Overcoming Catastrophic Forgetting for Multi-Label Class-Incremental Learning

Xiang Song[1], Kuang Shu[2,*], Songlin Dong[3], Jie Cheng[4], Xing Wei[1,✉], Yihong Gong[1,3]
[1]School of Software Engineering, Xi'an Jiaotong Univerisity, Xi'an, China
[2]Central Southern China Electric Power Design Institute Co.,
ltd. of China Power Engineering Consulting Group, Wuhan, China
[3]College of Artificial Intelligence, Xi'an Jiaotong University, Xi'an, China
[4]Huawei Base, Bantian, Shenzhen, China
songxiang@stu.xjtu.edu.cn, sk1123344@163.com, dsl972731417@stu.xjtu.edu.cn,
chengjie8@huawei.com, {weixing,ygong}@mail.xjtu.edu.cn

Abstract

Despite the recent progress of class-incremental learning (CIL) methods, their capabilities in real-world scenarios such as multi-label settings remain unexplored. This paper focuses on a more practical CIL problem named multi-label class-incremental learning (MLCIL). MLCIL requires the vision models to overcome catastrophic forgetting of old knowledge while learning new classes from multi-label samples. Direct application of existing CIL methods to MLCIL leads to label absence, representative sample selection, and feature dilution problems. To address these problems, we present a novel AdaPtive Pseudo-Label-drivEn (APPLE) framework consisting of three components. First, the adaptive pseudo-label strategy is proposed to solve the label absence problem, which leverages the old model to annotate old classes for new samples. Second, a cluster sampling strategy is proposed to obtain more diverse samples to alleviate catastrophic forgetting under the MLCIL setting better. Finally, a class attention decoder is designed to mitigate the object feature dilution problem in multi-label samples. The extensive experiments on PASCAL VOC 2007 and MS-COCO demonstrate that our proposed method significantly outperforms other representative state-of-the-art CIL methods.

1. Introduction

Class-incremental learning (CIL) [19, 27, 36, 39, 40] refers to that with the continuous arrival of new datasets, the deep neural network model learns new classes while keeping the knowledge of old classes from being forgotten. Many efforts [4, 8, 17, 27, 43–45, 49–51] have been made by the artificial intelligence community to mitigate the catastrophic forgetting problem in CIL, promoting the recognition performance in all encountered classes. Most of them share a similar structure, *i.e.*, the knowledge distillation (KD) [15] on the model's feature extractor, and auxiliary

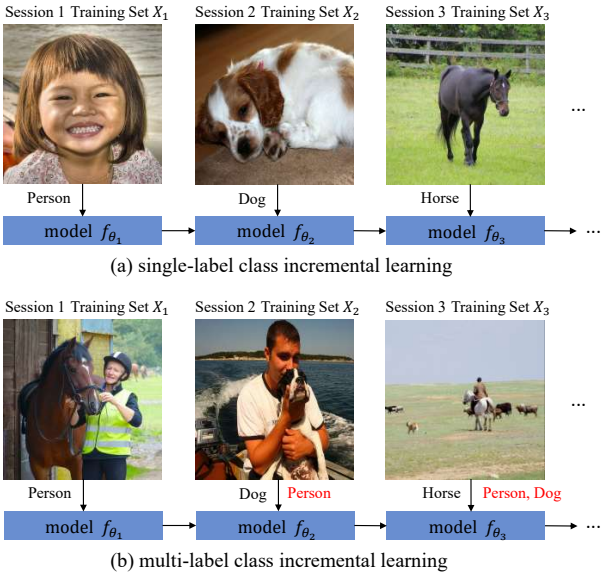


Figure 1. Comparison of single-label incremental learning and multi-label incremental learning scenarios. The former has been extensively studied, in which samples are usually single-label and the model only needs to focus on new classes in each new session. In contrast, the latter studies a more practical problem, in which samples usually have multiple objects, and the model not only needs to learn new classes (marked in black) but also classify old ones (marked in red) in the new dataset.

Xiang Song and Kuang Shu* contribute equally to this work. Xing Wei✉ is the corresponding author.

data memory which preserves a small number of preceding representative samples. Although each part keeps pace with the up-to-date methods, existing CIL methods only focus on single-label classification and ignore the more general multi-label classification problem in the real world. We term the CIL method with multi-label classification capability as *multi-label class-incremental learning* (MLCIL). As shown in Fig. 1(a), in the (single-label) CIL setting, a sample contains only one new class, and the model only needs to classify this class. However, in the MLCIL setting, a sample contains multiple objects, and the model needs to classify not only new classes (marked in black) but also old ones (marked in red). For example, in Fig. 1(b), the model has learned to classify the person in session 1, and then in session 2, it needs to learn a new class (Dog) and retain the ability to classify the old one (Person). Likewise, in session 3, the model learns a new class (Horse) and is required to classify old ones (Person and Dog).

A feasible way to solve the MLCIL problem is to migrate the existing CIL methods to the MLCIL problem. However, compared with CIL, the MLCIL problem poses several new challenges: 1) **Label absence problem**. A sample always contains multiple objects, where old classes are not labeled, leading to a label absence problem. This situation could lead to more serious catastrophic forgetting; 2) **Multi-label representative sample selection.** Existing rehearsal-based CIL methods select representative single-label samples based on the entire image. It's hard for them to handle multi-label samples containing multiple classes; 3) **Feature dilution problem.** Objects of inconsistent size are distributed in different locations. Classical CIL methods directly extract features from the whole image, which may cause the features of some objects (such as smaller, less prominent, or poor image quality objects) to be diluted, resulting in poor classification results. Furthermore, another plausible approach is implementing the class-incremental object detection (CIOD) or semantic segmentation (CISS) methods in MLCIL. However, although CIOD and CISS also focus on the multi-label prediction problem, they require bounding-box information or pixel-level mask annotations, which greatly increase the labor cost. On the contrary, the labels of multi-label classification are simple (only the object category information in the image is required).

