remote sensing

MDPI

Article

A Class-Incremental Learning Method for SAR Images Based on Self-Sustainment Guidance Representation

Qidi Pan, Kuo Liao *, Xuesi He, Zhichun Bu and Jiyan Huang

School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China; 202122011338@std.uestc.edu.cn (Q.P.); huangjiyan@uestc.edu.cn (J.H.)
* Correspondence: liaokuo@uestc.edu.cn

Abstract: Existing deep learning algorithms for synthetic aperture radar (SAR) image recognition are performed with offline data. These methods must use all data to retrain the entire model when new data are added. However, facing the real application environment with growing data, retraining consumes much time and memory space. Class-Incremental Learning (CIL) addresses this problem that deep learning faces in streaming data. The goal of CIL is to enable the model to continuously learn new classes without using all data to retrain the model while maintaining the ability to recognize previous classes. Most of the CIL methods adopt a replay strategy to realize it. However, the number of retained samples is too small to carry enough information. The replay strategy is still trapped by forgetting previous knowledge. For this reason, we propose a CIL method for SAR images based on self-sustainment guidance representation. The method uses the vision transformer (ViT) structure as the basic framework. We add a dynamic query navigation module to enhance the model's ability to learn the new classes. This module stores special information about classes and uses it to guide the direction of feature extraction in subsequent model learning. In addition, the method also comprises a structural extension module to defend the forgetting of old classes when the model learns new knowledge. It is constructed to maintain the representation of the model in previous classes. The model will learn under the coordinated guidance of old and new information. Experiments on the Moving and Stationary Target Acquisition and Recognition (MSTAR) dataset show that our method performs well with remarkable advantages in CIL tasks. This method has a better accuracy rate and performance dropping rate than state-of-the-art methods under the same setting and maintains the ability of incremental learning with fewer replay samples. Additionally, experiments on a popular image dataset (CIFAR100) also demonstrate the scalability of our approach.

Keywords: class-incremental learning; SAR images recognition; vision transformer

Citation: Pan, Q.; Liao, K.; He, X.; Bu, Z.; Huang, J. A Class-Incremental Learning Method for SAR Images Based on Self-Sustainment Guidance Representation. *Remote Sens.* **2023**, *15*, 2631. https://doi.org/10.3390/rs15102631

Academic Editors: Bo Tang, Xinghua Li, Zongxu Pan, Fan Zhang, Zhongling Huang and Wei Yao

Received: 23 March 2023
Revised: 12 May 2023
Accepted: 16 May 2023
Published: 18 May 2023

1. Introduction

Synthetic Aperture Radar (SAR) [1] uses echo coherence processing to obtain high-resolution images. SAR images have been widely used in the fields of military target detection and civilian remote sensing. The research of automatic target recognition technology [2,3] based on two-dimensional SAR images with high resolution has important military and civilian values. With the development of Convolution Neural Networks (CNN), the recognition of SAR images using CNN [4,5] has been studied intensively. Gao et al. [6] propose a multi-feature fusion network and the weighted distance classifier to extract classification features and classify them. The neural network performs excellently in learning offline data. However, the neural network is less effective with streaming data for SAR images. When updating the model with new data, the data representation and decision boundary are changed. The model will rapidly forget what it has learned previously with the change of parameters, this is called catastrophic forgetting [7]. The most primitive method to deal with the forgetting problem is to use all data to retrain the whole model whenever a new task comes. However, retraining will waste a lot of time, and

the model will collapse when the data increases to unaffordable status. Furthermore, in some realistic situations, the samples from previous classes are often difficult to obtain at once. It is impossible for the model to retain all data.

Class-Incremental Learning (CIL) [8], also called Continue Learning (CL) [9], Life-Long Learning (LLL) [10], solves the problem of learning from streaming data and gives the possibility for the model to learn new tasks continually. The major challenge of CIL is that performance on previous tasks cannot significantly decrease with new tasks added. The aim is to find a better trade-off between the plasticity and stability of the model [11], where plasticity refers to the ability to integrate new information from new tasks and stability is to retain previous knowledge while learning it. Instead of retraining the entire model, CIL introduces various methods to make the model learn continually, such as the replay strategy [12–14], regularization-based methods [15,16] and parameter isolation method [17]. Previous works widely adopt the replay strategy: storing small samples of old classes and reusing them when the model learns new classes. For example, Rebuffi et al. [12] proposed a Herding-based preferential samples selection method (iCaRL). Wu et al. [18] optimized the classification bias in the extracted samples (BiC). However, this kind of method only reshows a few samples from old classes. Since the number of retained samples is less than those from new classes, the distribution of features in the old and new classes is unbalanced. The model prefers to learn new classes. As a result, it is still challenging to alleviate forgetting only under the replay strategy. Convinced by this fact, we highlight two major barriers to be addressed for CIL: (i) Besides repaying samples of old classes, how can we keep the knowledge alive in the future? (ii) Under the limited structure and storage space, how can we store more information to recreate a similar situation in the past?

Complementary Learning Systems (CLS) theory [19] researches the memory style of mammals. When recalling past scenes, mammals will construct a new insight under the guidance of the current scenario. For example, people's attitudes toward others change with experience. This memory pattern is gradually being used in incremental learning to reduce the forgetting of models. In Natural Language Processing (NLP), prompt-based learning (prompting) [20] is a new transfer learning technique developed from CLS theory. Prompting techniques design inputs with templated or learnable prompt tokens. These prompt tokens contain additional task information, allowing the language model to obtain more information from the input. Incremental methods from CLS theory and prompting techniques handle the CIL problems to a certain extent. For instance, in incremental learning, Wang, et al. [21] propose a trainable prompt pool module to store encoded knowledge (L2P). However, L2P is a method of freezing the feature extraction layer and only training the classifier [22–24]. When tasks are significantly different, the frozen feature extraction layer will extract similar features. Figure 1 shows that these similar features do not represent the characteristics of tasks in the feature space and negatively affect the result of recognition. These methods are just used for akin tasks.

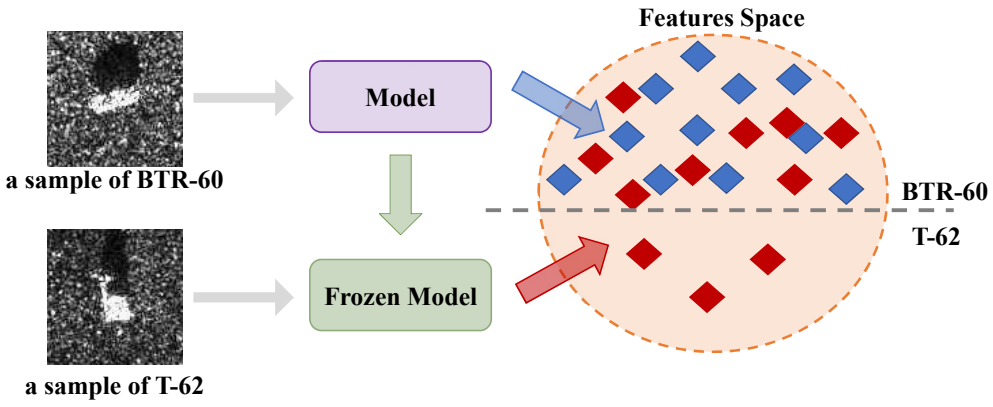

Figure 1. Demonstrates the negative impact of freezing feature extraction layer.

Motivated by the above discussions, we propose a Class-Incremental Learning method for SAR images based on self-sustainment guidance representation to maintain plasticity and stability simultaneously. In relation to plasticity, we use little storage space to store the remarkable features of different classes, and this information guides the direction of feature extraction when the model is learning. For example, it is known that some features are essential for classification, such as a pig's nose. When samples of the pig are given, the model focuses on the nose for classification under the guidance of the special features stored before. For stability, we use extra structure to construct a bridge between the past and the future to imitate how mammals remember. The new samples go through the extra structure which is trained by previous knowledge, and its output may be the features of new classes under the old situation. The features combine with current knowledge of new classes from basic structure to assist the model's learning.

Specifically, we built a base framework with a vision transformer (ViT) [25] for feature extraction and classification recognition. We designed a dynamic query navigation module to maintain plasticity. The module retains the special information of classes and adds this information to the input. The direction of the feature extraction layer is changed dynamically to the current class with the guidance of special information. In addition, we also propose a structural extension module to hold the stability. The extra structure is designed to keep the model's knowledge representation of old classes. Through the fusion of information and structural expansion of some encoding layers in the transformer encoder, the model learns under the guidance of both new and old knowledge. We introduce knowledge distillation to transfer the structural information from the extra model to the new model.

