang is an Associate Editor for the IEEE ACCESS and a reviewer of IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING, IEEE GEOSCIENCE AND REMOTE SENSING LETTERS and *Journal of Radars*.

Fei Ma (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electronic and information engineering from the Beihang University, Beijing, China, in 2013, 2016, and 2020 respectively.

From 2017 to 2018, he was a Research Fellow with Department of Electrical Engineering, McGill University, Montreal, Canada. Since 2020, he has been an Associate Professor with the College of Information Science and Technology, Beijing University of Chemical Technology, Beijing, China. His research interests include synthetic aperture radar (SAR) image processing, machine learning, artificial intelligence, and target detection.

Yongsheng Zhou (Member, IEEE) received the B.E. degree in communication engineering from the Beijing Information Science and Technology University, Beijing, China, in 2005, the Ph.D. degree in signal and information processing from Institute of Electronics, Chinese Academy of Sciences, Beijing, China, in 2010.

He is currently a Professor of electronic and information engineering with the Beijing University of Chemical Technology, Beijing, China. His research interests lie in target detection and recognition from microwave remotely sensed image, digital signal and image processing, etc.

HengChao Li (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees from Southwest Jiaotong University, Chengdu, China, in 2001 and 2004, respectively, and the Ph.D. degree from the Graduate University of Chinese Academy of Sciences, Beijing, China, in 2008, all in information and communication engineering.

He is currently a Full Professor with the School of Information Science and Technology, Southwest Jiaotong University. From November 2013 to October 2014, he was a Visiting Scholar working with Prof. William J. Emery with the University of Colorado Boulder, Boulder, CO, USA. He has authored more than 100 research papers, including 71 JCR journal papers, such as published in the *ISPRS Journal of Photogrammetry and Remote Sensing*, IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING, IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, and *Pattern Recognition*. His research interests include statistical analysis of SAR images, remote sensing image processing, and pattern recognition.

Prof. Li was the recipient of the 2018 Best Reviewer Award from IEEE Geoscience and Remote Sensing Society for his service to the IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING (JSTARS). He is an Associate Editor for the IEEE JSTARS, and an Editorial Board Member of the *Journal of Southwest Jiaotong University* and the *Journal of Radars*. In addition, he has was a Guest Editor of Special Issues of *Journal of Real-Time Image Processing*, IEEE JSTARS, and IEEE JOURNAL ON MINIATURIZATION FOR AIR AND SPACE SYSTEMS, a Program Committee Member for the 26th International Joint Conference on Artificial Intelligence (IJCAI-2107) and the 10th International Workshop on the Analysis of Multitemporal Remote Sensing Images (MULTITEMP-2019), and the Session Chair for the 2017 International Geoscience and Remote Sensing Symposium (IGARSS-2017) and the 2019 Asia-Pacific Conference on Synthetic Aperture Radar (APSAR-2019). He is also an Active Reviewer of more than 30 international journals.