CKDF: Cascaded Knowledge Distillation Framework for Robust Incremental Learning

Kunchi Li, Jun Wan, *Senior Member, IEEE*, and Shan Yu

***Abstract*— Recently, owing to the superior performances, knowledge distillation-based (kd-based) methods with the exemplar rehearsal have been widely applied in class incremental learning (CIL). However, we discover that they suffer from the feature uncalibration problem, which is caused by directly transferring knowledge from the old model immediately to the new model when learning a new task. As the old model confuses the feature representations between the learned and new classes, the kd loss and the classification loss used in kd-based methods are heterogeneous. This is detrimental if we learn the existing knowledge from the old model directly in the way as in typical kd-based methods. To tackle this problem, the feature calibration network (FCN) is proposed, which is used to calibrate the existing knowledge to alleviate the feature representation confusion of the old model. In addition, to relieve the task-recency bias of FCN caused by the limited storage memory in CIL, we propose a novel image-feature hybrid sample rehearsal strategy to train FCN by splitting the memory budget to store the image-and-feature exemplars of the previous tasks. As feature embeddings of images have much lower-dimensions, this allows us to store more samples to train FCN. Based on these two improvements, we propose the Cascaded Knowledge Distillation Framework (CKDF) including three main stages. The first stage is used to train FCN to calibrate the existing knowledge of the old model. Then, the new model is trained simultaneously by transferring knowledge from the calibrated teacher model through the knowledge distillation strategy and learning new classes. Finally, after completing the new task learning, the feature exemplars of previous tasks are updated. Importantly, we demonstrate that the proposed CKDF is a general framework that can be applied to various kd-based methods. Experimental results show that our method achieves state-of-the-art performances on several CIL benchmarks.**

***Index Terms*— Feature calibration, hybrid exemplars, cascaded knowledge distillation, incremental learning.**

I. Introduction

MANY computer vision applications in the real world require the capability that can incrementally learn about new classes while preserving the existing knowledge. For example, for construction safety, a system that can identify whether a worker is wearing a safety vest or a hard hat is wished to add the ability to detect improper footwear [1]. However, most deep learning approaches suffer from catastrophic forgetting [2], [3] when the past data are unavailable.

To alleviate catastrophic forgetting, kd-based approaches with the exemplar rehearsal have been proposed [4]–[7]. As shown in Fig. 1, this kind of methods commonly have the same process of transferring knowledge from the old network immediately to the new model when learning new classes in CIL.

In this work, we focus on this kind of methods. To visualize the learning process, we compare kd-based approaches with the process of a child's incremental learning in Fig. 2. It shows the learning process by a child to recognize fruits incrementally.

When a child sees a cactus, he may learn simple knowledge, such as the appearance of shape and color of cactus (i.e., ellipsoidal, piliferous and green). After he knows the characteristics of cactus, we take away the cactus and show him a rambutan. At the first sight, the child might wrongly recognize the rambutan as a cactus because of the same characteristics he learned (e.g., ellipsoidal and piliferous). After one corrects him, he can compare the rambutan with the cactus and correct the existing knowledge through experience (feature) replay. The child then adjusts the relative importance of the features of cactus and would make feature calibration in

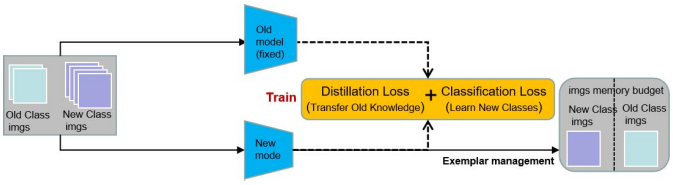


Fig. 1. The overview of widely used basic knowledge distillation-based framework [4]–[7].

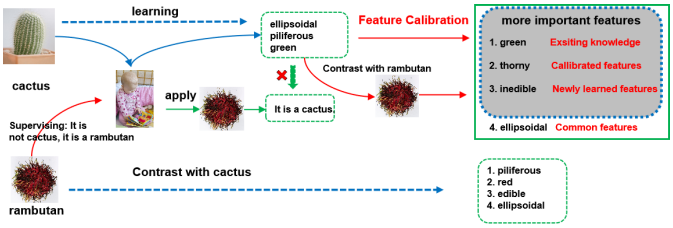


Fig. 2. It illustrates how to recognize fruits incrementally (e.g., learn to recognize cactus and rambutan orderly) by a child.

Manuscript received October 22, 2021; revised February 26, 2022 and April 16, 2022; accepted May 4, 2022. Date of publication May 24, 2022; date of current version June 2, 2022. This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFA0105203, in part by the Science and Technology Development Fund of Macau under Project 0070/2020/AMJ, in part by the National Natural Science Foundation of China under Project U21A20515, in part by the Strategic Priority Research Program of the Chinese Academy of Sciences (CAS) under Grant XDB32040200, in part by the International Partnership Program of CAS under Grant 173211KYSB20200021, in part by the Beijing Academy of Artificial Intelligence (BAAI), and in part by the Open Research Projects of Zhejiang Laboratory under Grant 2021PF0AB01. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Ran He. *(Corresponding author: Shan Yu.)*

The authors are with the National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing 100190, China, and also with the School of Artificial Intelligence, University of Chinese Academy of Sciences (UCAS), Beijing 100190, China (e-mail: shan.yu@nlpr.ia.ac.cn).

Digital Object Identifier 10.1109/TIP.2022.3176130

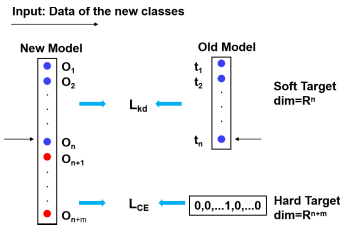


Fig. 3. The kd loss in kd-based methods encourages the new model to mimic the output of the old model while the CE loss computed by using the new data encourages the output of the new model at the positions of learned classes to zero.

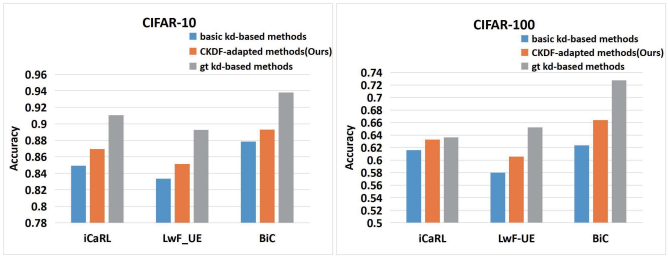


Fig. 4. The comparison of various kd-based methods using different models as the old model in Fig. 1 on CIFAR-10 and CIFAR-100. Each experiment has 2 incremental learning batches. CKDF-adapted methods are the methods which adapt the basic kd-based methods by our approach (CKDF). LwF-UE (**LwF U**sing **E**xemplars) is an extended version of LwF [1] which improves LwF by using exemplars of old classes in this work.

his brain to deepen understanding of cactus. For example, cactus is thorny but not piliferous and compared with the shape (ellipsoidal), the features of thorny and color are more important to distinguish cactus from rambutan. And the child also learns a new distinctive feature that cactus is inedible. By the comparison with the calibrated knowledge of cactus, he will also acquire a more comprehensive knowledge of rambutan, such as piliferous, red, edible and ellipsoidal.

As the above example illustrates, the child does not simply preserve the fixed existing knowledge but calibrates the existing knowledge via adjusting the relative importance of cactus features and extracting new characteristics from cactus to solve the conflict between the old knowledge of cactus and the knowledge of rambutan before he learns to recognize rambutan. On the contrary, typical kd-based methods try to preserve the old knowledge of the previous task model by minimizing the knowledge distillation loss (kd loss) to encourage the new model to mimic the output of the old model [1], [4]–[7]. However, this feature learning strategy in typical kd-based methods suffers from some problems. For example, typical kd-based methods would suffer from the task-recency bias caused by the data imbalance problem in CIL [5], [8], [9]. Besides, Ahn *et al.* [10] validate that the kd loss computed by using the old model directly in typical kd-based methods preserves this bias and reduces the performances of kd-based methods. In this work, we discover that typical kd-based methods would suffer from a new problem, which is referred to as the feature uncalibration problem.

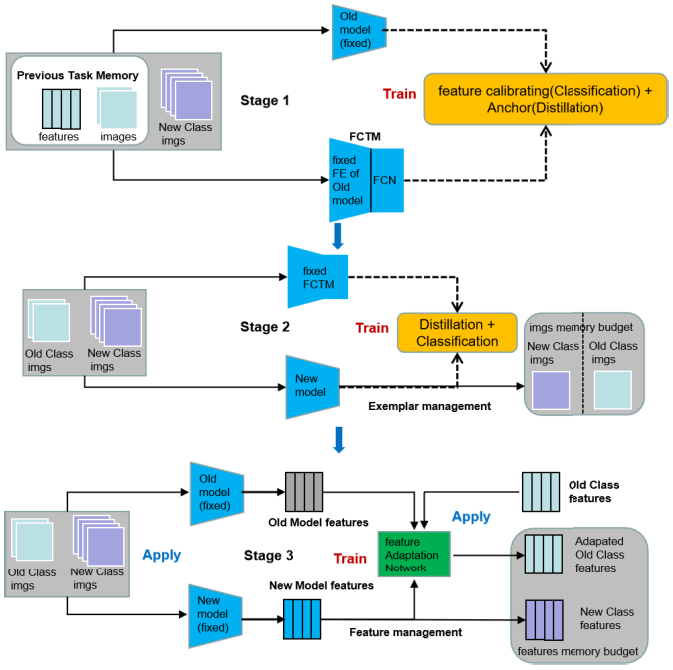


Fig. 5. Overview for the new model training of our approach employing the image-feature hybrid storage strategy. For displaying our approach in detail, we give details of the feature adaptation algorithm [7] at stage 3.