In summary, the contributions of this letter are as follows:

(1) We propose a learnable dynamic query navigation module to enhance the feature extraction ability of the model. The module learns and retains the special information of the class more conducive to achieving target recognition. Inputting this special information as auxiliary information can guide the focus of feature extraction of the model. The features acquired are more beneficial for classification recognition. The module can also apply to non-incremental learning models.
(2) We propose a structural extension module to address the catastrophic forgetting problem of the model. Considering the knowledge expressions related to the old classes, the module multi-dimensionally integrates the representation of knowledge in new and old classes. It achieves situational memory of past information rather than a simple recapitulation. The module achieves better results in preventing forgetting than the conventional replay method.
(3) We conducted comprehensive experiments from multiple perspectives to demonstrate the effectiveness of our method. Experiments on the MSTAR SAR dataset show that our method is significantly better than existing incremental learning methods. Experiments on the CIFAR100 dataset show that our method can also be used in a more general scenario.

2. Related Works

2.1. Class-Incremental Learning Methods

The goal of Class-Incremental Learning is to maintain the performance of the model in previous classes when the model learns new classes. Some works [16,26,27] deal with the forgetting problem without previous data. However, popular methods are based on the replay strategy with limited data. These works are mainly divided into two parts: the representation methods and classification learning methods.

2.1.1. Representation Methods

Current methods are based on three categories. *The distillation-based* methods [12–14,28–30] adopt knowledge distillation [31] to retain the knowledge of previous classes. iCaRL [12] introduces a strategy of sampling based on herding and adds distillation loss to the outputs.

Castro et al. [13] improve the iCaRL by which the feature extraction layer and classification can learn together. Zhu et al. [28] use a main-branch distillation scheme to maintain the distinction of the new model on previous features and a prototype selection mechanism to select samples similar to the old class for distillation. Gao et al. [29] introduce a new distillation method called relation knowledge distillation to transfer the structural relation of new data from the old model to the current model. The distillation methods depend on the relationship between the samples selected and the whole previous samples. Xu et al. [32] use knowledge distillation to classify the hyperspectral image.

Regularization-based methods [15,33–35] predict the importance of all parameters in the neural network. During the later tasks, the variation in essential parameters is penalized by designing the loss function. Kirkpatrick et al. [15] use the Laplace approximation to obtain the maximum posterior probability. However, with the increase in tasks, mistakes between the estimated values and true values accumulate. As a result, the model will definitely forget what it learned before. Zenke et al. [35] break the way WEC proposed, and they estimate the importance of parameters during the training. Aljundi et al. [33] use an unsupervised and online matter. Aljundi et al. [34] extend the settings on none-task.

Structure-based methods [17,36–40] learn the new tasks under the fixed parameters of the previous model. Mallya et al. [17] keep the important parameters and cut the redundant parameters for future tasks. Rajasegaran et al. [37] introduce a method that progressively chooses the best network to be the sub-net of the new class. Liu et al. [38] restrain the distance between samples from the same classes in feature space to alleviate forgetting. Besides these, some prevalent methods adopt the topology of classes. Tao et al. [39] use a neural gas network to construct the topology of classes, and it reduces forgetting by limiting the changing of topology on old classes. Tao et al. [36] improve TOPIC by giving room to change the topology of old classes. Wang et al. [40] construct the topology for each sample. However, these methods have many parameters, which adds burdens to learning the model, especially in the last incremental tasks.

Except for directly repaying the past samples to retrain the model, our method stores the special information of classes and uses the information to extend the input for guiding the direction of the feature extraction. Besides this, we construct a new structure to store the old knowledge, which is smaller than the whole model. The new structure is penalized by the knowledge distillation loss to store structure information of the old model.

2.1.2. Classification Learning Methods

Some methods [22–24] freeze the feature extraction layer to prevent changes in parameters, and they only train the classifier to fit the new tasks. However, the frozen feature extraction layer limits the ability of the model to learn new knowledge since the output of feature extraction prefers to show the old classes. Zhou et al. [22] measure the relationships between classes in the feature space and use the relationships between classes to guide the synthesis of new classifiers. Wu et al. [23] propose a two-phase training strategy that fine-tunes the additional network branches on the new data. Wang et al. [24] retain the old model and freeze its parameters, extending it with a trainable new feature extractor.

Unlike keeping the model, our method freezes the feature extraction layer to learn the new tasks for maintaining the plasticity of the model.

2.2. Incremental Learning Based on SAR Images Recognition

These methods [41–44] based on SAR images explore the incremental methods for reducing the forgetting of models. Tao et al. [41] design a loss function that consists of a reconstruction part and a classification part. The construction loss reconstructs the weight of the features, and the classification loss restrains the distribution of different classes. Zheng et al. [42] introduce class-balance loss to deal with the imbalance of new and old classes, and it uses a covariance pooling network to improve the ability to recognize features. Wang et al. [43] propose a method to assign higher weights to more important information when the parameters are updated. Wang et al. [44] use contextual information

from the past to the present to adjust the classification weight. In fact, studies on SAR images place more consideration on the characteristics of the samples, such as the grayscale value, scattering type, projection type, etc.

2.3. Transformer Technique

The transformer structure [45] was first proposed for NLP, such as translation and question answering. Dosovitskiy et al. [25] introduce ViT into the Computer Version tasks. At present, the transformer is widely used in classification, segmentation, and object detection. The biggest innovation of transformer is that it directly rejects the architecture of Recurrent Neural Network (RNN) and CNN, and fully utilizes the attention mechanism to have powerful semantic feature extraction ability. Our work uses the vision transformer as the main structure.

3. Method

In this section, we first describe the concept of Class-Incremental Learning (CIL). Afterward, we show the general structure and details of the method.

3.1. Problem Statement

Incremental learning is the process by which the model learns incrementally on a series of tasks $\{T_0, T_1, \ldots, T_T\}$. These tasks include data sets $\{D_0, D_1, \ldots, D_T\}$ and label sets $\{Y_0, Y_1, \ldots, Y_T\}$. In incremental learning, T_0 is called base task, $\{T_1, \ldots, T_T\}$ are called incremental tasks. Each incremental task $T_i, i = \{1, \ldots, T\}$ has the same number of new classes, and the classes do not intersect, i.e., $\{N_{Y_i} = N_{Y_j}, Y_i \cap Y_j = \varnothing | i, j \in \{1, 2, 3 \ldots T\}\}$. The goal of incremental learning is that the model learns new datasets D_t on the current task t while maintaining the ability to recognize previous datasets $\{D_0 \cup \ldots \cup D_{t-1}\}$.

3.2. Model Architecture

The whole structure of our model consists of three parts: a basic framework based on a vision transformer for feature extraction and classification recognition, a dynamic query navigation module for storing the special information of classes, and a structural extension module for maintaining the knowledge representation of the old classes. Figure 2 depicts the complete structure of the model. Algorithm 1 explains the process of the overall structure of our method.

When the model is learning task t, the training samples D_t of new classes are combined with selected samples S_{t-1} saved in the sample library:

$$TD_t = \{D_t, S_{t-1}\} \tag{1}$$

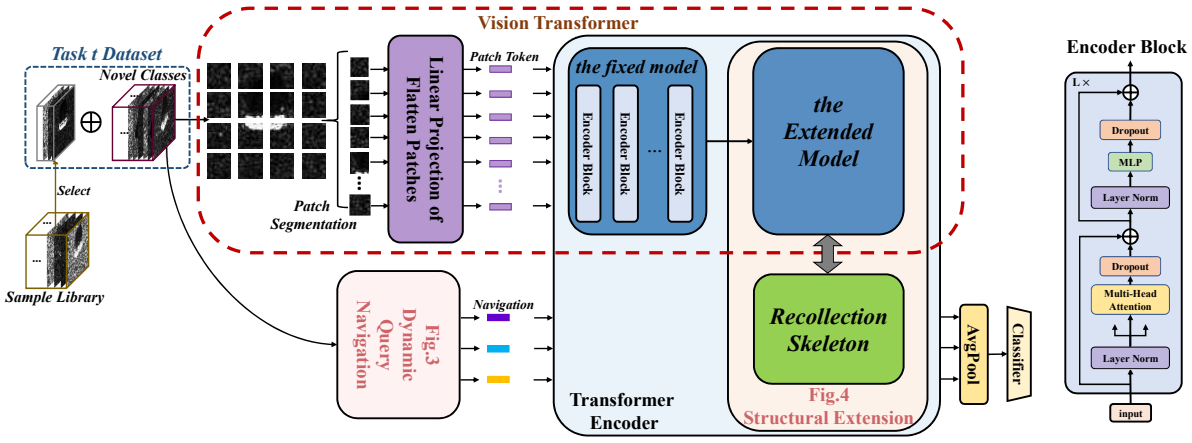

Figure 2. Demonstrates a Class-Incremental Learning method based on self-sustainment guidance representation on a standard ViT backbone.