To address the MLCIL problem, we propose an AdaPtive Pseudo-Label-drivEn (APPLE) framework, which consists of an adaptive pseudo-label strategy, a cluster sampling strategy, and a class attention decoder module. The adaptive pseudo-label strategy leverages the old model from the previous session to label new samples, thereby alleviating the impact of label absence. The cluster sampling strategy captures the features of each class in the multi-label image, thus saving more representative samples. The class attention decoder is utilized to encode spatial information in feature maps, so as to help the model learn better representations, solving the feature dilution problem. Moreover, we adopt the knowledge distillation (KD) [15] to alleviate the catastrophic forgetting problem. To demonstrate the effectiveness of our method, we conduct comparative experiments with representative CIL methods on two benchmark datasets, MS-COCO [22] and PASCAL VOC 2007 [11]. To summarize, our main contributions are as follows:

- We propose the APPLE framework, a novel approach aimed at addressing the more realistic *multi-label class-incremental learning* (MLCIL) problem.
- Our method comprises three components to solve challenges that MLCIL brings: an adaptive pseudo-label to alleviate the impact of label absence, a cluster sampling strategy to boost the quality of replay data, and the class attention decoder to promote the learning ability of the overall model.
- Extensive experiments on MS-COCO and PASCAL VOC 2007 datasets demonstrate that our proposed APPLE framework outperforms state-of-the-art CIL methods on the MLCIL problem.

2. Related Work

2.1. Class-Incremental Learning

Recently, incremental learning research mainly focuses on the class-incremental learning (CIL) problem, which aims to learn a unified model that is able to classify all the encountered classes through a series of incremental learning sessions. The major challenge of incremental learning is the catastrophic forgetting problem, *i.e.*, the model's performance improves quickly in new classes, whereas deteriorates sharply in old ones in each session. In order to solve such issue, extensive works have been devoted to devising CIL methods, which can be primarily divided into regularization-based, rehearsal-based, and architectural-based methods.

Regularization-based methods introduce regularization terms [1, 19, 32] into the loss functions to limit the changes of critical parameters during the learning of new sessions. James *et al.* [19] first attempt to illustrate this idea, which uses the Fisher matrix to calculate an importance weight for each parameter in the model, and then regularize the parameters using these weights. Another regularization solution is to use distillation to prevent the forgetting of old knowledge. For example, Li *et al.* [21] use the distillation loss function to constrain the output logits change between the old model and new model to mitigate knowledge forgetting. Castro *et al.* [4] distill and integrate multiple classifiers to obtain end-to-end classification results. PODNet [10] finds that knowledge distillation loss applied to each stage of the backbone can enhance model stability.

Rehearsal-based methods first store a small number of representative samples from previous data and then train with new data together. iCaRL [27] is the first of this kind of method, which provides a herding strategy to update its exemplars and incrementally learns a nearest-neighbor classifier for new classes. ER [30] constructs a memory buffer to save samples from old sessions for replay. TPCIL [35] utilizes topology-preserving loss to mitigate the forgetting of old knowledge. PASS [51] proposes to preserve prototypes instead of samples, showing memory efficiency. DER++ [45] addresses the general incremental learning problem through mixing rehearsal with knowledge distillation and regularization. Another rehearsal-based method is to use generative algorithms to synthesize previous data. Shin *et al.* [34] utilize a generative model to mimic past data. Ven *et al.* [36] propose the generative classification strategy which uses Bayes' rule to perform classification.

Architectural-based methods use different parameters to handle different tasks. Progressive neural networks [31] take the first shot. It simply adds a whole network and fully connects it with the previous ones to learn the new tasks, so there is no conflict on different tasks. However, this operation results in parameter increases over time. Several methods [12, 20, 33, 48] have been proposed to handle this limitation. Moreover, Wang *et al.* [39, 40] propose to learn independent prompts for each incremental task based on a pre-trained ViT model, which achieves state-of-the-art results on multiple single-label incremental learning tasks.

2.2. Multi-Label Classification

Multi-Label Classification task has attracted increasing interest recently. One method to solve the multi-label classification problem is locating regions of interest. Earlier methods [24, 41, 46] use proposals and divide multi-label classification problems into single-label classification problems. Wang *et al.* [38] propose to use the spatial transformation layer to identify regions with semantic information, and then find their interrelationships through long short-term memory network [16]. Later works like [13] introduce attention mechanisms to boost these methods further. Inspired by the work [5], several works [6, 37, 47] try to improve the multi-label classification results by modeling label correlations. However, there are some arguments that spurious correlations can be learned when the label statistics are insufficient. Another method to solve the multi-label classification problem is improving loss functions. Several methods [2, 42] modify the loss to reduce the impact of negative samples and fit this situation well. Lately, some approaches [23, 29] leverage the modeling ability of transformers [9] to implicitly capture the label correlations and achieve better results.

In this paper, we focus on the MLCIL setting and address this new problem by drawing lessons from class-incremental learning and multi-label classification research.

3. Methodology

In this section, we first illustrate the formulation of MLCIL and then show how each part in our proposed framework addresses this problem. The overall framework is shown in Fig. 2.

3.1. Problem Definition

In real-world classification applications, when we want to train a model, since we may only be interested in several classes, or only have the energy to label a small part, there are often unlabeled classes in the training set. The goal of MLCIL is to learn a unified model to recognize all learned classes that are presented in test samples. Considering a stream of T incremental sessions $\{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_T, \mathcal{Z}\}$, $\mathcal{D}_t$ is the training set of session t and all sessions share a common test set $\mathcal{Z}$. Meanwhile, we denote the data distribution as $\mathcal{D}_t = \{X_t, Y_t\}$ for each session t, where X_t is the training sample set, and Y_t means the annotated label set for session t and $\cap_{t=1}^{T} Y_t = \varnothing$. Furthermore, we define the label set $\tilde{Y}_t = \cup_{i=1}^{t} Y_i$, which represents the classes that are expected to be recognized by the model in session t. In each session t, X_t is the only accessible training set, and the obtained model will be evaluated using the test set $\mathcal{Z}_t$, where $\mathcal{Z}_t$ is produced by labeling $\mathcal{Z}$ with $\tilde{Y}_t$.

3.2. Framework

An image classification model f_θ can be parameterized by θ. Initially, we train the base model f_{θ_1} using $\mathcal{D}_1$ with the multi-label classification loss L_{ASL} (see Eq. (7) in Sec. 3.6). Next, using L_{ASL} and L_{KD}, we incrementally train the base model on $\mathcal{D}_2, \mathcal{D}_3, \cdots$, obtaining $f_{\theta_2}, f_{\theta_3}, \cdots$, respectively.