A. Feature Uncalibration Problem

As the original training data of old classes and the data of new classes are usually out-of-distribution (OOD) in CIL, the old model confuses the feature representations between the old classes and the new classes, and the confusion problem may become more serious when there are similar classes between the old and new tasks (cf. Sec. V). This is detrimental to kd-based methods. First, the kd loss in typical kd-based methods encourages the preservation of the old model's confused decision boundary in the new model (encourages the new model to mimic the output of the old model). Moreover, because of the confusion problem, the kd loss and the classification cross-entropy (CE) loss, which are computed by using the new data are heterogeneous (Typical kd-based methods use data of the new classes to compute the kd loss and the CE loss, cf. Fig. 3). Typical kd-based methods ignore this problem and compute the kd loss at each training iteration by directly using the old model with the confused decision boundary, so they suffer from the confusion problem (cf. Fig. 4 and Sec. V). Overall, we refer to the above phenomenon as the feature uncalibration problem in this work.

Fig. 4 briefly shows the feature uncalibration problem in typical kd-based methods. As shown in Fig. 4, **gt** kd-based methods that use the **g**round-**t**ruth model as the old model in Fig 1 improve basic kd-based methods by large margins. That is because the ground-truth model is trained with all training data from all tasks in CIL and it has no feature uncalibration problem. So our main goal in this work is to obtain a feature calibrated teacher model which is ideally without the feature uncalibration problem since the ground-truth model is unavailable in the CIL setting.

From the above observations, we believe that it is important to deal with the feature uncalibration problem. Therefore, in this paper, a feature calibration network to calibrate the existing knowledge of the old model to separate the feature representations between the previous and current tasks is proposed. Furthermore, owing to the limited memory budget, the task-recency bias [8] usually occurs in the image-exemplar rehearsal stage. Inspired by the child incremental learning through experience (feature) replay, we propose an image-feature hybrid sample rehearsal strategy to train the FCN. The main advantage of this strategy is that a large number of samples from previously learned tasks can be stored and used.

Based on the above improvements, we propose CKDF with three stages to train the new model when learning the new task in CIL. Fig. 5 is the overview framework of the proposed CKDF. At stage 1, we train the FCN to calibrate the existing knowledge of the old model to relieve the feature uncalibration problem. At stage 2, we generate the **f**eature **c**alibrated **t**eacher **m**odel (**FCTM**) by attaching FCN to the fixed feature extractor (FE) of the old model. And we transfer knowledge from FCTM instead of the old model when learning the new classes. At last, we adapt the old feature exemplars to be in the right feature space of the new model through feature adaptation [7]. Both stages 1 and 2 utilize the kd technique to train the models. And stage 2 is very similar to typical kd-based methods but we use FCTM instead of the old model to make knowledge transferring. Compared with typical kd-based methods, our approach has additional two stages (stages 1 and 3) in each new task and we store feature exemplars to train FCN. From Fig. 4, we can see that the proposed CKDF improves the performances of three basic kd-based methods (iCaRL [4], BiC [5] and LwF-UE, an extended version of LwF [1] in this work) by [1.66%, 4.02%] and [1.46%, 2.03%] in the term of average accuracy [11] on the experiments with 2 incremental learning batches on CIFAR-10 and CIFAR-100, respectively. Besides, these results also reflect that the feature uncalibration problem exists in typical kd-based methods.

We note that CKDF is a general framework for kd-based methods with the exemplar rehearsal. It can be easily used to adapt and improve typical kd-based methods by training FCN and transferring knowledge from FCTM instead of the old model in training the new model while keeping other processes of the methods unchanged.

In summary, the main contributions of the work include:

1) We discover that typical kd-based approaches suffer from the feature uncalibration problem. To tackle the problem, we propose the FCN to calibrate the existing knowledge of the old model to separate the feature representations between the previous and current tasks. To the best of our knowledge, it is the first solution to deal with this problem in CIL.
2) To relieve the task-recency bias in FCN, we propose an image-feature hybrid exemplar rehearsal strategy. It can keep the lower-dimensional feature embeddings of images to reduce the memory footprint significantly. So we can store more samples to train an effective FCN.
3) To combine FCN and the image-feature hybrid exemplar rehearsal strategy, we propose a simple, effective and general framework (CKDF) that can be easily extended to various kd-based methods.
4) Experiments demonstrate that the performances of three kd-based methods have been improved significantly when combined with the proposed CKDF and achieved state-of-the-art performances on several CIL benchmarks.

II. Related Work

A. Continual Learning and Catastrophic Forgetting

CIL belongs to Continual Learning [12] and the core problem is catastrophic forgetting. Many methods aiming to solve the problem have been proposed in different task settings. Generally, these methods can be divided into three categories [13].

1) Regularization Approaches: The first is regularization approaches [14]–[20]. This family introduces an extra regularization term in the loss function, to consolidate previous knowledge when learning on new data. However, regularization approaches are not designed specifically for CIL and can not generalize well on convolution networks [21].

2) Parameter Isolation Approaches: The second category is parameter isolation approaches [22]–[27]. This family distributes different model parameters to each task preventing any possible forgetting. Generally, the model in these works will become bigger and bigger with learning new tasks incrementally. Moreover, the approaches require a task oracle, activating corresponding masks or task branches during prediction, so parameter isolation approaches are applicable to tasks incremental learning but not CIL.

3) Replay Approaches: The third one is replay approaches [4]–[7], [9], [28]–[33], they store samples in raw format or generate pseudo-samples with a generative model. These previous task samples are replayed while learning a new task to alleviate forgetting. The previous replay methods usually make exemplar management by a random or herding [4] strategy. Recently, Liu *et al.* [34] propose a dynamic memory management strategy that is optimized for the incremental phases and different object classes, which improves several replay methods obviously. Most previous works do not use all the samples as the model inputs to train the new classifier network immediately but utilize the kd technique [4]–[7], [9], [30] or the manifold learning technique [28], [31], [33] to constrain optimization of the new task loss to prevent the task interference. Liu *et al.* recently propose AANets [35] including stable blocks and plastic blocks, which can learn automatically to trade off stability and plasticity [36] during training the new model in an end-to-end way to improve replay methods significantly. Especially, Tao *et al.* [37] propose a novel topology-preserving method that constructs and grows the elastic Hebbian graph restrictively to preserve the feature space topology of old classes to alleviate forgetting when learning new classes, and they extend the approach to tackle the few-shot class-incremental learning problem (FSCIL) in [38]. Inspired by

the above works, Liu *et al.* [39] and Dong *et al.* [40] develop the approaches preserving the structural knowledge of old classes for CIL and FSCIL, respectively. But all of these works did not address the feature uncalibration problem. Our approach belongs to replay methods for CIL and we focus on the kd-based methods. Different from the previous works, we focus on relieving the feature uncalibration problem to improve the performance of kd-based methods. There are also some works that fix the FE or the first part of the convolution layers of FE and train other parts of the model [41], [42] while the performances of these approaches are limited.

B. Knowledge Distillation-Based Methods

LwF [1] is the first work to introduce the kd technique into CIL. LwF does not store exemplars but just uses data of the current task to make knowledge transferring. After that, most kd-based methods combine the kd technique with the exemplar rehearsal [4]–[7]. Most of them perform well and achieve the state-of-the-art performances. Here, we summarize the popular strategies for training the network and handling catastrophic forgetting.

1) Problem Formulation: We are given a dataset $\mathcal{D} = \{(x, y)|x \in \mathcal{X}, y \in \mathcal{Y}\}$ where $\mathcal{X}$ is a set of images with labels $\mathcal{Y}$ belonging to the classes in $\mathcal{C}$. In the CIL setting, $\mathcal{C}$ is split to T subsets $\mathcal{C}^1, \mathcal{C}^2, \ldots, \mathcal{C}^T$, where $\mathcal{C} = \mathcal{C}^1 \cup \mathcal{C}^2 \cup \cdots \cup \mathcal{C}^T$ and $\mathcal{C}^i \cap \mathcal{C}^j = \emptyset$ for $i \neq j$. We denote $\mathcal{D}^t$ as the dataset of class set $\mathcal{C}^t$, $\mathcal{X}^t = \{x|(x, y) \in \mathcal{D}^t\}$ and $\mathcal{Y}^t = \{y|(x, y) \in \mathcal{D}^t\}$ as the new training images and labels of the new classes at task t. We denote $\hat{E}^t$ as all the stored image exemplars of the old classes at task t for experience replay methods and $\hat{E}^t = \emptyset$ when methods do not store image exemplars. We denote $\hat{\mathcal{D}}^t = \mathcal{D}^t \cup \hat{E}^t$, $\hat{\mathcal{X}}^t = \{x|(x, y) \in \hat{\mathcal{D}}^t\}$, $\hat{\mathcal{Y}}^t = \{y|(x, y) \in \hat{\mathcal{D}}^t\}$ as all the observable dataset, all the available images and labels at task t, respectively. At task t, we have a model $f^{t-1}_{\theta,W}$ which has incrementally learned the old classes $\hat{\mathcal{C}}^{t-1} = \{\mathcal{C}^1, \mathcal{C}^2, \ldots, \mathcal{C}^{t-1}\}$. Now, observing the new classes $\mathcal{C}^t$, the goal is to train a new model $f^t_{\theta,W}$ that can perform classification on all the classes $\hat{\mathcal{C}}^t$ with the dataset $\hat{\mathcal{D}}^t$.