Algorithm 1 The Overall Structure

Input: image x ; linear projection ϕ_{linear}; the dynamic query navigation module $M_{navi}(x)$; the structural extension module $M_{exten}(x)$
Hyper-parameters: Epoch number E
Output: Prediction output $\hat{y}$

for $t \leq E$ **do**
  Transfer x to 1-dimensional tokens: $X_p \leftarrow \phi_{linear}(x)$
  Transfer x to 1-dimensional navigations: $V_k \leftarrow M_{navi}(x)$
  Combine the two variations: $XV = X_p + V_k$
  Extract features from the fixed model: $Z_{fix} = \phi_{fix}(XV)$
  Extract features from the structural extension module: $Z^L = M_{exten}(Z_{fix})$
  Calculate the average at the output of navigations: $Z = \frac{1}{M}\sum_{k=1}^{M} Z_k$
  Input classifier to obtain prediction results
  $t + 1$
end for
return Result $\hat{y}$

Given an input of two-dimensional images $x \in R^{H\times W\times C}$ in TD_t, where $H \times W$ is the dimension of the images, C is the number of channels. In the first stage, the images go through two modules to obtain the input of the transformer encoder layer. On the one hand, through the linear projection layer of ViT, x is split and mapped as a series of 1-dimensional tokens $X_p = \{x_1, \ldots, x_{N_e}\} \in R^{N_e\times D}$, where N_e is the number of tokens, and D is the fixed dimension of the input of VIT. On the other hand, through the dynamic query navigation module, x is transferred to a handful of one-dimensional navigations $V_k = \{v_1, \ldots, v_M\} \in R^{M\times D}$, where M is the number of navigations and D means that the navigation has the same dimension as the token. The results of the above two variations on the image x are combined as $XV = \{x_1, \ldots, x_{N_e}, v_1, \ldots, v_M\}$ and fed into the transformer encoder for feature extraction.

In the second stage, the transformer encoder layer acting as the feature extraction layer in ViT extracts the features from XV. It is generally considered that the lower layers of the model can extract the basic characteristics of the target, which can be used for almost all categories with a strong generalization. Moreover, the higher layers extract the unique characteristics of the target which only be used for the single class. Therefore, as the transformer encoder consists of L encoder block layers, we divide the transformer encoder into two parts: the front $\frac{L}{2}$ layers form a fixed module for extracting general features, and the back $\frac{L}{2}$ layers form a structural extension module with the recollection skeleton.

The self-attention mechanism integrates information from all one-dimensional inputs. Every position of output contains information from other positions. Therefore, we use the output of navigations on the transformer encoder. The output is averaged and fed into a softmax classifier to obtain the recognition result of the model.

3.3. Dynamic Query Navigation Module

When people are identifying animals, they pay attention to the remarkable features of different animals, such as the red crown of the rooster, the pointed ear and striped pattern of a tabby, or the round ear and single-color pattern of the bear. However, some redundant features might negatively affect the result. For example, when identifying a rooster, both the red crown and the flattened feathers are characteristics of it. If we pay attention to all the characteristics, the incorrect answer is probably a bird with a red crown. Apparently, the marked features are what we need to pay more attention to when identifying.

The ideal model shares knowledge for similar classes while keeping knowledge independent for different classes. Like the striped pattern and the round ear can describe a tiger, these features also belong to a tabby and a bear. As a result, The shared information assists the learning of tasks similar to the past and alleviates the stress of storage.

Therefore, inspired by reference [21], we propose the learnable navigation set $V = \{v_1, v_2, \ldots, v_N\}, v_i \in R^{1\times D}$ to store the remarkable information of different classes, where N is the number of navigations. The set is shared with all tasks. We use the information to guide the direction of feature extraction by expanding the input of the transformer encoder with the navigation. For this reason, the navigation and the input of the transformer encoder should have the same dimension D.

Furthermore, we propose a learnable key set $K = \{k_1, k_2, \ldots, k_N\}, k_i \in R^{1\times D_k}$ to choose the remarkable information which is the sample needed, where N is the number of key vectors, D_k is the length of k_i. The key set is the bridge between the navigation and the category of the sample. On the one hand, each key and navigation are paired for finding navigations by keys. That means the number of keys and the number of navigations is the same, denoted as $\{(k_1, v_1), (k_2, v_2), \ldots, (k_N, v_N)\}$. On the other hand, in order to obtain the probable category of the sample, we introduce a backbone vision transformer model to pre-classify it. The backbone model uses the basic structure from the reference [25], and the parameters were pre-trained on the Google private dataset (JFT-300M). By computing the similarities between keys and the query features $f_x \in R^{1\times D_k}$ from the backbone model, we can choose the special information the sample needs. As a result, the key and the query features should have the same dimension D_k.

As shown in Figure 3, the whole dynamic query navigation module consists of a navigation set, a key set and a backbone vision transformer model. Algorithm 2 explains the training and testing process of this module.

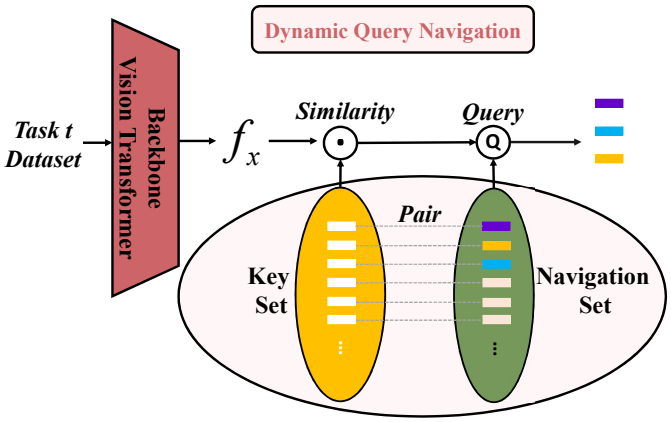

Figure 3. The details of the dynamic query navigation module. It learns and retains the special information of classes more conducive to achieving target recognition.

Algorithm 2 The process of the dynamic query navigation module

Input: image x ; backbone ViT model ϕ_{back}
Output: Chosen navigations V_k
1: Pre-classify the image by backbone ViT model: $f_x = \phi_{back}$
2: Calculate the similarity $Cori$ between key and f_x by Equation (2)
3: Choose indexes d_k of the first M maximum similarities
4: Choose the correspond navigations V_k from index d_k
5: **return** V_k

When training the model, keys and navigations are trained to obtain the information of the current samples. When testing, the current sample is put into the backbone model to obtain the query features f_x of their potential class. We then calculate the similarity between query features q_x and all keys in the key set:

$$Cori = Cor(f_x, k_i) = \| f_x \|_2 \cdot \| mean\,(k_i) \|_2, i = 1, 2, \ldots, N \quad (2)$$

where $\|\cdot\|_2$ is the $L2$ parametrization.

The indexes of the key of the first M maximum similarities are taken and according to the correspondence between keys and navigation, the first M navigations V_k are queried. V_k is the remarkable information of classes we wish to obtain for identification.

3.4. Structural Extension

The CLS theory in neuroscience suggests that mammals have two learning systems: the neocortex and the hippocampus. The hippocampus rapidly replays recently learned memories, integrating them as interconnections of neocortical neurons and becoming a more solid memory. Based on this, we construct an extra structure called the recollection skeleton as the neocortex to store the solid memory of the past. The feature extraction layer acts as the hippocampus for learning new knowledge. In addition, the research also shows that the hippocampus does not directly recall past events when replaying memories, but constructs a new insight called situational memory, according to a realistic scenario. For machine learning, we consider the output of the new data on the old model as the memory associated with the old class, equivalent to the situational memory of the hippocampus. Since we have constructed the recollection skeleton to store parameters of the old model, the new data is put through it to recall past knowledge.

Therefore, we propose a structural extension module to imitate the learning style of the mammalian. The module consists of an extended model and a recollection skeleton structure.

As we explained in Section 3.2, the lower layers of the model extract the basic characteristics and the higher layers extract the unique ones. Thus, the special information of classes is extracted from the higher layers of the transformer encoder module in ViT. For most of the classes, the features extracted at the lower levels are similar. The parameters of lower layers might not change dramatically when new data are added. Therefore, we froze the front $\frac{L}{2}$ layers of the transformer encoder to extract the basic characteristics, and we use the back $\frac{L}{2}$ layers to construct the extended model.

The recollection skeleton acts as a bridge between the past and the current. It is used to store past knowledge. When new data is put through it, the old information in the structure converts to situational memory. In order to guide the updating of the extended model with this situational memory, we propose a new embedding method. In this method, we first design the recollection skeleton to have the same structure as the extended model, and both of them consist of $\frac{L}{2}$ encoder blocks. We then set our sights on the layers to realize information interaction. As shown in Figure 4, every encoder block layer in the recollection skeleton embeds into the corresponding layer in the extended model. Algorithm 3 explains the process of this module.