As described in Sec. 1, simply fine-tuning the base model on a new training set without any constraints causes the catastrophic forgetting problem, resulting in significant performance drops on old test data. To tackle this problem, as shown in Fig. 2, we propose the APPLE framework, which consists of an adaptive pseudo-label strategy, a cluster sampling strategy, and a class attention decoder (CAD) module. Specifically, after getting model $f_{\theta_{t-1}}$, we use the cluster sampling strategy to select representative samples of each class as the replay data, which can help the model recall the old knowledge better. Then when training in the new session t, we freeze the model $f_{\theta_{t-1}}$ to generate pseudo-labels, which are combined with the current labels to jointly train the new model f_{θ_t}. Moreover, the learnable class tokens are fed into the CAD along with the output of the backbone to better focus on spatial information, alleviating the object feature dilution problem. The following parts of this section provide detailed descriptions of these three components.

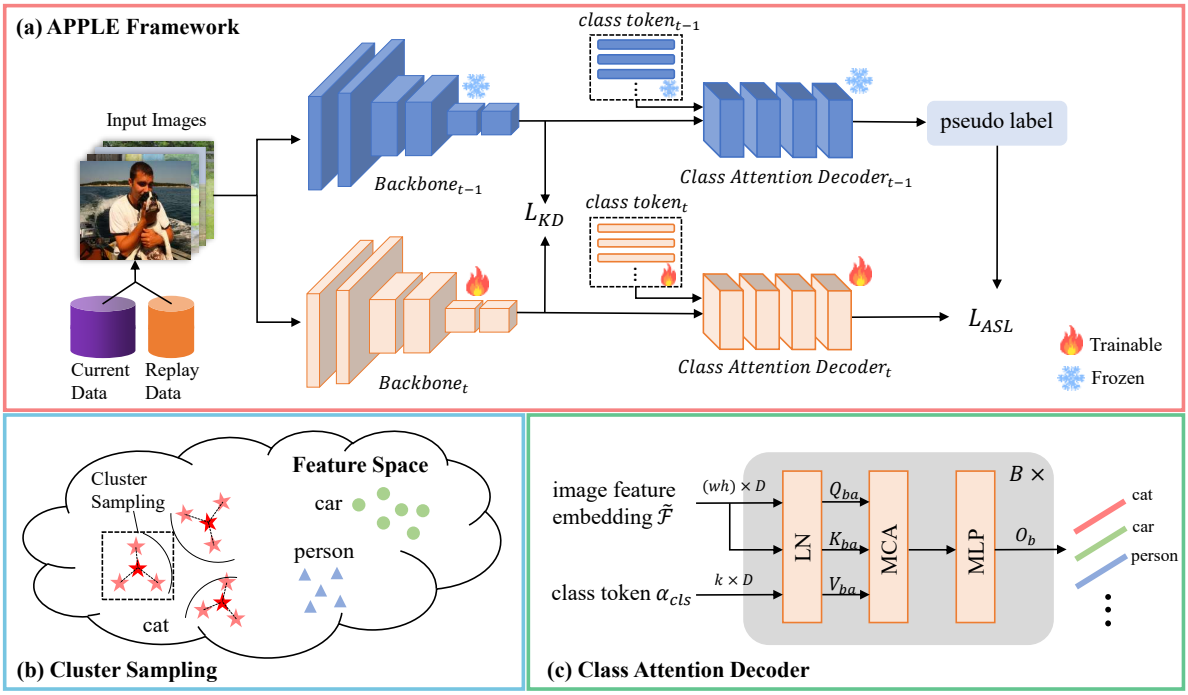


Figure 2. An overview of our proposed APPLE framework. (a) illustrates the training process of our method in session t. The model trained in the session $t-1$ is preserved to calculate knowledge distillation loss L_{KD} and produces pseudo-labels for samples of current session. The training samples are drawn from current training data and a small memory of replay data. (b) presents our proposed cluster sampling strategy. In the feature space, the features of the same class are close. We use K-means algorithm to split intra-class samples of each class, and then select central samples from each cluster, improving the diversity of selected samples. (c) shows the structure of class attention decoder (CAD), which has B blocks. Each block is composed of layer normalization (LN), multi-head cross attention (MCA), and multi-layer perceptron (MLP). CAD takes the embedded image features $\tilde{\mathcal{F}}$ as the key and value of MCA and regards learnable class tokens α_{cls} as the query.

3.3. Adaptive Pseudo-Label

We propose an adaptive pseudo-label strategy to increase the label information in the training data. In the MLCIL setting, the training data may include objects that belong to the previously learned classes but are not annotated in the current session. Hence, we use the model $f_{\theta_{t-1}}$ trained in the previous session to obtain the output probability for learned categories, thereby obtaining pseudo-labels to alleviate the impact of label absence problem. For an input sample $x \in X_t$, the output probability of model $f_{\theta_{t-1}}$ can be represented by $P = \{p_1, p_2, \cdots, p_K\}$, $p_k \in (0,1)$, where K is the number of categories learned by $f_{\theta_{t-1}}$. Assuming the pseudo-label set $\hat{Y}_t$, the pseudo-label $\hat{y} = \{\hat{y}_1, \hat{y}_2, \cdots, \hat{y}_K\} \in \hat{Y}_t$ can be obtained as follows:

$$\hat{y}_k = \begin{cases} 1, & \text{if } p_k > \epsilon \\ 0, & \text{if } p_k \leq \epsilon \end{cases} \tag{1}$$

where ϵ is a threshold hyper-parameter.

According to Eq. (1), the size of the pseudo-label set $\hat{Y}_t$ can be controlled by the threshold ϵ. If ϵ is small, too many pseudo-labels could affect the learning of new classes. By contrast, if ϵ is large, too few pseudo-labels lead to poor recognition accuracy for old classes. Therefore, we introduce an adaptive strategy to help determine the threshold rather than use a fixed number. Since the training data is randomly divided into different sessions, when the dataset is large, we can assume that the proportion of objects of the same category in different sessions is the same. As an example, in session $t-1$, the model has learned knowledge of K classes, there are L_{t-1} annotated labels, and the total number of samples is M_{t-1}. The average number of K class objects per image can be defined as $c_{t-1} = \frac{L_{t-1}}{M_{t-1}}$. Similarly, in each training epoch of session t, the average number of K class objects per image is $\tilde{c}_t$, $\tilde{c}_t \approx c_{t-1}$. Hence, we can adjust the threshold according to the relationship between $\tilde{c}_t$ and c_{t-1}:

$$\epsilon = \begin{cases} min(\epsilon_{max}, \epsilon + \Delta\epsilon), & \text{if } \tilde{c}_t > c_{t-1} \\ max(\epsilon_{min}, \epsilon - \Delta\epsilon), & \text{if } \tilde{c}_t \leq c_{t-1} \end{cases} \tag{2}$$

where $\Delta\epsilon$ represents the step size, ϵ_{max} and ϵ_{min} denote the maximum and minimum threshold, respectively. Finally, these pseudo-labels are added into the current label set for training.