2) Training Strategy: Most previous works of kd-based methods have the common process that trains the new model by minimizing two losses: the CE loss and the kd loss. We denote the learned classifier typically a *convolutional neural network* by $f_{\theta,W} : \mathcal{X} \to \mathbb{R}^N$, where N is the number of learned classes. The learned classifier can be divided into two components denoted by h_θ and g_W which are the *feature extractor* (FE) and the linear classifier, respectively, where θ, W are the parameters. The CE loss of classification (L_{CE}) is typically computed as follows:

$$L_{CE} = \sum_{(x,y)\in\hat{\mathcal{D}}^t} \sum_{i=1}^{m+n} -\delta_i(x) log[\sigma_i(f^t_{\theta,W}(x))] \tag{1}$$

where $\delta_i(x)$ is the label indicator function, m, n are the number of learned and new classes respectively and σ is either the $softmax$ or $sigmoid$ function. The kd loss (L_{KD}) is used to encourage the new network $f^t_{\theta,W}$ to mimic the output of the previous task model $f^{t-1}_{\theta,W}$. It is typically computed as follows:

$$L_{KD} = \sum_{x\in\hat{\mathcal{X}}^t} \sum_{i=1}^{m} -\sigma_i(f^{t-1}_{\theta,W}(x)) log[\sigma_i(f^t_{\theta,W}(x))]. \tag{2}$$

So the new model $f^t_{\theta,W}$ is trained by the overall loss:

$$L = \lambda_1 L_{KD} + \lambda_2 L_{CE} \tag{3}$$

where λ_1, λ_2 are hyper parameters. Note that $f^t_{\theta,W}$ is continually updated at task t, whereas the network $f^{t-1}_{\theta,W}$ is frozen and will not be stored after the completion of task t. We use the same denotations for the rest of this paper.

After training the new model, previous works have different subsequent processing. For example, iCaRL [4] takes the nearest-mean-of-exemplars (NME) classification strategy to classify all the observed classes. Some previous works further correct the task-recency bias of a part of (e.g., the linear classifier) or the whole of the new model with a small validation dataset, such as BiC [5], EEIL [6]. Most previous works ignore the feature uncalibration problem and use the previous task model to make knowledge transferring directly. Different from the previous works, we do not learn the existing knowledge from $f^{t-1}_{\theta,W}$ but from FCTM and we utilize an image-feature hybrid storage strategy to train FCN.

III. Our Approach

A. The Overall Design

Learning knowledge of previous tasks from the ground-truth model instead of the old model in Fig. 1 in training the new model improves basic kd-based methods significantly (cf. Fig. 4 and Section V). Therefore, a key point of our approach is to train a feature calibrated teacher model (FCTM) with a much relieved feature uncalibration problem, since the ground truth model is unavailable in the CIL setting. So, we put forward feature calibration for the old model and propose CKDF to train the new model.

As demonstrated in Fig. 5, our approach for training the new model is a cascaded knowledge distillation framework and has three stages in training each new task except task 1 (The first task is a typical classification task). Stage 1 is used to train FCN. We use all the available data including the stored images and features of the old classes and the images of the new classes to train an FCN to alleviate the feature uncalibration problem. Stage 2 is used to train the new model. After training FCN, we obtain FCTM composed of FCN and the frozen FE of the old model. Then, we replace the old model in Fig. 1 with FCTM to train the new model. Finally, the feature-exemplars of the old classes are updated in stage 3. After completing a new task, the new model changes and the feature exemplars of the previous tasks are out-of-date. We apply feature adaptation [7] to adapt the old feature exemplars to be in the right feature space of the new model.

B. Stage 1: Training FCN

To obtain a feature calibrated teacher model with a less feature uncalibration problem to be used in training, we design FCN with two objectives: (1) FCN can generate new features

of an old class that are distinct from the new classes. It can adjust the relative importance of the extracted features of the old class which are used to distinguish it from other classes especially the new ones. Therefore, it can mitigate the feature confusion of the old model and the conflict between the existing knowledge of the old model and the knowledge of the new classes. (2) FCN also has the ability to preserve the knowledge of the old model, which is still important to distinguish among different old classes.

In the CIL settings, the data of future classes are unavailable, but we can observe the newly coming classes when learning the new task. So we can use the data of the newly coming classes and the exemplars of old classes to calibrate the existing knowledge of the old model.

We use the features extracted by the old model and the feature exemplars as the input for training FCN. In this work, we focus on image classification tasks. FCN is a shallow fully connecting network (2-3 hidden layers) and we use the multi-classes CE loss to achieve the feature calibration. We denote $\mathcal{V}^t$ and $\hat{\mathcal{V}}^t$ as the selected features of the new classes to be stored and the feature exemplars from all the previous tasks at task t, respectively. The feature calibrating loss (L_{FC}) is computed as follows:

$$L_{FC} = \sum_{(x,y)\in\hat{\mathcal{D}}^t} \sum_{i=1}^{m+n} -\delta_i(x) log\sigma_i(f^t_{FCN}(h^{t-1}_\theta(x))) + \sum_{(v,y)\in\hat{\mathcal{V}}^t} \sum_{j=1}^{m} -\delta_j(v) log\sigma_j(f^t_{FCN}(v)) \quad (4)$$

where f^t_{FCN} is FCN to be trained at task t, σ is the $softmax$ function, h^{t-1}_θ is the $feature\ extractor$ (FE) of $f^{t-1}_{\theta,W}$ which is fixed. The rest denotation is the same as Section II-B. Because the kd loss in typical kd-based methods encourages the preservation of the old model's confused decision boundary in the new model and typical kd-based methods ignore this problem and so suffer from the feature uncalibration problem. Through Equation 4, we use the multi-classes cross-entropy (CE) loss function computed by all observed data to calibrate the decision boundary of the old model. Moreover, the image-feature hybrid sample rehearsal strategy alleviates the task-recency bias of FCN caused by the data imbalance problem when training FCN. So after training FCN, the feature uncalibration problem is relieved effectively, which helps to improve the performances of kd-based methods.

It is worth noting that although we use the multi-classes CE loss function to train FCN, we do not aim to learn new knowledge to distinguish new classes but focus on adjusting the existing knowledge to relieve the conflict between the knowledge of the old model and the knowledge of the new classes (Objective 1). In section IV-F, we conduct experiments with a different feature calibrating loss which sees all the new classes as one super-class, and it also outperforms the baselines.

To achieve the second objective, FCN must preserve the information in the old model that is important for the classification among old classes. To this end, we use the original old model as an anchor to prevent the calibrated knowledge from drifting far away from the old knowledge. We utilize the kd technique to compute the anchor drifting loss (L_{AD}) as follows:

$$\begin{aligned} L_{AD} = & \sum_{(x,y)\in\hat{\mathcal{D}}^t} -\sum_{y=1}^{m} \hat{q}_y(x)\log q_y(x) \\ & + (1-\hat{q}_y(x))\log(1-q_y(x)) \\ & + \sum_{(v,y)\in\hat{\mathcal{V}}^t} -\sum_{y=1}^{m} \hat{p}_y(v)\log p_y(v) \\ & + (1-\hat{p}_y(v))\log(1-p_y(v)) \end{aligned}$$

$$\begin{aligned} \hat{q}_y(x) &= \frac{1}{1+exp(-\frac{(f^{t-1}_{\theta,W}(x))_y}{T_1})} \\ q_y(x) &= \frac{1}{1+exp(-\frac{(f^t_{FCN}(h^{t-1}_\theta(x)))_y}{T_2})} \\ \hat{p}_y(v) &= \frac{1}{1+exp(-\frac{(g^{t-1}_W(v))_y}{T_1})} \\ p_y(v) &= \frac{1}{1+exp(-\frac{(f^t_{FCN}(v))_y}{T_2})} \end{aligned} \quad (5)$$

where g^{t-1}_W is the frozen linear classifier of the original teacher model $f^{t-1}_{\theta,W}$ at task $t-1$, T_1 and T_2 are temperatures. If $T_1 = T_2$, L_{AD} is a typical loss of knowledge distillation. Hinton *et al.* [43] suggest that higher temperatures will increase the weight of smaller logit values and encourage the network to better encode similarities among classes. $T_1 > T_2$ is equivalent to a relaxed anchor (the soft target computed from the output of the old model). In this work, we employ a relaxed anchor on most experiments. L_{AD} is added to the feature calibrating loss, resulting in the overall loss function for training FCN as:

$$L_{FCN} = L_{FC} + \lambda L_{AD} \quad (6)$$

where λ is the hyper-parameters.