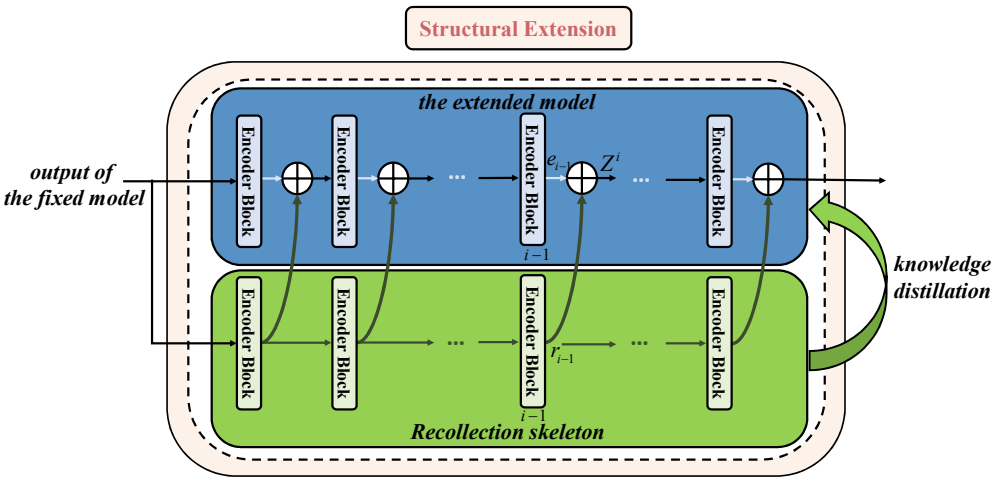

Figure 4. The details of the structural extension module. It keeps the model's knowledge representation of old classes.

Algorithm 3 The process of the structural extension module

Input: the output of the fixed model Z_{fix}
Output: the output of the transformer encoder layer Z^L
1: **for** $\frac{L}{2} + 2 \leq layer \leq L$ **do**
2: Calculate the output of layer in the extended model: $e_i = \phi^i_{exten}(Z^{i-1})$
3: Calculate the output of layer in the recollection skeleton: $r_i = \phi^i_{reco}(Z^{i-1})$
4: Element-level summation by Equation (3)
5: $layer + 1$
6: **end for**
7: **return** Z^L

The input of the structural extension module is the output of the fixed model. For the layer i of the extended model, its input vector Z^i is the fusion of output of layer $i-1$ both in the extended model and the recollection skeleton, denoted as:

$$Z^i = r_{i-1} \oplus e_{i-1}, \forall i = \frac{L}{2} + 2, \ldots, L \tag{3}$$

where e_{i-1} is the outputs of the layer $i-1$ in the extended model, r_{i-1} is the outputs of the layer $i-1$ in the recollection skeleton, $\oplus$ is the element-level summation of two vectors.

Because navigations that are introduced in Section 3.3 are learnable, we take the output of the last layer corresponding to the position of navigations $V_k = \{v_1, \ldots, v_M\}$ as the result of the transformer encoder. This result is then averaged to obtain the feature vector Z of the model:

$$Z = \frac{1}{M} \sum_{k=1}^{M} Z_k \tag{4}$$

At last, Z is fed to the subsequent classifier module for classification.

Before the model learns a new task, the recollection skeleton is frozen to store the parameters of the old model. When new data of the new tasks are added, the recollection skeleton structure guides the model to learn under the memories of the past. After the new tasks are added, the recollection skeleton is updated by this new data, and previous data is stored in the sample library.

It is worth noting that our model does not require the memory of the old class on the recollection skeleton to remain the same, but instead produces a memory that better fits the current stage as the new class changes. This prevents the overfitting of the model to the old class and the model can extract features that better express the properties of the new class. The structural module makes a better trade-off between plasticity and stability.

3.5. Optimizer

To achieve self-sustainment guidance representation, the optimization objectives of the model are as follows.

We use the cross-entropy loss L_{class} as the basic classification function. It determines the classification accuracy of the model and updates the model's parameters and navigations. The cross-entropy loss denoted as:

$$L_{class} = L_{ce}(\hat{y}, y)) = -[y \log(\hat{y}) + (1-y) \log(1-\hat{y})] \tag{5}$$

where y is the ground truth.

For the dynamic query navigation module, we proposed query loss function L_{query}. It denotes the difference between the keys selected by the dynamic query navigation module and the query features. It is used to update the keys. The query loss is denoted as:

$$L_{query} = \sum_{d_x} Cor(f_x, k_{d_x}) \tag{6}$$

where *Cor* is the cosine similarity calculation, given by Equation (2).

For the structural extension module, we use the recollection skeleton to guide the learning of the extended model. Only using the information interaction of the structure does not positively affect the stability of the model. Therefore, we add the knowledge distillation loss to enhance the stability. The recollection skeleton acts as the teacher model to guide the extended model learning under the past information. The knowledge distillation loss is denoted as:

$$L_{kd} = L_{ce}(q_{r_i}, q_{e_i})) = -[q_{e_i} \log(q_{r_i}) + (1 - q_{e_i}) \log(1 - q_{r_i})] \tag{7}$$

$$q_{z_i} = \frac{exp(\frac{z_i}{T})}{\sum_j exp(\frac{z_j}{T})} \tag{8}$$

where q_{r_i} is the output of the recollection skeleton, q_{e_i} is the output of the extended model, z_i means the predicted result of the model, and T is the temperature coefficient.

In conclusion, the loss of the base task L_{base} is defined as:

$$L_{base} = \lambda_1 L_{class} + \lambda_2 L_{query} \tag{9}$$

where λ_1, λ_2 are weights.

the loss of the incremental task L_{incre} is defined as:

$$L_{incre} = \gamma_1 L_{class} + \gamma_2 L_{query} + \gamma_3 L_{kd} \tag{10}$$

where γ_1 , γ_2 , γ_3 are weights. According to the reference [31], we can obtain $\gamma_1 + \gamma_3 = 1$.

4. Experiments and Results

4.1. Dataset and Experimental Design

Dataset. To verify the performance of our method, we evaluate the performance on two datasets. The first dataset MSTAR is the SAR image. We demonstrate the feasibility on SAR images of our method. This dataset is provided by U.S. Defense Advanced Research Projects Agency and Air Force Research Office which provides identification of ground vehicle targets. A total of 10 types of ground targets were included in the standard working acquisition conditions, namely 2S1 (cannon), BRDM2 (truck), BTR60 (armored transport vehicle), D7 (bulldozer), SN_132 (tank), SN_9563 (tank), SN_C71 (armored car), T62 (tank), ZIL131 (truck) and ZSU23/4 (cannon). Some of the images are shown in Figure 5. The orientation range of these targets is 0∼360°, and the orientation interval is about 1° or 2°. The resolution of the radar used to acquire the SAR images is 0.3 m × 0.3 m, and the target SAR image is 128 × 128 pixels. The training set data were collected at 17° imaging side view and the test set data were collected at 15° side view. The details of the sample data (target type, number of samples, target sequence number, acquisition side view, etc.) for the standard working acquisition condition are shown in Table 1. Furthermore, we observe that the targets of SAR images in MSTAR are all in the middle of the image, and the noise of edge pixels will instead interfere with target classification. We cut the SAR image of 128 × 128 pixels into 32 × 32 pixels without affecting the target condition, as shown in Figure 6.

The second dataset CIFAR100 [46] is a marker subset of the tiny image dataset commonly used in the field of object recognition. We test our method on the CIFAR100 to assess the extensiveness of our methods and to demonstrate that it can be used in different scenarios. The CIFAR100 dataset has 100 classes, such as aquatic animals, household electrical devices and large man-made outdoor things. Each class has 600 RGB images with a size of 32 × 32, and these images are divided into 500 training images and 100 testing images.

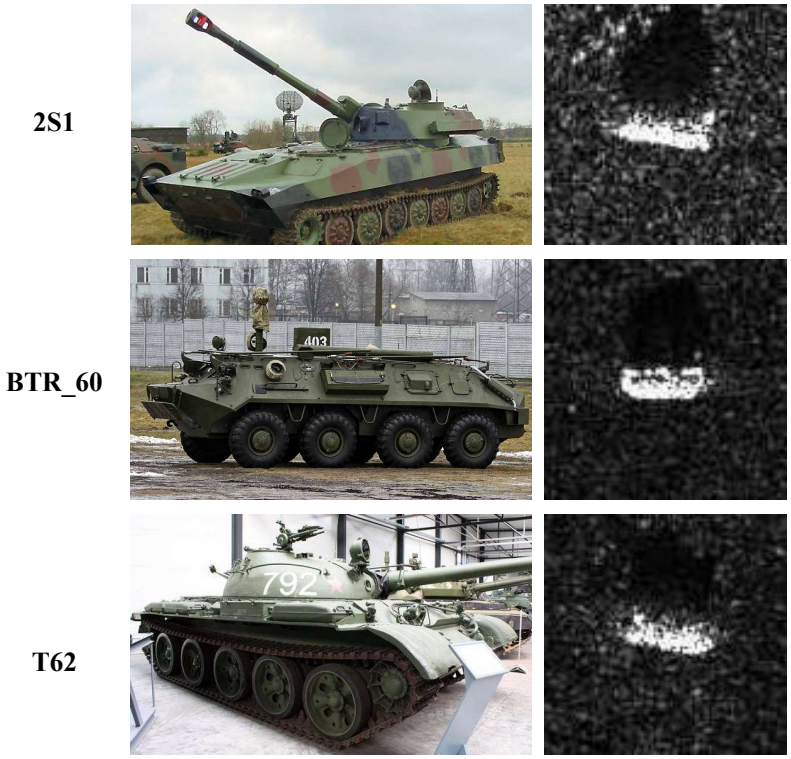

Figure 5. Some images of the MSTAR dataset and their optical comparison chart.