3.4. Cluster Sampling

As a commonly used strategy in CIL, data replay [27] significantly alleviates catastrophic forgetting. In single-label CIL, some methods [17, 27] are proposed to take random or herding sampling as their sampling strategy. These selection methods are based on the feature of the whole image. However, in MLCIL setting, an image often corresponds to several objects, and the features of different objects are coupled in one feature map, which makes it difficult to select the most representative samples for each class. Thereby, we propose a new sampling strategy, namely cluster sampling, to improve the quality of selected samples. As shown in Fig. 2(b), we select samples based on the features of the objects. The size of the feature map output by class attention decoder is $K * D$, where K is the number of categories and D is the embedding dimension. According to the label set Y_t, we can get the $1 * D$ size feature corresponding to each object. Accordingly, for k-th class, the feature set $\mathbf{F}_t^k$ can be expressed as follows:

$$\mathbf{F}_t^k = \{\mathbf{f}_t^{ik} | \mathbf{f}_t^{ik} = f_{\theta_t}(x_t^i, y_t^{ik}; \theta_t), x_t^i \in X_t, y_t^{ik} \in Y_t\}, \quad (3)$$

where $\mathbf{f}_t^{ik}$ denotes the feature of the k-th class object of the i-th sample in session t, which can be obtained by the model f_{θ_t}, x_t^i represents the i-th sample in X_t, and y_t^{ik} is the label of the k-th class object of x_t^i.

Although the object features of the same class are close, they are still distinguishable in different dimensions, indicating intra-class differences. Thus, we divide the $\mathbf{F}_t^k$ into different clusters by the K-means [26] clustering method. For instance, in Fig. 2(b), the red stars represent the feature set of cats, which are divided into three clusters. It is obvious that samples in the same cluster have similar characteristics, and sampling from different clusters results in more diverse samples, so as to further enrich the feature set of cats. Here, we use the hyper-parameter m to determine the number of clusters. Through this cluster sampling strategy, we are able to select more diverse and representative samples.

3.5. Class Attention Decoder

Different from single-label classification, in multi-label classification, images often have several objects in different positions. If the global features are directly used for classification, small objects may be ignored, resulting in the feature dilution problem. Hence, we propose the class attention decoder (CAD), which can pay more attention to the location information of the object and extract local features adaptively. As illustrated in Fig. 2 (c), the CAD contains B blocks which have multi-head cross attention (MCA) modules and multi-layer perceptron (MLP) modules. The feature embeddings obtained by the feature extractor can be termed as $\mathcal{F} \in \mathbb{R}^{H \times W \times D}$, where H, W, and D represent the height, width, and embedding dimension of the feature map, respectively. By reshaping $\mathcal{F}$ to $(H \cdot W) \times D$ and adding position embedding we can get $\widetilde{\mathcal{F}}$. Then we use learnable class tokens $\alpha_{cls} \in \mathbb{R}^{K \times D}$ as query, where K is the number of categories, and take the feature embedding $\widetilde{\mathcal{F}}$ as the key and value. The class tokens α_{cls} are compared with $\widetilde{\mathcal{F}}$ at different spatial locations to generate attention maps, thereby extracting the object features adaptively. In the first block of CAD ($b = 1$), taking $\widetilde{\mathcal{F}}$ and α_{cls} as input, the output is shown as follows:

$$\begin{aligned}
Q_{ba} &= LN(\alpha_{cls}) \cdot W_{ba}^q, \\
K_{ba} &= LN(\widetilde{\mathcal{F}}) \cdot W_{ba}^k, \\
V_{ba} &= LN(\widetilde{\mathcal{F}}) \cdot W_{ba}^v, \\
\mathcal{H}_{ba} &= Softmax(\frac{Q_{ba} \cdot K_{ba}^T}{\sqrt{D/A}}) \cdot V_{ba}, \\
O_b &= MLP_b([\mathcal{H}_{b1}, \mathcal{H}_{b2}, \cdots, \mathcal{H}_{bA}] \cdot W_b^o),
\end{aligned} \quad (4)$$

where $LN(\cdot)$ means the layer normalization, and W_{ba}^q, W_{ba}^k and W_{ba}^v represent the weight matrices of the a-th attention head of the b-th block. A denotes the number of attention heads. By concatenating A attention heads and multiplying the weight matrix W_b^o, we can get the output O_b of block b. If $b > 1$, we only need to replace Q_{ba} in Eq. (4) with Eq. (5), and the others remain unchanged.

$$Q_{ba} = LN(O_{b-1}) \cdot W_{ba}^q. \quad (5)$$

Since in each block, the query for each class α_{cls} examines the feature embeddings $\widetilde{\mathcal{F}}$ and chooses the most relevant part to combine, the CAD could efficiently extract the spatial information in $\widetilde{\mathcal{F}}$, leading to better performance under multi-label circumstances.

3.6. Overall Objective

Apart from the above strategies, we are inspired by the current progress in class-incremental learning and multi-label classification field. To be specific, in order to overcome catastrophic forgetting, as illustrated in Fig. 2(a), we adopt knowledge distillation (KD) loss [15], which applies regularization on the feature space. Denoting the outputs of the backbone as o_t, the distillation loss can be defined as:

$$L_{KD} = \|o_{t-1} - o_t\|^2. \quad (6)$$

Additionally, we use asymmetric loss (ASL) [2] to address the positive-negative sample imbalance problem in multi-label classification tasks. Different γ values are used to manipulate the impact of positive and negative samples. Given a sample x, the model can predict its category probabilities $P = \{p_1, p_2, ..., p_K\} \in \mathbb{R}^K$, where there are totally K categories. Then the ASL loss can be defined as follows:

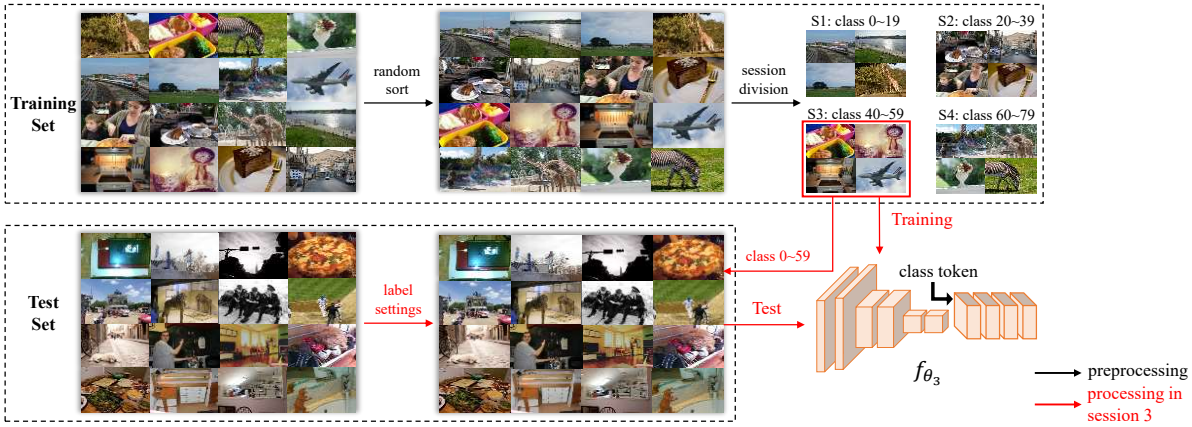


Figure 3. The division of the MS-COCO dataset under the $B0$-$C20$ protocol in the MLCIL setting. In this figure, S indicates session and the model is trained and tested in session 3. Each class label appears in just one session, even if classes overlap between sessions.

Table 1. Experimental results (mAP%) of our method and comparison CIL methods on MS-COCO dataset.

Methods	MS-COCO $B40$-$C10$				MS-COCO $B0$-$C20$			
	ResNet101		TResNet-L		ResNet101		TResNet-L	
	Last Acc	Avg Acc	Last Acc	Avg Acc	Last Acc	Avg Acc	Last Acc	Avg Acc
FT	9.94	31.88	11.12	35.83	21.44	48.54	23.60	51.87
JT	**85.88**	-	**86.49**	-	**85.88**	-	**86.49**	-
iCaRL [27]	64.28 (↓ 9.23)	75.10	65.60 (↓ 9.01)	76.69	62.00 (↓ 12.94)	74.52	64.54 (↓ 12.11)	76.53
ER [30]	38.34 (↓ 35.17)	56.16	64.05 (↓ 10.56)	72.30	45.53 (↓ 29.41)	58.32	50.14 (↓ 26.51)	63.59
TPCIL [35]	59.05 (↓ 14.46)	67.10	71.20 (↓ 3.41)	75.28	62.16 (↓ 12.78)	69.41	68.89 (↓ 7.76)	73.54
PODNet [10]	63.66 (↓ 9.85)	74.58	66.12 (↓ 8.49)	77.11	59.94 (↓ 15.00)	73.06	61.01 (↓ 15.64)	74.78
PASS [51]	59.07 (↓ 14.44)	72.32	59.44 (↓ 15.17)	73.80	54.91 (↓ 20.03)	74.03	49.88 (↓ 26.77)	72.16
DER++ [3]	52.68 (↓ 20.83)	58.56	55.77 (↓ 18.84)	66.71	62.60 (↓ 12.34)	69.41	67.33 (↓ 9.32)	73.82
APPLE(ours)	**73.51** (↓ 0.00)	**80.94**	**74.61** (↓ 0.00)	**82.05**	**74.94** (↓ 0.00)	**81.62**	**76.65** (↓ 0.00)	**83.49**

$$L_{ASL} = \frac{1}{K}\sum_{k=1}^{K}\begin{cases}(1-p_k)^{\gamma+}\log(p_k), & \text{if } y_k = 1\\ (p_k)^{\gamma-}\log(1-p_k), & \text{if } y_k = 0\end{cases} \quad (7)$$

where y_k is a binary label to indicate whether the sample x has label k or not. Thus, the overall objective of our framework can be stated as follows:

$$L = L_{ASL} + \lambda L_{KD}, \quad (8)$$

where λ is a hyper-parameter to balance these two terms.

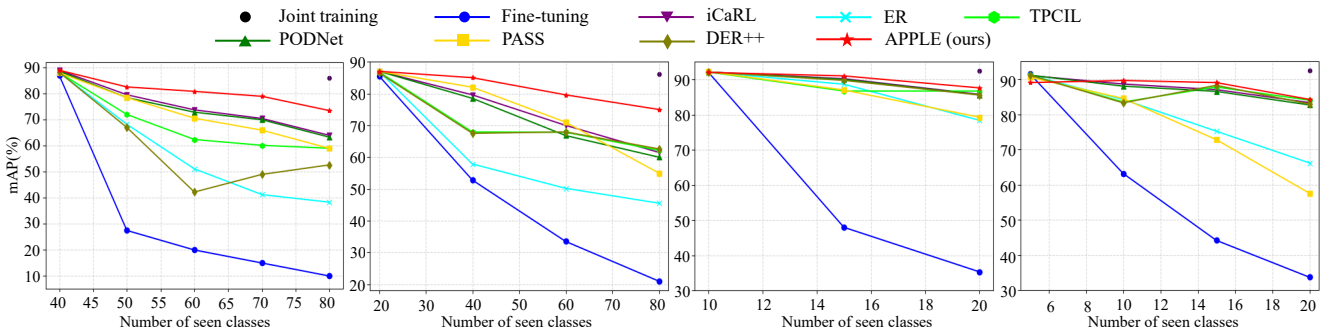
4. Experiments

4.1. Experiment Setup and Implementation Details

Datasets and Protocols. We conduct several experiments on MS-COCO [22] and PASCAL VOC 2007 [11] datasets to verify the effectiveness of our proposed method. MS-COCO dataset is widely used to evaluate multi-label image classification and we adopt the 2014 split. It consists of 122,218 images and has 80 categories of common objects, with an average of 2.9 labels per image. PASCAL VOC 2007 dataset is also a commonly used benchmark in multi-label classification. We conduct the incremental learning tasks on the `train-val` set with 5,000 images and then evaluate it on `test` set with 5,000 images.

We adopt the protocols which are commonly used in CIL [45]. The protocol could be represented by a unified terminology Bi-Cj, where i denotes the class number to be learned in the base session and j is the class number to be learned in each incremental session. We evaluate the models on MS-COCO dataset with $B40$-$C10$ and $B0$-$C20$ protocols and on PASCAL VOC 2007 dataset with $B10$-$C5$ and $B0$-$C5$ protocols.