C. Stage 2: Training New Model

After training FCN, we obtain FCTM with a less feature uncalibration problem which attaches FCN to the frozen FE of the previous task model. Our approach (CKDF) can be applied to adapt typical kd-based methods as long as we replace the previous task model with FCTM to train the new model. Our main method (**CKDF-iCaRL**) is adapted from **iCaRL** by **CKDF**. The CE loss (L_{CE}) and the kd loss (L_{KD}) are computed as follows:

$$\begin{aligned} L_{CE} &= -\sum_{(x_i,y_i)\in\hat{\mathcal{D}}^t} \sum_{y=1}^{m+n} \delta_{y=y_i}(x_i)\log\sigma_y(f^t_{\theta,W}(x_i)) \\ &= (\delta_{y\neq y_i}(x_i))\log(1-\sigma_y(f^t_{\theta,W}(x_i))) \end{aligned} \quad (7)$$

$$\begin{aligned} L_{KD} = & -\sum_{(x_i,y_i)\in\hat{\mathcal{D}}^t} \sum_{y=1}^{m} \sigma_y(f^t_{FCTM}(x_i))\log\sigma_y(f^t_{\theta,W}(x_i)) \\ & + (1-\sigma_y(f^t_{FCTM}(x_i)))\log(1-\sigma_y(f^t_{\theta,W}(x_i))) \end{aligned} \quad (8)$$

where σ is the sigmoid function, $f^t_{FCTM}(x_i) = f^t_{FCN}(h^{t-1}_\theta(x_i))$ is the feature calibrated teacher model which attaches the trained FCN (f^t_{FCN}) to the FE (h^{t-1}_θ) of the previous task model ($f^{t-1}_{\theta,W}$) at task t and it is fixed when training the new model. The overall loss function for training the new model is:

$$L_{overall} = L_{CE} + L_{KD} \tag{9}$$

At stage 2, we use all available images including the stored image exemplars of old classes and the images of the current task to train $f^t_{\theta,W}$ without using the stored feature exemplars. This is because the stored feature exemplars and the calibrated feature exemplars, generated by mapping the stored feature exemplars to the calibrated feature space by FCN, are not in the right feature space of $f^t_{\theta,W}$. We split the memory budget to store image and feature exemplars for training FCN and the new model. CKDF employing the image-feature hybrid storage performs better than using the pure image or feature storage strategy (cf. Section IV-G.1).

D. Stage 3: Updating Feature Exemplars

At stage 1, we use the stored feature exemplars to train FCN as shown in Fig. 5. The stored feature exemplars $\mathcal{V}^t$ of the new classes extracted by $f^t_{\theta,W}$ at task t can be immediately used in training the next task $t+1$. But $\hat{\mathcal{V}}^t$ is unsuitable to be used at task $t+1$, because $\hat{\mathcal{V}}^t$ is not in the same feature space as $\mathcal{V}^t$. We utilize **f**eature **a**daptation (FA) [7] to update $\hat{\mathcal{V}}^t$ to generate new feature exemplars set $\hat{\mathcal{V}}^{t+1} = \hat{\mathcal{V}}^t_{FA} \cup \mathcal{V}^t$ when completing task t. $\hat{\mathcal{V}}^t_{FA}$ is the adapted features that are generated by mapping $\hat{\mathcal{V}}^t$ to the feature space of $\mathcal{V}^t$ through feature adaptation. After adapting all the old feature exemplars, all features of previous tasks are in the same feature space as $\mathcal{V}^t$ [7].

After training the new model, CKDF-iCaRL does not use the new model directly but employs the NME strategy to make inference like in iCaRL.

IV. Experiments

In this section, we first give details for benchmark datasets and experimental setup. Then, we conduct several experiments to demonstrate the effectiveness of the proposed CKDF. In addition, we also prove the generalization ability of the proposed method by extending it to three basic kd-based methods. Finally, we analyze the importance of each component contained in CKDF and show the effect of the loss function employed in our method. Our code is available at https://github.com/CSTiger77/CKDF.

A. Datasets and Exemplar Management

We conduct experiments on CIFAR-10, CIFAR-100 [44] and randomly choose 50 classes from ImageNet ILSVRC 2012 (referred to ImageNet-50) [45] to validate our approach. We pad 4 pixels for images in CIFAR-10 and CIFAR-100 and then random crop them into 32 × 32 pixels. Images in ImageNet-50 are first resized to 256 × 256, then random cropped to 224 × 224 pixels. We use horizontal and vertical flips in all image pre-processings.

The memory budget is fixed to the size of storing 2000 images [4]–[7] for all experiments in this work and it is divided into two parts for the image-feature hybrid methods: one is for storing image exemplars and the other is for storing feature exemplars. In this work, the split ratio is set to 90 : 10, i.e. 90% of memory budget is for image exemplars and 10% for feature exemplars. Although the size of memory for the feature exemplars storage is much smaller than that for the image exemplars storage, we still store a large number of feature exemplars because the feature of an image has a much lower dimension. For example, we can store 1800 (90% of 2000) image exemplars and 58800 (10% × 2000 × 294) feature exemplars on ImageNet-50. We follow the exemplar management of iCaRL to select image and feature exemplars, where the image-exemplars and the feature-exemplars are not overlapping.

B. Baselines and Metrics

We choose two state-of-the-art kd-based methods with the exemplars rehearsal (iCaRL [4], BiC [5]) as baselines. We also use an extended version of LwF [1], namely **LwF-UE** (LwF **u**sing **e**xemplars) as a compared method. LwF doesn't apply any experience replay technique. In this work, we focus on kd-based methods with the exemplar rehearsal, so we adapt LwF by introducing the exemplar rehearsal to it. The adapted method (LwF-UE) outperforms LwF significantly, and we use it instead of LwF as a baseline for a fair comparison.

We use a common metric in the continual learning literature: average accuracy [11] for overall performances to validate our approach.

C. Implementation Details

We use ResNet18 (d = 512) for ImageNet-50 and ResNet34 (d = 512) [46] for CIFAR-10 and CIFAR-100. When training networks, we follow the standard practices for fine-tuning existing networks. The training details of the three stages are as follows:

1) Training FCN: During training FCN, we train FCN by oversampling the image exemplars and feature exemplars of old classes. FCN is a 4-layers (3 hidden layers) multi-layer perceptron (MLP) with ReLU activation. FCN uses the features extracted by the previous task model or the feature exemplars (d = 512) as the input. The first two layers are hidden layers with dimensions $d' = 16d$. The output of the penultimate layer is the calibrated features (d = 512) and the last layer of FCN is the linear classifier. We use the same set of hyper-parameters and utilize Adam for training FCN in all experiments. The number of training epochs is 60. The learning rate is set to 0.001 and reduced to 1/5 of the previous rate after 20, 40, 55 epochs. The weight decay and the momentum are set to 0.0001 and 0.9, respectively. The batch size is 128. T_1 and T_2 in Equation 5 are set to 2, 1. λ in Equation 6 is set to the square of T_1 of Equation 5.

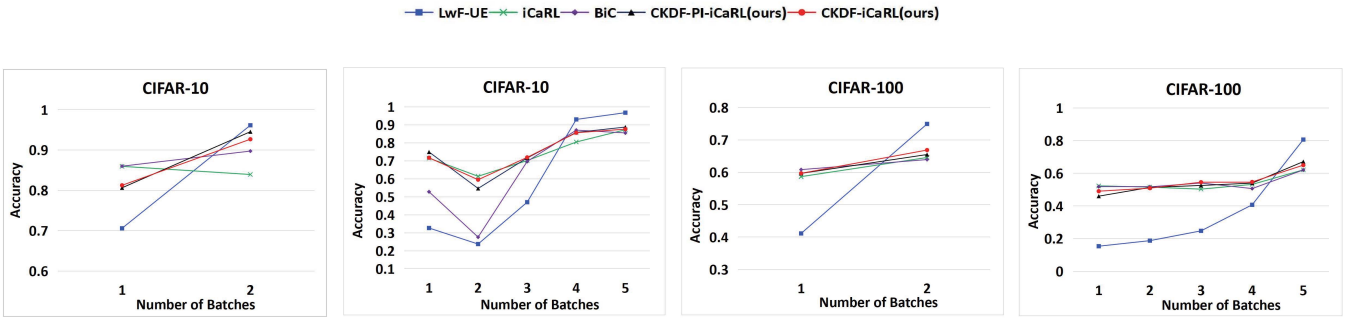


Fig. 6. The accuracy over classes of each incremental batch at the final stage of the experiments with 5 (a), 2 (b) classes per batch on CIFAR-10 and 50 (c), 20 (d) classes per batch on CIFAR-100.

TABLE I
COMPARISON ON CIFAR-10 (ACCURACY %)

Classes	5	10	2	4	6	8	10
upperbound	/	95.08	/	/	/	/	95.08
LwF-UE	95.32	83.35	98.70	88.77	77.03	72.58	58.62
iCaRL	93.56	84.92	98.30	88.90	80.33	77.76	74.15
BiC	95.32	**87.85**	98.70	86.35	75.77	72.76	64.42
CKDF-PI-iCaRL (ours)	93.26	87.57	98.30	**89.72**	**81.78**	77.75	75.14
CKDF-iCaRL (ours)	93.20	86.95	98.30	88.67	80.92	**78.3**	**75.26**

The left result is for the 2 incremental batches experiment that is 5 classes per batch and the right is for the 5 incremental batches experiment with 2 classes per batch.