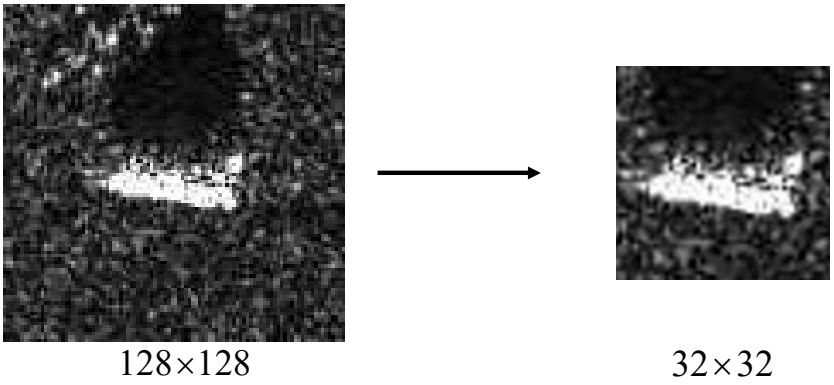

Figure 6. Sample cropping diagram of the MSTAR dataset.

Table 1. The details of the MSTAR dataset.

Target Type	Training Set (17°)	Testing Set (15°)
2S1	274	299
BRDM_2	274	298
BTR60	195	256
D7	274	299
SN_132	232	196
SN_9563	233	195
SN_C71	233	196
T62	273	299
ZIL131	274	299
ZSU23/4	274	299
Total	2536	2636

Protocol. We use the common incremental learning settings. The model will first learn half of the classes in the dataset. Then, the new classes are added incrementally in steps. For the MSTAR dataset, followed by reference [44], three classes of targets are learned first. The remaining seven classes are added incrementally with the incremental setting of 1, 2 and 3 classes accordingly. For the CIFAR100 dataset, followed by reference [30], 50 classes of targets are learned first. The remaining 50 classes are added incrementally with the incremental step of 5 steps of 10 classes, 10 steps of 5 classes and 25 steps of 2 classes.

Training details. These two datasets have the same training set. After each session, we randomly select 20 samples for each class and store them in the sample library for memory playback at the next incremental training. For the base task, we use the epochs of 30, the learning rate of 0.001, the weight decay of 0.9, the steps of 20, the momentum of 0.9 and the batch size of 32. For the loss functions, we set $\lambda_1 = \lambda_2 = 1$. For the incremental task, we use the epochs of 15, the learning rate of 0.001, the weight decay of 0.9, the steps of 10, the momentum of 0.9 and the batch size of 32. For the loss functions, we set

$$\gamma_1 = 0.2, \gamma_2 = 1, \gamma_3 = 0.8$$

In the dynamic query module, we set $N = 10, M = 5$.

Metrics. We use the metrics commonly used in incremental learning, average accuracy (A_i) and performance dropping rate (PD), to evaluate the incremental learning effect of the method. Following [39], we denote the average accuracy A_i as the Top-1 accuracy after the last session. Higher average accuracy represents better classification accuracy. Following [47], we use the performance dropping rate to assess the forgetting rate of incremental models for old classes:

$$PD = A_0 - A_T \tag{11}$$

where A_0 represents the classification accuracy of the classes at the first session and A_T represents the classification accuracy at the end of the last session.

4.2. Performance Analysis

In this section, we compare our method with several other incremental learning methods. For the MSTAR dataset, we first show the t-SNE visualization of the test class representation in the feature space when the model is learning the MSTAR dataset. Secondly, the average accuracy and the performance dropping rate of our method are compared with iCaRL [12], EEIL [48] and CBesIL [49]. For the CIFAR100 dataset, we conducted the same experiment with the MSTAR dataset. However, since almost all the incremental learning methods use RGB images, we compare a large number of image incremental learning methods, such as LwF [16], iCaRL [12], LUCIR [30], PODNet [14], BiC [18], MBP [38] and FOSTER [24]. Because there are too many categories in the CIFAR100 dataset, we do not show the t-SNE visualization of feature space.

Table 2 summarizes the results of our approach to testing on the MSTAR dataset. We adopt three incremental settings and show the average accuracy and the performance dropping rate in each setting. The results show that our method performs best compared to the other methods. Firstly, the basic accuracy of our method achieves 98.49%, which is nearly 12% more than iCaRL. Secondly, our method performs better than other methods in all incremental settings. Especially with the incremental setting of $T = 3$ $(S = 3)$, our methods achieve an average accuracy of 79.98%, which is an increase of 6.64% over EEIL. Thirdly, for the performance dropping rate, our approach also achieves better results. With the incremental setting of $T = 3$ $(S = 3)$, the performance dropping rate of our method is 18.81%, which is 7.5% lower than CBesIL. The effectiveness of our proposed method on SAR data can be demonstrated from all aspects.

Table 3 summarizes the results of our benchmark tests on the CIFAR100 dataset. The result shows that our method achieves the best results under different incremental segmentation cases. With the incremental learning setting of $T = 25$ $(S = 2)$, our method

achieves an accuracy of 71.33%, which is about 8% higher than the current better method FOSTER. Notably, the accuracy of our method hardly decreases as the number of tasks increases. At the same time, the accuracy of BIC decreases by 5.33%. This effectively proves that our approach demonstrates good results for future sustainable use. Furthermore, it can be seen that the performance dropping rate of our method is only 9.98% with the incremental learning setting of $T = 10(S = 5)$, which is 4.5% lower than FOSTER. This demonstrates that our method can deal with the catastrophic forgetting problem and can be applied in the field of image recognition.

Table 2. Results on MSTAR (more than 3 runs on average). T is the number of incremental tasks and S is the number of classes of each incremental session. We count each method's average accuracy rate A and performance dropping rate PD.

Methods	$A_0(\%) \uparrow$	$T = 7$ $(S = 1)$		$T = 4$ $(S = 2)$		$T = 3$ $(S = 3)$	
		$A_T(\%) \uparrow$	$PD(\%) \downarrow$	$A_T(\%) \uparrow$	$PD(\%) \downarrow$	$A_T(\%) \uparrow$	$PD(\%) \downarrow$
iCaRL [12]	87.47	63.24	24.23	65.02	22.45	67.52	19.95
EEIL [48]	98.67	71.20	27.47	72.87	25.80	73.34	25.33
CBesIL [49]	98.06	72.15	25.91	70.21	27.85	71.75	26.31
OURS	**98.49**	**74.65**	**24.14**	**77.15**	**21.64**	**79.98**	**18.81**

Table 3. Results on CIFAR100 (more than 3 runs on average). T is the number of incremental tasks and S is the number of classes of each incremental session. We count each method's average accuracy rate A and performance dropping rate PD.

Methods	$T = 25$ $(S = 2)$		$T = 10$ $(S = 5)$		$T = 5$ $(S = 10)$	
	$A_T(\%) \uparrow$	$PD(\%) \downarrow$	$A_T(\%) \uparrow$	$PD(\%) \downarrow$	$A_T(\%) \uparrow$	$PD(\%) \downarrow$
LwF [16]	45.07 ± 0.35	41.14	47.51 ± 0.77	43.66	49.78 ± 0.35	47.40
iCaRL [12]	49.15 ± 0.53	34.92	52.96 ± 0.95	31.60	57.03 ± 0.42	19.92
LUCIR [30]	56.68 ± 0.19	27.90	59.93 ± 2.40	27.74	63.46 ± 0.08	24.180
PODNet [14]	61.00 ± 1.62	20.39	63.76 ± 0.81	19.16	66.98 ± 0.65	15.98
BiC [18]	63.41 ± 0.27	19.59	66.24 ± 0.43	17.86	69.97 ± 0.46	13.02
MBP [38]	62.61 ± 0.41	17.95	64.29 ± 0.41	13.16	66.93 ± 0.41	11.596
FOSTER [24]	63.83 ± 0.82	15.86	67.95 ± 0.95	10.23	-	-
OURS	**71.33 ± 0.28**	**10.67**	**72.02 ± 0.66**	**9.98**	**71.17 ± 0.52**	**10.83**

The variations of the average accuracy with the classes increase in the two datasets are shown in Figure 7. This demonstrates that the method outperforms other methods most of the time. Our method maintains a high accuracy rate both on the MSTAR and CIFAR100 datasets. It proves that our method can be used in SAR recognition and more generalized scenarios. Moreover, with the new classes added, the accuracy of all methods is decreasing. The performance dropping rate seems more essential to judge the merits of a method. By the value of the difference between the starting point and the ending point on the vertical coordinate, all of the variation pictures show that our method can effectively alleviate forgetfulness.