To illustrate how we handle the dataset in the MLCIL setting, we show the division method of the MS-COCO dataset under the $B0$-$C20$ protocol in Fig. 3. The model is trained and tested in session 3 in this figure. For the training set, we first use seed 1998 to sort it randomly. Then, the number of training samples in each session is obtained by the ratio of current session's number of classes among the total classes. For example, the $B0$-$C20$ protocol learns 20 classes in each session, so we divide the training set into 4 parts on average. Finally, we specify the classes to be learned for each session. In this figure, the model learns classes $0 \sim 19$, $20 \sim 39$, $40 \sim 59$, and $60 \sim 79$ in order



(a) MS-COCO, B40-C10 (b) MS-COCO, B0-C20 (c) PASCAL VOC 2007, B10-C5 (d) PASCAL VOC 2007, B0-C5

Figure 4. Performance curves (mAP%) of the nine methods with respect to session $1 \sim T$ on MS-COCO dataset. Two different MLCIL protocols are adopted and the backbone is ResNet101.

Table 2. Experimental results (mAP%) of our method and comparison CIL methods on PASCAL VOC 2007 dataset.

Methods	PASCAL VOC 2007 $B10$-$C5$				PASCAL VOC 2007 $B0$-$C5$			
	ResNet101		TResNet-L		ResNet101		TResNet-L	
	Last Acc	Avg Acc	Last Acc	Avg Acc	Last Acc	Avg Acc	Last Acc	Avg Acc
FT	35.34	58.85	60.09	74.82	33.73	58.37	59.73	74.10
JT	**94.80**	-	**93.79**	-	**94.80**	-	**93.79**	-
iCaRL [27]	85.75 (↓ 1.84)	89.66	87.07 (↓ 2.29)	90.78	83.31 (↓ 0.91)	87.62	84.78 (↓ 0.84)	88.33
ER [30]	78.37 (↓ 9.22)	86.70	73.91 (↓ 15.45)	86.01	66.18 (↓ 18.04)	79.64	68.31 (↓ 17.31)	82.68
TPCIL [35]	86.70 (↓ 0.89)	88.84	84.18 (↓ 5.18)	90.19	83.99 (↓ 0.23)	86.24	79.38 (↓ 6.24)	87.95
PODNet [10]	85.61 (↓ 1.98)	89.64	86.35 (↓ 3.01)	90.35	82.65 (↓ 1.57)	86.89	84.12 (↓ 1.50)	87.78
PASS [51]	79.26 (↓ 8.33)	87.07	76.93 (↓ 12.43)	86.01	57.59 (↓ 26.63)	76.42	51.84 (↓ 33.78)	75.58
DER++ [3]	85.55 (↓ 2.04)	89.46	83.95 (↓ 5.41)	90.22	82.92 (↓ 1.30)	86.84	84.82 (↓ 0.80)	88.95
APPLE(ours)	**87.59** (↓0.00)	**90.27**	**89.36** (↓ 0.00)	**91.68**	**84.22** (↓ 0.00)	**87.83**	**85.62** (↓0.00)	**89.52**

during four sessions. When testing, we set the labels including the already learned classes to 1 (*i.e.*, class $0 \sim 59$), and set the other labels to 0.

Implementation Details. For adaptive pseudo-label, we set the initial threshold $\epsilon_{init} = 0.75$, the step size $\Delta\epsilon = 0.02$, the maximum threshold $\epsilon_{max} = 0.99$, and the minimum threshold $\epsilon_{min} = 0.01$. The number of replay samples is set to 20 per class for all datasets. We divide each class feature into 20 clusters by K-means and select the central feature of each cluster as the sample to be preserved. In the CAD module, we adopt 4 blocks ($B = 4$). The λ is set to 25 for KD loss L_{KD}. For the asymmetric loss L_{ASL}, the γ^+ and γ^- are set to 4 and 1, respectively. To verify the effectiveness of our proposed framework, we evaluate models using two backbones, including ResNet101 [14] and TResNet-L [28]. Furthermore, when training in the first session, we load the ImageNet [7] pretrained model, and when training in session $t(t > 1)$, we load the model trained in the previous session. We train the models for 40 epochs using Adam [18] optimizer, with true weight decay [25] of 1e-4, and the learning rate of 1e-4. All images are resized to 448×448 for training and testing.

Evaluation Metrics. We adopt the mean average precision (mAP) across all categories for evaluation. The model performance is calculated on those already learned classes. We report the mAP after the final session as 'Last Acc' and the average mAP among all sessions as 'Avg Acc'.

4.2. Quantitative Results

Comparison Methods. Tab. 1 and Tab. 2 summarize the results of our method and several comparison methods on MS-COCO and PASCAL VOC 2007. In incremental learning, fine-tuning (FT) and joint training (JT) are treated as the lower and upper bound, in which FT fine-tunes the model without any anti-forgetting constraints and JT conducts supervised training on all data. Moreover, we select six representative CIL methods to compare with our proposed method, including iCaRL [27], ER [30], TPCIL [35], PODNet [10], PASS [51] and DER++ [3], where iCaRL and ER are classical rehearsal-based methods, TPCIL, PODNet, and Der++ are best-performing rehearsal-based methods which save old samples to replay, and PASS is the best-performing rehearsal-based methods which save the prototype of each class to replay. We implement these methods in the MLCIL setting. For a fair comparison, we change their original cross-entropy loss to the ASL loss to address the multi-label classification problem. As for these methods of saving old samples, as in our method, we adopt 20 samples per class for replay.

Results on MS-COCO. Tab. 1 summarizes the Last and average (Avg) results of ResNet101/TResNet-L backbone under the $B40$-$C10$ and $B0$-$C20$ protocols on the MS-

COCO dataset. For the $B40$-$C10$ protocol, APPLE with the ResNet101 backbone achieves the last accuracy of **73.51%** and the average accuracy of **80.94%**, which both significantly surpass previous models and obtain the state-of-the-art performance. In comparison, the second-best method iCaRL achieves the last accuracy of 64.38% and the average accuracy of 75.10%. Our proposed method outperforms iCaRL by up to **9.23%** (**64.28%**→**73.51%**) for the 'Last Acc' and **5.84%** (**75.10%**→**80.94%**) for the 'Avg Acc'. For the TResNet-L backbone, APPLE outperforms the second-best TPCIL by **3.41%** (**71.20%**→**74.61%**) for the 'Last Acc' and **6.77%** (**75.28%**→**82.05%**) for the 'Avg Acc'. The experimental results above also prove the robustness of our method for different network structures. For the $B0$-$C20$ protocol, APPLE achieves the last accuracy of **74.94%** and **76.65%** on the ResNet101 and TResNet-L backbones, respectively, exceeding the other state-of-the-art methods in the MLCIL setting.