TABLE II
COMPARISON ON CIFAR-100 (ACCURACY %)

Classes	50	100	20	40	60	80	100
upperbound	/	74.05	/	/	/	/	74.05
LwF-UE	76.24	58.02	78.4	62.78	51.13	44.4	36.04
iCaRL	73.15	61.61	73.65	65.87	61.50	56.60	53.89
BiC	76.58	62.37	78.4	66.55	61.38	56.5	54.06
CKDF-PI-iCaRL (ours)	73.06	62.57	73.65	**67.3**	61.85	58.08	54.21
CKDF-iCaRL (ours)	73.34	**63.27**	73.9	66.5	**62.27**	**58.18**	**54.84**

2) Training New Model: We use stochastic gradient descent (SGD) to train the new model with different hyper-parameters on different datasets. The training details are listed as follows:

CIFAR-10 We train the network for 120 epochs with 128 batch size, 0.00001 weight decay and 0.9 momentum at each task. The learning rate is set to 0.1 and reduced by a factor of 5 at 30, 60, 90, 100, 110 epochs.

CIFAR-100 The number of training epochs is 90 at each task. The learning rate is set to 1 and reduced the learning rate by a factor of 5 at epochs 50, 64 and 81. The weight decay and the momentum are set to 0.00001 and 0.9, respectively. The batch size is 128.

ImageNet-50 We train the network for 90 epochs with 128 batch size at each task. The weight decay and the momentum are 0.0005 and 0.9. The learning rate is set to 0.1 and reduced to $1/5$ of the previous rate after 20, 40, 60, 70, 80 epochs.

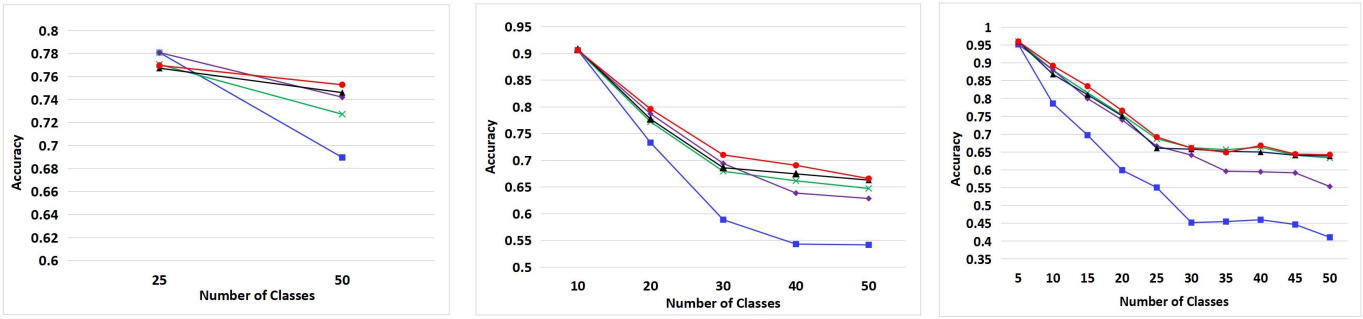
3) Updating Feature Exemplars: During updating feature exemplars, we train a feature adaptation network (FAN) with all the available images at the current task. The structure of FAN is the same as FCN. When training FAN, we copy FCN and fine-tune it. We use the same set of hyper-parameters to train FAN as training FCN.

D. Main Experiments

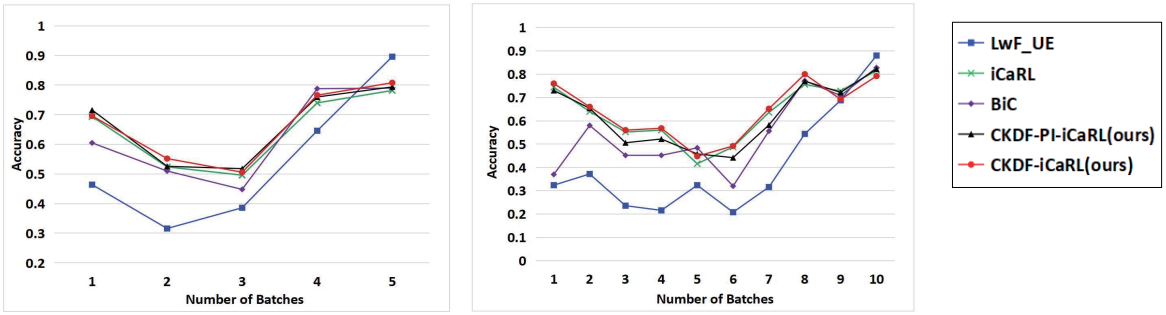
1) Evaluation on CIFAR-10 and CIFAR-100: We make experiments on CIFAR-10 and CIFAR-100, each of them is split into 2 and 5 incremental batches. Tables I and II show the results of these experiments, where CKDF-iCaRL is our main method employing the image-feature hybrid storage strategy and CKDF-PI-iCaRL is adapted from iCaRL by **CKDF** with the **p**ure **i**mage storage.

As shown in the tables, the performance of LwF-UE drops quickly. BiC that improves LwF-UE by utilizing the data rebalance to correct the bias of linear classifier of the new model is one of the state-of-the-art methods. Both our methods significantly outperform this baseline by about $[0.2\%, 11\%]$ over all experiments except for the 2 incremental batches on CIFAR-10. As a Not-End-to-End method, iCaRL that combines the kd technique with the NME strategy is another one of the state-of-the-art methods. Our methods outperform this strong baseline at nearly every stage. CKDF-iCaRL outperforms iCaRL at the final stage of all experiments by about $[0.95\%, 2\%]$ and CKDF-PI-iCaRL outperforms iCaRL by about $[0.32\%, 2.6\%]$. Besides, CKDF-iCaRL that applies the image-feature hybrid storage performs better than CKDF-PI-iCaRL with the pure image storage strategy. Fig 6 shows a comparison of the average accuracy over each incremental batch at the final stage of the experiments between our methods and the baselines. The number of batches in which our methods surpass the baselines is the same or more than those we lose, so our methods show better performances than the baselines.

2) Evaluation on ImageNet-50: 50 classes of ImageNet-50 are split into 2, 5, 10 incremental batches. All the average accuracies at the final stage are shown in Table III. Our method CKDF-iCaRL outperforms BiC over the three experiments by about 1.28%, 3.8% and 8.9%, respectively. The results of CKDF-iCaRL are above iCaRL by 2.5%, 1.8% and 0.9% on the three experiments. CKDF-PI-iCaRL also performs better than the baselines. Fig. 7a shows that the accuracy of



(a) Multi-class accuracy on ImageNet-50 with 25, 10, 5 classes per batch



(b) The accuracy over classes of each batch at the final stage of experiments with 10, 5 classes per batch on ImageNet-50.

Fig. 7. Experimental results of class-incremental training on ImageNet-50:(a) Our methods outperform the baselines at nearly every stage on ImageNet-50. (b) The accuracy over classes of each batch at the final stage of experiments with 10, 5 classes per batch on ImageNet-50.

TABLE III

COMPARISON ON IMAGENET-50 (ACCURACY %)

Incremental Batches	2	5	10
upperbound		82.46	
LwF-UE	68.96	54.16	41.08
iCaRL	72.72	64.72	63.36
BiC	74.20	62.84	55.32
CKDF-PI-iCaRL (ours)	74.60	66.28	63.88
CKDF-iCaRL (ours)	**75.28**	**66.56**	**64.24**

CKDF-iCaRL at nearly every stage is above the baselines. Fig. 7b demonstrates that CKDF-iCaRL performs better over nearly all the incremental batches than the baselines at the final stage of the experiments.

From the main experiments, we can see that our approach using the feature-image hybrid storage strategy (CKDF-iCaRL) performs better than using the pure image storage strategy (CKDF-PI-iCaRL) and both of them achieve the state-of-the-art average accuracies on CIFAR-10, CIFAR-100 and ImageNet-50.