Additionally, Figure 8 shows the t-SNE visualization on the MSTAR dataset under the incremental settings of $T = 7$ $(S = 1)$. This shows that our method is able to distinguish the different categories in the feature space as the number of categories increases. The data of different categories are better differentiated. In addition to task 1, the old categories in the following tasks can be distinguished well, with only a small number of retrained samples involved in the training. From these pictures, it can be observed that the 20 samples near the center of the category can roughly determine the distribution of the characteristics of the class.

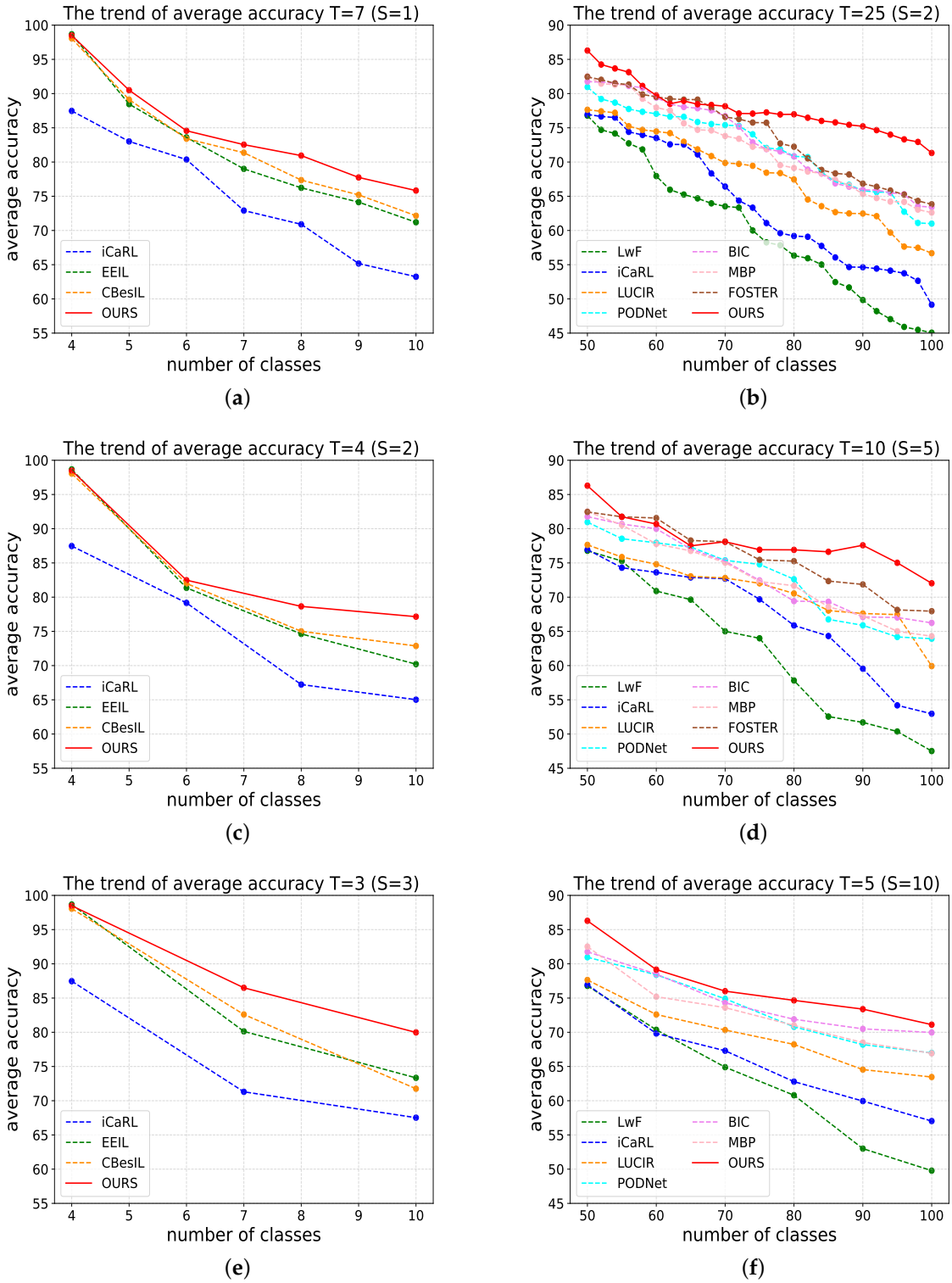

Figure 7. The variations of the average accuracy under different incremental settings. (**a**,**c**,**e**) show the results on the MSTAR compared with iCaRL, EEIL and CBesIL. (**b**,**d**,**f**) show the results on the CIFAR100 compared with LwF, iCaRL, LUCIR, PODNet, BiC, MBP and FOSTER. (**a**) $T = 7$ $(S = 1)$ of MSTAR; (**b**) $T = 10$ $(S = 5)$ of CIFAR100; (**c**) $T = 4$ $(S = 2)$ of MSTAR; (**d**) $T = 5$ $(S = 10)$ of CIFAR100; (**e**) $T = 3$ $(S = 3)$ of MSTAR; (**f**) $T = 25$ $(S = 2)$ of CIFAR100.

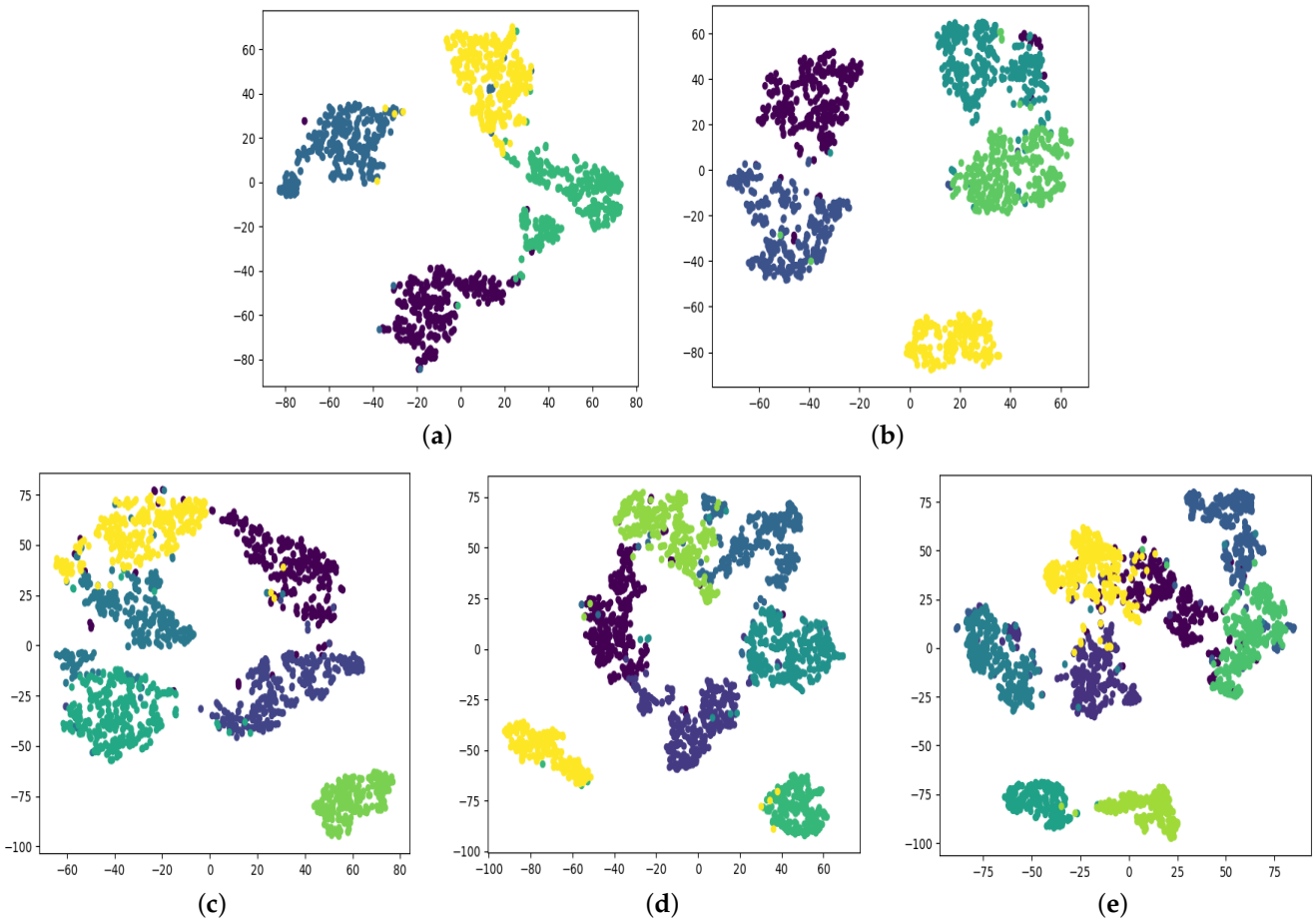

Figure 8. The t-SNE visualization of the test class representation in the feature space under the incremental setting of $T = 7$ $(S = 1)$. Different colors represent different classes and the spot is the sample. (**a**–**e**) Incremental process of the eight classes formed from the initial four classes. (**a**) Task 1; (**b**) Task 2; (**c**) Task 3; (**d**) Task 4; (**e**) Task 5.