Fig. 4(a) and (b) show the comparison curves of the different methods with ResNet101 backbone under the $B40$-$C10$/$B0$-$C20$ protocol. We can observe that our proposed APPLE consistently outperforms other CIL methods in each session and have a minimal gap with upper bound JT. As incremental learning proceeds, the superiority of APPLE becomes more pronounced, which illustrates that our method is more resistant to catastrophic forgetting.

Results on PASCAL VOC 2007. Tab. 2 summarizes the last and average results of ResNet101/TResNet-L backbone under the $B10$-$C5$ and $B0$-$C5$ protocols on the PASCAL VOC 2007 dataset, which have similar trends to the results on MS-COCO. For the $B10$-$C5$ protocol, the last and average accuracy of APPLE which uses the ResNet101 backbone surpass the second-best method TPCIL with about **0.89%** and **1.43%**, respectively. For the TResNet-L backbone, the last and average accuracy of APPLE outperforms the second-best method PODNet by about **3.01%** and **1.33%**. For the $B0$-$C5$ protocol, APPLE also exceeds other competitive methods. Fig. 4(c) and (d) show the comparison curves of the different methods which use ResNet101 backbone under the $B10$-$C5$/$B0$-$C5$ protocol. It is observed that APPLE is always superior to contrast methods.

4.3. Ablation Study

To verify the effectiveness of each component in our proposed method, we conduct ablation experiments under the $B40$-$C10$ protocol on MS-COCO dataset and the results are listed in Tab. 3. In model 1, we simply finetune a ResNet101 model as a baseline. In model 2, we add the CAD module to the baseline and get a **6.01%** (**21.44%**→**27.45%**) relative improvement of 'Last Acc', which demonstrates that paying attention to the spatial information of object features improves the classification performance of the model in the MLCIL setting. Based on this model, in model 3, we further add the KD and cluster sampling strategy, boosting the final accuracy by **37.52%** (**27.45%**→**64.97%**). Then, in model 6, we add the adaptive pseudo-label (APL) strategy to model 3, which is equivalent to the proposed APPLE framework, resulting in a significant improvement of **9.97%** (**64.97%**→**74.94%**). For comparison, we replace KD+CS and APL in model 6 with KD+herding sampling (HS) and fixed-threshold pseudo-label strategy (FPL), respectively, obtaining model 4 and model 5. The 'Last Acc' of model 4 and model 5 are **1.79%** and **2.04%** lower than ours, respectively. These results above strongly prove that our proposed three components are very effective in preventing catastrophic forgetting and improving performance in the MLCIL setting.

Table 3. Ablation study of our method under the $B0$-$C20$ protocol on MS-COCO dataset with ResNet101 backbone. KD is the knowledge distillation, which is used together with cluster sampling (CS) or herding sampling (HS). APL represents the adaptive pseudo-label strategy and FPL is the fixed-threshold pseudo-label strategy ($\epsilon = 0.75$). † denotes the comparison modules.

	CAD	KD+CS	APL	KD+HS†	FPL†	Last Acc	Avg Acc
Model 1	×	×	×	×	×	21.44	48.54
Model 2	✓	×	×	×	×	27.45	54.31
Model 3	✓	✓	×	×	×	64.97	76.44
Model 4	✓	×	✓	✓	×	73.15	81.13
Model 5	✓	✓	×	×	✓	72.90	80.13
Model 6	✓	✓	✓	×	×	**74.94**	**81.62**

5. Conclusion

In this paper, instead of the widely studied CIL, we focus on a more general and challenging *multi-label class-incremental learning* (MLCIL) setting, where models are required to alleviate the catastrophic forgetting problem while learning new classes from multi-label samples. In order to solve the label absence, representative sample selection, and feature dilution problems it brings, we propose a new framework, termed APPLE, which contains three components. First, APPLE uses an adaptive pseudo-label strategy to generate pseudo-labels for currently available data, solving the label absence problem. Second, a clustering sampling strategy is proposed to obtain more representative replay samples, which can better mitigate catastrophic forgetting. Finally, a class attention decoder is designed to concentrate on the spatial information of object features, alleviating the feature dilution problem in the multi-label scenario. Extensive experiments on PASCAL VOC 2007 and MS-COCO datasets show that our method outperforms comparison CIL methods in the challenging MLCIL setting.

Acknowledgement

This work was funded by the National Key Research and Development Project of China under Grant No. 2020AAA0105600, and by the National Natural Science Foundation of China under Grant No. U21B2048 and No. 62006183.

References

[1] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European conference on computer vision (ECCV)*, pages 139–154, 2018. 2

[2] Emanuel Ben Baruch, T. Ridnik, Nadav Zamir, Asaf Noy, Itamar Friedman, Matan Protter, and Lihi Zelnik-Manor. Asymmetric loss for multi-label classification. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 82–91, 2021. 3, 5

[3] Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. *Advances in neural information processing systems*, 33:15920–15930, 2020. 6, 7

[4] Francisco Manuel Castro, Manuel J. Marín-Jiménez, Nicolás Guil Mata, Cordelia Schmid, and Alahari Karteek. End-to-end incremental learning. *ArXiv*, abs/1807.09536, 2018. 1, 2

[5] Tianshui Chen, Muxi Xu, Xiaolu Hui, Hefeng Wu, and Liang Lin. Learning semantic-specific graph representation for multi-label image recognition. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 522–531, 2019. 3

[6] Zhao-Min Chen, Xiu-Shen Wei, Peng Wang, and Yanwen Guo. Multi-label image recognition with graph convolutional networks. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5172–5181, 2019. 3

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, K. Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 7

[8] Prithviraj Dhar, Rajat Vikram Singh, Kuan-Chuan Peng, Ziyan Wu, and Rama Chellappa. Learning without memorizing. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5133–5141, 2019. 1

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ArXiv*, abs/2010.11929, 2021. 3