E. Extended Experiments

To test the generalization ability of CKDF, we use CKDF to adapt three types of kd-based methods: LwF-UE, iCaRL and BiC. LwF-UE is an end-to-end method, while iCaRL is not end-to-end and BiC further corrects the task-recency bias of the linear classifier with an additional bias layer, which is trained on a small validation dataset through data balance techniques (i.e., under-sampling or over-sampling) after training the new model. We denote CKDF-LwF-UE, CKDF-BiC as the adapted methods of LwF-UE, BiC by CKDF, respectively. We conduct experiments on CIFAR-10, CIFAR-100 and

TABLE IV

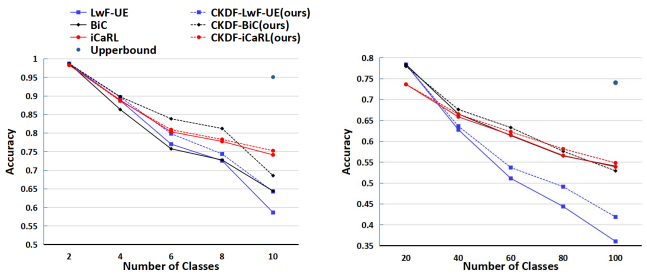
INCREMENTAL LEARNING RESULTS (AVERAGE ACCURACY % ON THE FINAL STAGE) OF EACH INCREMENTAL BATCH ON EXTENDED EXPERIMENTS WITH 5, 2 CLASSES PER BATCH ON CIFAR-10 AND 50, 20 CLASSES PER BATCH ON CIFAR-100

/	CIFAR-10		CIFAR-100	
Incremental Batches	2	5	2	5
LwF-UE	83.35	58.62	58.02	36.04
CKDF-LwF-UE(ours)	**85.12**	**64.28**	**60.55**	**41.89**
BiC	87.85	64.42	62.37	**54.06**
CKDF-BiC(ours)	**89.31**	**68.96**	**66.39**	52.93
iCaRL	84.92	74.15	61.61	53.89
CKDF-iCaRL(ours)	**86.95**	**75.26**	**63.27**	**54.84**

TABLE V

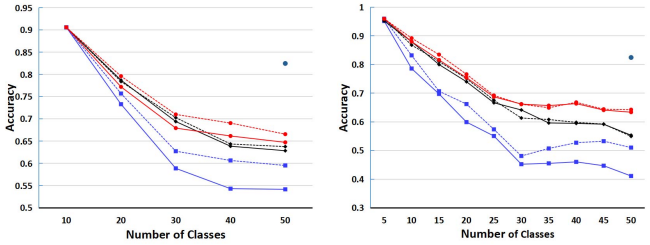
INCREMENTAL LEARNING RESULTS (AVERAGE ACCURACY % ON THE FINAL STAGE) OF EXTENDED EXPERIMENTS WITH 25, 10 AND 5 CLASSES PER BATCH ON IMAGENET-50

Incremental Batches	2	5	10
LwF-UE	68.96	54.16	41.08
CKDF-LwF-UE(ours)	**69.36**	**59.52**	**51.00**
BiC	74.20	62.84	**55.32**
CKDF-BiC(ours)	**76.16**	**63.76**	54.92
iCaRL	72.72	64.72	63.36
CKDF-iCaRL(ours)	**75.28**	**66.56**	**64.24**



(a) Extended experiments with 2 classes per batch on CIFAR-10

(b) Extended experiments with 20 classes per batch on CIFAR-100



(c) Extended experiments with 10, 5 classes per batch on ImageNet-50

Fig. 8. The trend of multi-class average accuracy of each incremental learning stage on extended experiments on CIFAR-10, CIFAR-100 and ImageNet-50.

ImageNet-50. Our main method CKDF-iCaRL is the adapted method of iCaRL and we have conducted experiments and shown the results in the main experiments. Here, we show the results once again together with the results of CKDF-LwF-UE and CKDF-BiC to validate that CKDF improves various types of kd-based methods with the exemplar rehearsal.

Table IV shows the comparison between the baselines and the corresponding adapted methods on CIFAR-10 and CIFAR-100. We find that our approach improves the baselines with different degrees. Specifically, CKDF-LwF-UE improves LwF-UE by more than 5% on the experiments with 2 classes per batch on CIFAR-10 and 20 classes per batch on CIFAR-100. CKDF-iCaRL outperforms iCaRL by $[0.95\%, 2.03\%]$. CKDF-BiC performs better than BiC by about $[1.5\%, 4.5\%]$ on CIFAR-10 and on the experiment with 2 incremental batches on CIFAR-100 but loses on the experiment of CIFAR-100 with 5 incremental batches.

Table V shows the similar results that the adapted methods outperform the baselines on ImageNet-50. Especially, our approach improves LwF-UE sizeable by about 10% on the experiment with 5 classes per batch on ImageNet-50. Fig. 8 shows the trend of multi-class average accuracy at each incremental learning stage. The adapted methods surpass the corresponding baselines nearly on every stage of the experiments. The results of these experiments demonstrate that our approach can be extended to various kd-based methods and improve them.

F. Alternative Experiments

When training FCN, we assume that adjusting the old knowledge to distinguish the old classes from the new classes is more important than learning new knowledge of new classes for classifying among them. To validate this assumption,

TABLE VI

INCREMENTAL LEARNING RESULTS (ACCURACY %) OF THE EXPERIMENTS WITH 2 CLASSES PER BATCH ON CIFAR-10

iCaRL	98.30	88.90	80.33	77.76	74.15
CKDF-OSC-iCaRL(ours)	98.35	**89.45**	80.83	77.6	75.16
CKDF-iCaRL(ours)	98.30	88.67	**80.92**	**78.3**	**75.26**
LwF-UE	98.70	88.77	77.03	72.58	58.62
CKDF-OSC-LwF-UE(ours)	98.7	89.32	78.72	74.12	63.5
CKDF-LwF-UE(ours)	98.5	**89.72**	**79.85**	**74.39**	**64.28**

TABLE VII

INCREMENTAL LEARNING RESULTS (ACCURACY %) OF THE EXPERIMENTS WITH 20 CLASSES PER BATCH ON CIFAR-100

iCaRL	73.65	65.87	61.5	56.6	53.89
CKDF-OSC-iCaRL(ours)	74.15	**66.87**	61.35	57.46	54.19
CKDF-iCaRL(ours)	73.65	66.5	**62.27**	**58.18**	**54.84**
LwF-UE	78.40	62.78	51.13	44.4	36.04
CKDF-OSC-LwF-UE(ours)	78.40	**63.97**	53.48	48.84	41.56
CKDF-LwF-UE(ours)	78.40	63.63	**53.72**	**49.15**	**41.89**

we conduct experiments with a new feature calibrating loss which sees all the new classes of the new task as one superclass. The new feature calibrating loss is computed as follows:

$$L_{FC} = \sum_{(x,y)\in\hat{\mathcal{D}}^t} \sum_{i=1}^{m+1} -\delta_i(x) log\sigma_i(f^t_{FCN}(h^{t-1}_\theta(x))) + \sum_{(v,y)\in\hat{\mathcal{V}}^t} \sum_{j=1}^{m} -\delta_j(v) log\sigma_j(f^t_{FCN}(v)) \quad (10)$$

where m and 1 of $m+1$ are the number of old classes and new classes, respectively. We adapt iCaRL and LwF-UE by our approach with the new feature calibrating loss, namely with a prefix CKDF-OSC- (CKDF-OneSuperClass-) and conduct experiments with 5 incremental batches on CIFAR-10 and CIFAR-100. The results of the experiments are shown in Tables VI and VII.

From Tables VI and VII, we can see that CKDF-OSC-iCaRL, CKDF-OSC-LwF-UE are inferior to CKDF-iCaRL and CKDF-LwF-UE but perform better than the original baselines. Especially, CKDF-OSC-LwF-UE improves LwF-UE by large margins: 4.9% on CIFAR-10 experiments and 5.5% on CIFAR-100 experiments. These results show that even though FCN just learns to distinguish old classes from new classes but not learns to classify new classes, CKDF-OSC-LwF-UE and CKDF-OSC-iCaRL still improve the corresponding baselines.

G. Ablation Study

1) The Sensitivity to Split of Memory Budget: In CKDF, we store and use the feature exemplars to train FCN. We also conduct ablation experiments without the feature exemplars. We conduct experiments through CKDF-PI-iCaRL which is adapted from iCaRL by our approach with the pure image storage strategy on CIFAR-10, CIFAR-100 and ImageNet-50. The results are shown in Tables I, II and III and in Fig. 6, 7. From these results, we find that CKDF with the pure image

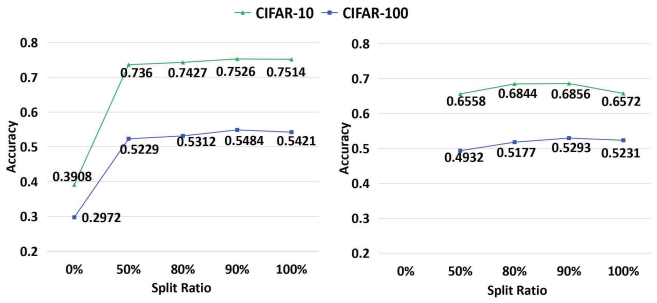


Fig. 9. Results on the split of memory budget experiments. The left is for CKDF-split-iCaRL which adapts iCaRL by our approach with various split ratios of memory for image exemplar storage, the right is for CKDF-split-BiC.

TABLE VIII

INCREMENTAL LEARNING RESULTS (ACCURACY %) OF THE ABLATION EXPERIMENTS WITH 2 CLASSES PER BATCH ON CIFAR-10

iCaRL	98.30	88.90	80.33	77.76	74.15
NoAnchor-iCaRL(ours)	98.30	**90.42**	**81.70**	**79.10**	74.05
CKDF-NOS-iCaRL(ours)	98.35	89.12	79.67	77.07	74.79
CKDF-iCaRL(ours)	98.30	88.67	80.92	78.3	**75.26**
LwF-UE	98.70	88.77	77.03	72.58	58.62
NoAnchor-LwF-UE(ours)	98.70	89.75	80.26	73.71	63.83
CKDF-NOS-LwF-UE(ours)	98.7	**89.87**	**80.27**	**76.63**	**65.74**
CKDF-LwF-UE(ours)	98.5	89.72	79.85	74.39	64.28

storage strategy without using feature exemplars is still effective and improves kd-based methods.