4.3. Ablation Study

Four ablation studies were conducted to evaluate the role of each module in our approach. The ablation studies first tested the effect of the dynamic query navigation module and the structural extension module on the model. Then, the effect of knowledge distillation loss on the results was tested. In addition, to investigate the specific values of the hyperparameters, we also explored the impact of the weights of the knowledge distillation loss function. The effect of retaining different numbers of samples on classification accuracy and memory.

Table 4 summarizes the impact of the dynamic query navigation module, the structural extension module and the knowledge distillation loss on MSTAR with an incremental learning setting of $T = 3$ $(S = 3)$. Without the dynamic query navigation module, the average accuracy of the model is just 67.79%, which is a 6.86% decrease. The possible reason for the decrease in accuracy is the reduced accuracy of the model in recognizing new categories. When learning new categories, the significant features that help in recognition cannot be found precisely. Moreover, the lower learning accuracy can lead to difficulties in subsequent tasks in accurately identifying the categories that appear in the present task. In addition, without the structural extension module, the average accuracy of the model is 65.43%, which is a decrease of 9.22% . In this case, the reason for the reduced accuracy of the model is the reduced ability to recall the old classes. Since a large number of categories are old, the overall accuracy of the model is even lower. For the knowledge distillation loss L_{kd}, the accuracy of the model on the MSTAR dataset was 67.01% in the absence of it. This

is 7.64% less accurate than adding the distillation losses. These ablation studies show that the various modules of our method and the distillation loss function have a positive impact on the results.

Table 4. The effectiveness of each component in our method on MSTAR. We count the average accuracy at the end of the last session A_T with an incremental learning setting of $T = 3$ $(S = 3)$.

Methods			$A_T(\%) \uparrow$
The Dynamic Query Navigation	**The Structural Extension**	L_{kd}	
✓	✓	✓	74.65
✓		✓	65.43
	✓	✓	67.79
✓	✓		67.01

Table 5 summarizes the impact of our approach under the dynamic query navigation module and structural extension module. The first line shows that the average accuracy rate is 71.17% when both two modules are present. The second row indicates that without the structural extension module and using only the dynamic query navigation module, the average accuracy is 64.90%, which is a decrease of 6.22% compared to the first line. The third line indicates that without the dynamic query navigation module and only the structural extension module added, the average accuracy is 65.85%, which is a decrease of 5.27% compared to the first line.

Table 5. The effectiveness of each component in our method on CIFAR100. We count the average accuracy at the end of the last session A_T on the CIFAR100 dataset with an incremental learning setting of $T = 5$ $(S = 10)$.

Methods		$A_T(\%) \uparrow$
The Dynamic Query Navigation	**The Structural Extension**	
✓	✓	71.17
✓		64.90
	✓	65.85

Figure 9 shows the effect of different weights of γ_1 and γ_3 on the results under the condition that $\gamma_1 + \gamma_3 = 1$ is satisfied. In this experiment, we set the solid weights, $\lambda_1 = \lambda_2 = \gamma_2 = 1$, and we conduct the experiment on the MSTAR dataset with the incremental setting of $T = 3$ $(S = 3)$. According to the literature [31], the larger γ_3 has a better effect on the results. Thus, we start the experiment with both equal and make γ_3 increase gradually. We can see from the results that the overall trend of normal distribution. From the results, we can see that there is an overall trend of normal distribution. The possible reason for this distribution is that when the loss weight of knowledge distillation is small, the model is less capable of remembering old classes. When the loss weight of distillation is larger, the model is less capable of learning new classes. Moreover, for incremental learning, the recall ability of the model is more important. This is because future tasks still need to recall information about the current task. Based on the information in the table, we choose the case of $\gamma_1 = 0.3, \gamma_3 = 0.7$ with the average accuracy is 70.21%, a moment that allows the model to achieve a better trade-off between stability and balance.

Figure 10 shows the effect of the number of samples stored per class in memory. We conducted the experiment on the MSTAR dataset with the incremental setting of $T = 3$ $(S = 3)$. In general, as the number of stored samples increases, the memory of past data is better and the average accuracy is higher. Therefore, we designed the experiments with the increment of stored samples. It can be observed that there is a large improvement in accuracy as the stored samples are increased from 10 to 20 per class. However, the increase

in accuracy from 20 to 40 per class does not have much effect on the accuracy. Considering the memory limitation, we chose to store 20 samples per class as the experimental setup.

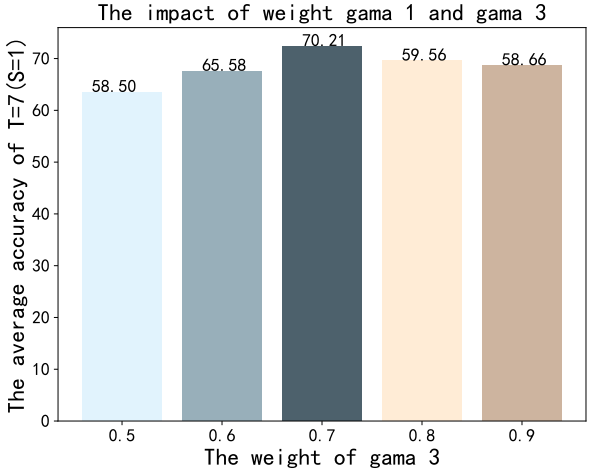

Figure 9. The effect of γ_1 and γ_3 on the results. The horizontal coordinates indicate B and the vertical coordinates indicate the accuracy. It can be seen that the best results are $\gamma_1 = 0.3, \gamma_3 = 0.7$.

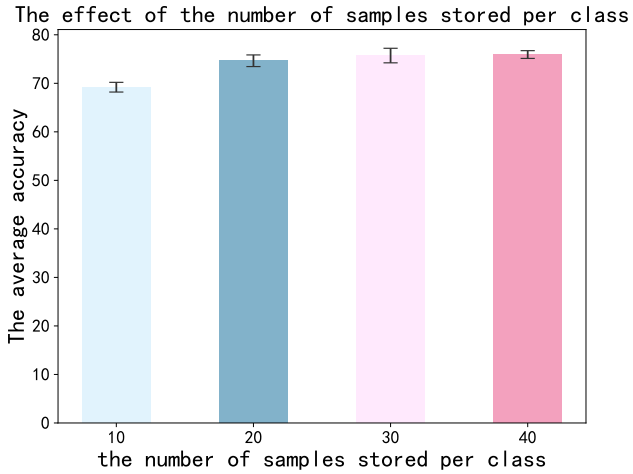

Figure 10. The effect of the number of samples stored per class. The short line in the graph represents the error. It can be observed that the accuracy increases with the number of stored samples. However, considering both accuracy and memory, 20 samples are chosen to be stored per class.

5. Conclusions

In this work, we propose a Class-Incremental Learning method for SAR images based on the self-sustainment guidance representation method to enhance the plasticity and stability of the model in incremental learning. We use the vision transformer as the basic structure for feature extraction and classification. Furthermore, we design a dynamic query navigation module to maintain the model's learning capability for new classes. This module retrains the special information of classes and uses this information to expand the input for guiding the direction of feature extraction in the transformer encoder. Additionally, we design a structural extension module to enhance the model's ability to recognize previous classes. This module introduces the recollection skeleton as a recall medium for remembering old knowledge when new data are added. At last, experiments on the MSTAR dataset and CIFAR100 dataset demonstrate that our method achieves extraordinary results on SAR datasets and can be extended to use on general image data. In the subsequent research, we will aim to make the following improvements to the method: (1) Considering the difficulty of obtaining past data in practical application scenarios, we will conduct

experiments without playback data. (2) Considering the characteristics of SAR images, we will make improvements specific to SAR images and the subsequent research will be more specific to SAR images.

Author Contributions: Conceptualization, Q.P.; methodology, Q.P. and X.H.; software, Q.P.; validation, X.H.; formal analysis, Q.P. and K.L.; investigation, X.H.; resources, K.L.; data curation, Z.B.; writing—original draft preparation, Q.P.; writing—review and editing, Q.P. and K.L.; visualization, J.H.; supervision, K.L. and J.H.; project administration, K.L.; funding acquisition, K.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Fundamental Research Funds for the Central Universities under Grant ZYGX2020ZB030.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank the anonymous referees for their suggestions and comments.