[10] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *ECCV*, 2020. 2, 6, 7

[11] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John M. Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88:303–338, 2009. 2, 6

[12] Chrisantha Fernando, Dylan S. Banarse, Charles Blundell, Yori Zwols, David R Ha, Andrei A. Rusu, Alexander Pritzel, and Daan Wierstra. Pathnet: Evolution channels gradient descent in super neural networks. *ArXiv*, abs/1701.08734, 2017. 3

[13] Bin-Bin Gao and Hong-Yu Zhou. Learning to discover multi-class attentional regions for multi-label image recognition. *IEEE Transactions on Image Processing*, 30:5920–5932, 2021. 3

[14] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 7

[15] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *ArXiv*, abs/1503.02531, 2015. 1, 2, 5

[16] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 3

[17] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 831–839, 2019. 1, 5

[18] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015. 7

[19] James Kirkpatrick, Razvan Pascanu, Neil C. Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114:3521 – 3526, 2017. 1, 2

[20] Xilai Li, Yingbo Zhou, Tianfu Wu, Richard Socher, and Caiming Xiong. Learn to grow: A continual structure learning framework for overcoming catastrophic forgetting. In *ICML*, 2019. 3

[21] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017. 2

[22] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 2, 6

[23] Shilong Liu, Lei Zhang, Xiao Yang, Hang Su, and Jun Zhu. Query2label: A simple transformer way to multi-label classification. *ArXiv*, abs/2107.10834, 2021. 3

[24] Yongcheng Liu, Lu Sheng, Jing Shao, Junjie Yan, Shiming Xiang, and Chunhong Pan. Multi-label image classification via knowledge distillation from weakly-supervised detection. In *Proceedings of the 26th ACM International Conference on Multimedia*, MM '18, page 700–708. Association for Computing Machinery, 2018. 3

[25] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 7

[26] J. MacQueen. Some methods for classification and analysis of multivariate observations. 1967. 5

[27] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, G. Sperl, and Christoph H. Lampert. icarl: Incremental classifier and representation learning. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5533–5542, 2017. 1, 3, 5, 6, 7

[28] T. Ridnik, Hussam Lawen, Asaf Noy, and Itamar Friedman. Tresnet: High performance gpu-dedicated architecture. *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1399–1408, 2021. 7

[29] T. Ridnik, Gilad Sharir, Avi Ben-Cohen, Emanuel Ben-Baruch, and Asaf Noy. Ml-decoder: Scalable and versatile classification head. 2021. 3

[30] Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and Gerald Tesauro. Learning to learn without forgetting by maximizing transfer and minimizing interference. *arXiv preprint arXiv:1810.11910*, 2018. 3, 6, 7

[31] Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *ArXiv*, abs/1606.04671, 2016. 3

[32] Jonathan Schwarz, Wojciech Czarnecki, Jelena Luketina, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. Progress & compress: A scalable framework for continual learning. In *International conference on machine learning*, pages 4528–4537. PMLR, 2018. 2

[33] Joan Serrà, Dídac Surís, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. In *ICML*, 2018. 3

[34] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. *Advances in neural information processing systems*, 30, 2017. 3

[35] Xiaoyu Tao, Xinyuan Chang, Xiaopeng Hong, Xing Wei, and Yihong Gong. Topology-preserving class-incremental learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIX 16*, pages 254–270. Springer, 2020. 3, 6, 7

[36] Gido M Van De Ven, Zhe Li, and Andreas S Tolias. Class-incremental learning with generative classifiers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3611–3620, 2021. 1, 3

[37] Ya Wang, Dongliang He, Fu Li, Xiang Long, Zhichao Zhou, Jinwen Ma, and Shilei Wen. Multi-label classification with label graph superimposing. In *AAAI*, 2020. 3

[38] Zhouxia Wang, Tianshui Chen, Guanbin Li, Ruijia Xu, and Liang Lin. Multi-label image recognition by recurrently discovering attentional regions. In *Proceedings of the IEEE international conference on computer vision*, pages 464–472, 2017. 3

[39] Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, et al. Dualprompt: Complementary prompting for rehearsal-free continual learning. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVI*, pages 631–648. Springer, 2022. 1, 3

[40] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 139–149, 2022. 1, 3

[41] Yunchao Wei, Wei Xia, Min Lin, Junshi Huang, Bingbing Ni, Jian Dong, Yao Zhao, and Shuicheng Yan. Hcp: A flexible cnn framework for multi-label image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38:1901–1907, 2016. 3

[42] Tong Wu, Qingqiu Huang, Ziwei Liu, Yu Wang, and Dahua Lin. Distribution-balanced loss for multi-label classification in long-tailed datasets. In *ECCV*, 2020. 3

[43] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Raymond Fu. Large scale incremental learning. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 374–382, 2019. 1

[44] Tianjun Xiao, Jiaxing Zhang, Kuiyuan Yang, Yuxin Peng, and Zheng Zhang. Error-driven incremental learning in deep convolutional neural network for large-scale image classification. In *Proceedings of the 22nd ACM International Conference on Multimedia*, MM '14, page 177–186. Association for Computing Machinery, 2014. 1

[45] Shipeng Yan, Jiangwei Xie, and Xuming He. Der: Dynamically expandable representation for class incremental learning. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3013–3022, 2021. 1, 3, 6

[46] Hao Yang, Joey Tianyi Zhou, Yu Zhang, Bin-Bin Gao, Jianxin Wu, and Jianfei Cai. Exploit bounding box annotations for multi-label object recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 280–288, 2016. 3

[47] Jin Ye, Junjun He, Xiaojiang Peng, Wenhao Wu, and Y. Qiao. Attention-driven dynamic graph convolutional network for multi-label image recognition. In *ECCV*, 2020. 3

[48] Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. *ArXiv*, abs/1708.01547, 2018. 3

[49] Bowen Zhao, Xi Xiao, Guojun Gan, Bin Zhang, and Shutao Xia. Maintaining discrimination and fairness in class incremental learning. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13205–13214, 2020. 1

[50] Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. Co-transport for class-incremental learning. In *Proceedings of the 29th ACM International Conference on Multimedia*, MM '21, page 1645–1654. Association for Computing Machinery, 2021. 1

[51] Fei Zhu, Xu-Yao Zhang, Chuan Wang, Fei Yin, and Cheng-Lin Liu. Prototype augmentation and self-supervision for incremental learning. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5867–5876, 2021. 1, 3, 6, 7