At the stage of training the new model, we just use all available images to train the new model without using the stored feature exemplars. So we must trade off the memory budget designated for the feature exemplars and the image exemplars. We conduct experiments with 5 incremental batches on CIFAR-10 and CIFAR-100 to analyze the sensitivity of our approach to the split ratio of memory budget. We select 0%, 50%, 80%, 90% and 100% of the memory budget to store image exemplars to conduct experiments. We use the prefix CKDF-split- to denote the adapted methods which adapt the baselines through CKDF with various split ratios of the memory budget. 0% of the memory budget to store image exemplars is equal to the pure feature storage and 100% is corresponding to the pure image storage strategy. We use the end-to-end output of the new model for inference on experiments of CKDF-split-iCaRL at 0%. BiC needs a validation set of exemplars of old classes to correct the bias of the new model, so we begin the split ratio from 50% on CKDF-split-BiC experiments. The results are shown in Fig. 9.

From the results, we can see that our approach is sensitive to the split ratio of memory budget between the feature exemplars and the image exemplars. And it seems that the average accuracy rises first and then falls along with the increasing of the split ratio of memory budget for the image exemplar storage and may obtain the best results with a ratio between 90% to 100% in our experiments.

2) Other Ablation Experiments: To analyze the function of the anchor drifting loss, we conduct ablation experiments which compare CKDF-iCaRL and CKDF-LwF-UE with their corresponding ablated methods without the anchor drifting loss, namely NoAnchor-iCaRL and NoAnchor-LwF-UE. In training FCN, we oversample the exemplars of old classes. Here, we validate our approach through ablation experiments without any oversampling, namely with the prefix CKDF-NOS-(**CKDF**-**N**o **O**ver **S**ampling).

TABLE IX

INCREMENTAL LEARNING RESULTS (ACCURACY %) OF THE ABLATION EXPERIMENTS WITH 20 CLASSES PER BATCH ON CIFAR-100

iCaRL	73.65	65.87	61.5	56.6	53.89
NoAnchor-iCaRL(ours)	74.15	64.1	56.83	50.47	45.45
CKDF-NOS-iCaRL(ours)	74.15	66.17	60.85	57.94	54.36
CKDF-iCaRL(ours)	73.65	**66.5**	**62.27**	**58.18**	**54.84**
LwF-UE	78.40	62.78	51.13	44.4	36.04
NoAnchor-LwF-UE(ours)	78.4	61.02	49.01	42.51	36.46
CKDF-NOS-LwF-UE(ours)	78.40	**64.15**	**54.62**	**49.88**	**42.45**
CKDF-LwF-UE(ours)	78.40	63.63	53.72	49.15	41.89

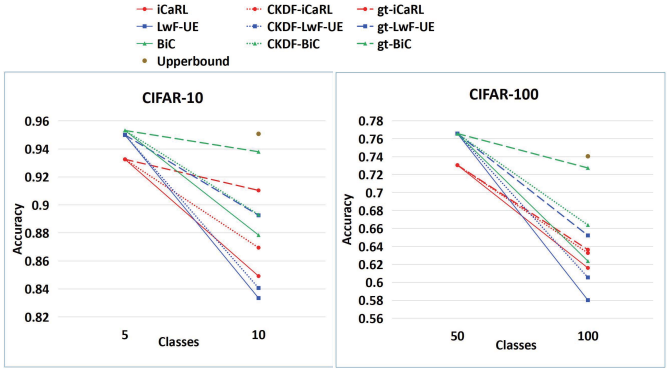


Fig. 10. Comparison on kd-based methods using different models as the old model in Fig. 1. Each experiment has 2 incremental learning batches. The methods named with the prefix CKDF are our approaches.

Table VIII demonstrates the performances of the ablated methods without the anchor drifting loss function. The ablated methods perform better at the beginning but drop quickly, resulting in relatively poor results finally on CIFAR-10, compared with CKDF-iCaRL and CKDF-LwF-UE. Table IX similarly demonstrates that the ablated methods are inferior to CKDF-iCaRL and CKDF-LwF-UE on CIFAR-100. The ablated methods without the anchor drifting loss perform worse than the original baselines as shown in the two tables. These results suggest that the anchor drifting loss function is important for our approach.

The results of ablation experiments without oversampling the exemplars of the old classes are also shown in Tables VIII and IX. The two tables show that CKDF-NOS-iCaRL without oversampling is inferior to CKDF-iCaRL while CKDF-NOS-LwF-UE performs better than CKDF-LwF-UE. Both the ablated methods without oversampling outperform the baselines on CIFAR-10 and CIFAR-100 by the margins of 0.47% and 6.41%, respectively. These results strongly suggest that the effectiveness of CKDF is relied on its own mechanism but not oversampling. At the same time, the data rebalance technique may improve our approach to some extent.

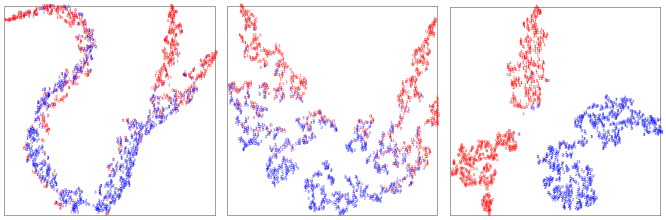


Fig. 11. Visualization experiments. The left is the visualization of features extracted by the old model, the middle is the visualization of calibrated features and the right is for the ground-truth model. The red samples are the first batch (old classes), the blue are the second batch (new classes), the digits in the figure are the labels.

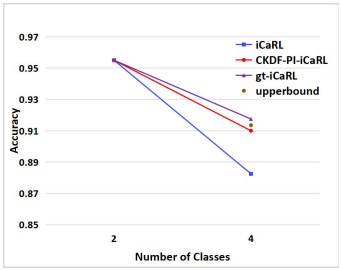


Fig. 12. The multi-class average accuracy on the subset with 4 classes selected from CIFAR-100 with 2 incremental learning batches.

V. Discussion

A. Diagnosis: Feature Uncalibration Problem

Our approach is based on the observation that typical kd-based methods suffer from the feature uncalibration problem. The feature uncalibration problem refers to the phenomenon that the feature confusion of the old model between the learned and new classes is detrimental to transferring the existing knowledge from the old model to the new model when learning new classes in the way as in typical kd-based methods.

We conduct experiments to demonstrate this phenomenon: First, we pre-train the ground-truth model using all the data including all the classes in CIL. Then, we conduct three groups of control experiments, the first group is to transfer knowledge from the old model to the new model according to the standard process of typical kd-based methods. The second group uses our approach to train the new model. The third group replaces the old model in Fig. 1 with the ground-truth model in training the new model. The management of memory budget is the same as described in Section IV. Each experiment has 2 incremental learning steps. The results are shown in Fig. 4 and 10.

As shown in Fig. 10, gt-iCaRL, gt-LwF-UE and gt-BiC using the **g**round-**t**ruth model, which is without the feature uncalibration problem outperform the original baselines by very large margins. For example, gt-BiC gets the accuracy: 0.9380, 0.7275 on experiments with 5 classes per batch on CIFAR-10 and with 50 classes per batch CIFAR-100, respectively, better than the corresponding results of original BiC: 0.8785, 0.6237. The results also show that our approach using FCTM loses to the methods using the ground-truth model yet still outperforms the baselines.

The results demonstrate that typical kd-based methods suffer from some unknown problems and the ground-truth model can greatly relieve these problems and improve their performances. To analyze the problems, considering that the ground-truth model can make inference on all the classes well, we assume that the previous task model trained with the data of old classes but without observing new classes confuses the new classes with the learned classes, therefore, the existing knowledge of the previous task model may conflict with the new knowledge of new classes.

To validate our assumption, we conduct visualization experiments including two groups of control experiments. We select a small subset with 4 classes from CIFAR-100, denoted as $\mathcal{D}$. We divide $\mathcal{D}$ into two incremental learning batches $\mathcal{D} = \{\mathcal{D}^1, \mathcal{D}^2\}$, where $\mathcal{D}^1 \cap \mathcal{D}^2 = \varnothing$. Then we train models according to two approaches: iCaRL, gt-iCaRL. The memory budget is fixed to 2000. At the second incremental learning step, we extract two groups of features of the training datasets, the first is using the feature extractor h_θ^1 of the previous task model which is trained by the dataset $\mathcal{D}^1$, the second group is using the ground-truth model trained by the dataset $\mathcal{D}$. Then we visualize the features through t-SNE [47] in Fig. 11. From Fig. 11, we can see that the features extracted by the original old model has the most mutual intrusion between the learned classes (the first batch classes) and the new classes (the second batch classes).

B. Analysis on FCN

From Fig. 4, 10 and section IV, we can see that CKDF is effective indeed. But why our approach is effective? Here, we assume the reason why the approach that transfers knowledge from the ground-truth model or FCTM is effective is that the ground-truth model and FCN relieve the feature uncalibration problem and alleviate the conflict between the existing knowledge of the old classes and the knowledge of the new classes.