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

Abbreviations

The following abbreviations are used in this manuscript:

SAR	Synthetic Aperture Radar
CIL	Class-Incremental Learning
ViT	vision transformer
MSTAR	Moving and Stationary Target Acquisition and Recognition
CL	Continue Learning
LLL	Life-Long Learning
CLS	Complementary Learning Systems
NLP	Natural Language Processing

References

1. Curlander, J.C.; McDonough, R.N. *Synthetic Aperture Radar*; Wiley: New York, NY, USA, 1991; Volume 11.
2. Chen, S.; Wang, H. SAR target recognition based on deep learning. In Proceedings of the 2014 International Conference on Data Science and Advanced Analytics (DSAA), Shanghai, China, 30 October–1 November 2014; pp. 541–547.
3. Richards, M.A.; Scheer, J.; Holm, W.A.; Melvin, W.L. *Principles of Modern Radar*; Citeseer: Princeton, NJ, USA, 2010; Volume 1.
4. Chierchia, G.; Cozzolino, D.; Poggi, G.; Verdoliva, L. SAR image despeckling through convolutional neural networks. In Proceedings of the 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Fort Worth, TX, USA, 23–28 July 2017; pp. 5438–5441.
5. Lin, Z.; Ji, K.; Kang, M.; Leng, X.; Zou, H. Deep convolutional highway unit network for SAR target classification with limited labeled training data. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 1091–1095. [CrossRef]
6. Gao, F.; Xu, J.; Lang, R.; Wang, J.; Hussain, A.; Zhou, H. A Few-Shot Learning Method for SAR Images Based on Weighted Distance and Feature Fusion. *Remote Sens.* **2022**, *14*, 4583. [CrossRef]
7. Goodfellow, I.J.; Mirza, M.; Xiao, D.; Courville, A.; Bengio, Y. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv* **2013**, arXiv:1312.6211.
8. Robins, A. Catastrophic forgetting, rehearsal and pseudorehearsal. *Connect. Sci.* **1995**, *7*, 123–146. [CrossRef]
9. Lange, M.D.; Jia, X.; Parisot, S.; Leonardis, A.; Slabaugh, G.; Tuytelaars, T. Unsupervised model personalization while preserving privacy and scalability: An open problem. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 5 August 2020; pp. 14463–14472.
10. Chen, Z.; Liu, B. Lifelong machine learning. *Synth. Lect. Artif. Intell. Mach. Learn.* **2018**, *12*, 1–207.
11. Grossberg, S.T. *Studies of Mind and Brain: Neural Principles of Learning, Perception, Development, Cognition, and Motor Control*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2012; Volume 70.
12. 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. 2001–2010.

13. 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.
14. Douillard, A.; Cord, M.; Ollion, C.; Robert, T.; Valle, E. Podnet: Pooled outputs distillation for small-tasks incremental learning. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 86–102.
15. 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]
16. Li, Z.; Hoiem, D. Learning without forgetting. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 2935–2947. [CrossRef]
17. Mallya, A.; Lazebnik, S. Packnet: Adding multiple tasks to a single network by iterative pruning. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7765–7773.
18. 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.
19. O'Reilly, R.C.; Bhattacharyya, R.; Howard, M.D.; Ketz, N. Complementary learning systems. *Cogn. Sci.* **2014**, *38*, 1229–1248. [CrossRef]
20. Liu, P.; Yuan, W.; Fu, J.; Jiang, Z.; Hayashi, H.; Neubig, G. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.* **2023**, *55*, 1–35. [CrossRef]
21. Wang, Z.; Zhang, Z.; Lee, C.Y.; Zhang, H.; Sun, R.; Ren, X.; Su, G.; Perot, V.; Dy, J.; Pfister, T. Learning to prompt for continual learning. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 139–149.
22. Zhou, D.W.; Ye, H.J.; Zhan, D.C. Co-transport for class-incremental learning. In Proceedings of the 29th ACM International Conference on Multimedia, Virtual, 20–24 October 2021; pp. 1645–1654.
23. Wu, T.Y.; Swaminathan, G.; Li, Z.; Ravichandran, A.; Vasconcelos, N.; Bhotika, R.; Soatto, S. Class-incremental learning with strong pre-trained models. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 9601–9610.
24. Wang, F.Y.; Zhou, D.W.; Ye, H.J.; Zhan, D.C. Foster: Feature boosting and compression for class-incremental learning. In Proceedings of the Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, 23–27 October 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 398–414.
25. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16 $\times$ 16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
26. Zhu, F.; Zhang, X.Y.; Wang, C.; Yin, F.; Liu, C.L. Prototype augmentation and self-supervision for incremental learning. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 5871–5880.
27. Smith, J.; Hsu, Y.C.; Balloch, J.; Shen, Y.; Jin, H.; Kira, Z. Always be dreaming: A new approach for data-free class-incremental learning. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 9374–9384.
28. Zhu, K.; Zhai, W.; Cao, Y.; Luo, J.; Zha, Z.J. Self-sustaining representation expansion for non-exemplar class-incremental learning. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 9296–9305.
29. Gao, Q.; Zhao, C.; Ghanem, B.; Zhang, J. R-DFCIL: Relation-Guided Representation Learning for Data-Free Class Incremental Learning. In Proceedings of the Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, 23–27 October 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 423–439.
30. 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.
31. Hinton, G.; Vinyals, O.; Dean, J. Distilling the knowledge in a neural network. *arXiv* **2015**, arXiv:1503.02531.
32. Xu, M.; Zhao, Y.; Liang, Y.; Ma, X. Hyperspectral Image Classification Based on Class-Incremental Learning with Knowledge Distillation. *Remote Sens.* **2022**, *14*, 2556. [CrossRef]
33. Aljundi, R.; Babiloni, F.; Elhoseiny, M.; Rohrbach, M.; Tuytelaars, T. Memory aware synapses: Learning what (not) to forget. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 139–154.
34. Aljundi, R.; Kelchtermans, K.; Tuytelaars, T. Task-free continual 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. 11254–11263.
35. Zenke, F.; Poole, B.; Ganguli, S. Continual learning through synaptic intelligence. In Proceedings of the International Conference on Machine Learning, PMLR, Sydney, NSW, Australia, 6–11 August 2017; pp. 3987–3995.
36. Tao, X.; Chang, X.; Hong, X.; Wei, X.; Gong, Y. Topology-preserving class-incremental learning. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 254–270.
37. Rajasegaran, J.; Hayat, M.; Khan, S.H.; Khan, F.S.; Shao, L. Random path selection for continual learning. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 12669–12679.

38. Liu, Y.; Hong, X.; Tao, X.; Dong, S.; Shi, J.; Gong, Y. Model behavior preserving for class-incremental learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, 1–12. [CrossRef]
39. Tao, X.; Hong, X.; Chang, X.; Dong, S.; Wei, X.; Gong, Y. Few-shot 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. 12183–12192.
40. Wang, C.; Qiu, Y.; Gao, D.; Scherer, S. Lifelong graph learning. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 13719–13728.
41. Tao, L.; Jiang, X.; Li, Z.; Liu, X.; Zhou, Z. Multiscale incremental dictionary learning with label constraint for SAR object recognition. *IEEE Geosci. Remote Sens. Lett.* **2018**, *16*, 80–84. [CrossRef]
42. Zheng, Z.; Nie, X.; Zhang, B. Fine-Grained Continual Learning for SAR Target Recognition. In Proceedings of the IGARSS 2022—2022 IEEE International Geoscience and Remote Sensing Symposium, Kuala Lumpur, Malaysia, 17–22 July 2022; pp. 2207–2210.
43. Wang, Z.; Li, H.X.; Chen, C. Incremental reinforcement learning in continuous spaces via policy relaxation and importance weighting. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *31*, 1870–1883. [CrossRef]
44. Wang, L.; Yang, X.; Tan, H.; Bai, X.; Zhou, F. Few-Shot Class-Incremental SAR Target Recognition Based on Hierarchical Embedding and Incremental Evolutionary Network. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*. [CrossRef]
45. Khan, S.; Naseer, M.; Hayat, M.; Zamir, S.W.; Khan, F.S.; Shah, M. Transformers in vision: A survey. *ACM Comput. Surv. (CSUR)* **2022**, *54*, 1–41. [CrossRef]
46. Krizhevsky, A.; Hinton, G. Learning multiple layers of features from tiny images. *Comput. Sci.* **2009**, 32–33. Available online: https://www.cs.toronto.edu/kriz/learning-features-2009-TR.pdf (accessed on 22 March 2023).
47. Zhou, D.W.; Ye, H.J.; Ma, L.; Xie, D.; Pu, S.; Zhan, D.C. Few-shot class-incremental learning by sampling multi-phase tasks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, 1–16. [CrossRef]
48. Castro, F.M.; Marín-Jiménez, M.J.; Mata, N.G.; Schmid, C.; Karteek, A. End-to-End Incremental Learning. *arXiv* **2018**, arXiv:1807.09536.
49. 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]