We conduct visualization experiments to validate this assumption. We add a control experiment which is conducted according to CKDF-PI-iCaRL in the same setting of the experiments in Fig. 11. We first extract features of all the classes by h_θ^1 then map the features to the calibrated feature space to obtain the calibrated features by f_{FCN}^2 at the second incremental learning step. Then we visualize the calibrated features through t-SNE. From the visualization (Fig. 11), we can see that the ground-truth model has scarcely any mutual intrusion between the features of old classes (red samples) and new classes (blue samples), while the original old model has the most mutual intrusion and FCN is at the middle ground. With these three group features, we train three linear classifiers, the training accuracy rates are 72.35%, 74.22% and 98.44% respectively and the test accuracy rates are 22.25%, 31.75% and 62.75%. Ground-truth model and FCTM alleviate the mutual intrusion really and as Fig. 12 shows, the approaches using the ground-truth model and FCTM outperform iCaRL. Interestingly, gt-iCaRL using the ground-truth model outperforms the ground-truth model itself (the upper bound).

C. Conclusion and Future Work

In this work, we discover that the previous kd-based methods suffer from the feature uncalibration problem in CIL and the ground-truth model can greatly relieve the problem and improve the kd-based methods significantly. Based on this observation, we put forward the feature calibration for the old model and propose CKDF to realize the feature calibration. In CKDF, we take an image-feature hybrid storage strategy that trades off the split of the memory budgets to store image and feature exemplars of the previous tasks for training FCN and the new model. CKDF which employs the hybrid storage performs better than using the pure image or feature storage strategy. Our main method (CKDF-iCaRL) which adapts iCaRL by CKDF achieves state-of-the-art performances on three CIL benchmarks. Moreover, CKDF is a general framework and can be easily extended to various kd-based methods to improve performances significantly. We also put insight into components of our approach and analyze the mechanism why using the ground-truth model or FCTM instead of the old model in Fig. 1 is effective through experiments.

We think that the feature uncalibration problem virtually reflects the possible conflict between the existing knowledge of the previous task model and the new knowledge of the current task. It is better to make calibration for the existing knowledge to relieve the conflict before learning the existing knowledge from the previous task model when learning the new task. At the same time, most previous works on CIL use the old model directly. For example, regularization-based approaches consolidate the previous knowledge of the old model by introducing an extra regularization term in the loss function. They may also suffer from the feature uncalibration problem and it would be informative to study the influence of this problem on them in future works.

REFERENCES

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

[2] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," in *Psychology of Learning and Motivation*, vol. 24. Amsterdam, The Netherlands: Elsevier, 1989, pp. 109–165.

[3] I. J. Goodfellow, M. Mirza, D. Xiao, A. Courville, and Y. Bengio, "An empirical investigation of catastrophic forgetting in gradient-based neural networks," 2013, *arXiv:1312.6211*.

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

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

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

[7] A. Iscen, J. Zhang, S. Lazebnik, and C. Schmid, "Memory-efficient incremental learning through feature adaptation," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2020, pp. 699–715.

[8] M. Masana, X. Liu, B. Twardowski, M. Menta, A. D. Bagdanov, and J. van de Weijer, "Class-incremental learning: Survey and performance evaluation on image classification," 2020, *arXiv:2010.15277*.

[9] S. Hou, X. Pan, C. C. Loy, Z. Wang, and D. Lin, "Learning a unified classifier incrementally via rebalancing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 831–839.

[10] H. Ahn, J. Kwak, S. Lim, H. Bang, H. Kim, and T. Moon, "SS-IL: Separated softmax for incremental learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 844–853.

[11] D. Shim, Z. Mai, J. Jeong, S. Sanner, H. Kim, and J. Jang, "Online class-incremental continual learning with adversarial Shapley value," in *Proc. AAAI Conf. Artif. Intell.*, 2021, vol. 35, no. 11, pp. 9630–9638.

[12] Z. Chen and B. Liu, "Lifelong machine learning," *Synth. Lectures Artif. Intell. Mach. Learn.*, vol. 12, no. 3, pp. 1–207, 2018.

[13] M. Delange *et al.*, "A continual learning survey: Defying forgetting in classification tasks," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Feb. 5, 2021, doi: 10.1109/TPAMI.2021.3057446.

[14] K. James *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proc. Nat. Acad. Sci. USA*, vol. 114, no. 13, pp. 3521–3526, Mar. 2017.

[15] S.-W. Lee, J.-H. Kim, J. Jun, J.-W. Ha, and B.-T. Zhang, "Overcoming catastrophic forgetting by incremental moment matching," 2017, *arXiv:1703.08475*.

[16] F. Zenke, B. Poole, and S. Ganguli, "Continual learning through synaptic intelligence," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 3987–3995.

[17] X. Liu, M. Masana, L. Herranz, J. Van de Weijer, A. M. Lopez, and A. D. Bagdanov, "Rotate your networks: Better weight consolidation and less catastrophic forgetting," in *Proc. 24th Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2018, pp. 2262–2268.

[18] R. Aljundi, F. Babiloni, M. Elhoseiny, M. Rohrbach, and T. Tuytelaars, "Memory aware synapses: Learning what (not) to forget," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 139–154.

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

[20] G. Zeng, Y. Chen, B. Cui, and S. Yu, "Continual learning of context-dependent processing in neural networks," *Nature Mach. Intell.*, vol. 1, no. 8, pp. 364–372, Aug. 2019.

[21] B. Cui, G. Hu, and S. Yu, "Deepcollaboration: Collaborative generative and discriminative models for class incremental learning," in *Proc. AAAI Conf. Artif. Intell.*, 2021, vol. 35, no. 2, pp. 1175–1183.

[22] C. Fernando *et al.*, "PathNet: Evolution channels gradient descent in super neural networks," 2017, *arXiv:1701.08734*.

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

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

[25] A. Rosenfeld and J. K. Tsotsos, "Incremental learning through deep adaptation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 3, pp. 651–663, Mar. 2020.

[26] P. Singh, V. K. Verma, P. Mazumder, L. Carin, and P. Rai, "Calibrating CNNs for lifelong learning," in *Proc. NeurIPS*, 2020, pp. 15579–15590.

[27] S. Jung, H. Ahn, S. Cha, and T. Moon, "Continual learning with node-importance based adaptive group sparse regularization," 2020, *arXiv:2003.13726*.

[28] A. Chaudhry, M. Ranzato, M. Rohrbach, and M. Elhoseiny, "Efficient lifelong learning with A-GEM," 2018, *arXiv:1812.00420*.

[29] D. Isele and A. Cosgun, "Selective experience replay for lifelong learning," in *Proc. AAAI Conf. Artif. Intell.*, 2018, vol. 32, no. 1, pp. 3302–3309.

[30] J. Zhang *et al.*, "Class-incremental learning via deep model consolidation," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2020, pp. 1131–1140.

[31] D. Lopez-Paz and M. Ranzato, "Gradient episodic memory for continual learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 6467–6476.

[32] D. Rolnick, A. Ahuja, J. Schwarz, T. P. Lillicrap, and G. Wayne, "Experience replay for continual learning," 2018, *arXiv:1811.11682*.

[33] A. Chaudhry, N. Khan, P. K. Dokania, and P. H. S. Torr, "Continual learning in low-rank orthogonal subspaces," 2020, *arXiv:2010.11635*.

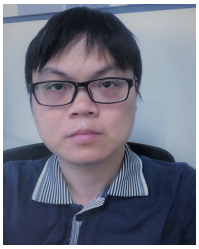
[34] Y. Liu, B. Schiele, and Q. Sun, "RMM: Reinforced memory management for class-incremental learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 3478–3490.

[35] Y. Liu, B. Schiele, and Q. Sun, "Adaptive aggregation networks for class-incremental learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 2544–2553.

[36] M. Mermillod, A. Bugaiska, and P. Bonin, "The stability-plasticity dilemma: Investigating the continuum from catastrophic forgetting to age-limited learning effects," *Frontiers Psychol.*, vol. 4, p. 504, Aug. 2013.

[37] X. Tao, X. Chang, X. Hong, X. Wei, and Y. Gong, "Topology-preserving class-incremental learning," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2020, pp. 254–270.

[38] X. Tao, X. Hong, X. Chang, S. Dong, X. Wei, and Y. Gong, "Few-shot class-incremental learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 12183–12192.
[39] Y. Liu, X. Hong, X. Tao, S. Dong, J. Shi, and Y. Gong, "Structural knowledge organization and transfer for class-incremental learning," in *Proc. ACM Multimedia Asia*, Dec. 2021, pp. 1–7.
[40] S. Dong, X. Hong, X. Tao, X. Chang, X. Wei, and Y. Gong, "Few-shot class-incremental learning via relation knowledge distillation," in *Proc. AAAI Conf. Artif. Intell.*, 2021, vol. 35, no. 2, pp. 1255–1263.
[41] Y. Xiang, Y. Fu, P. Ji, and H. Huang, "Incremental learning using conditional adversarial networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6619–6628.
[42] R. Kemker and C. Kanan, "FearNet: Brain-inspired model for incremental learning," 2017, *arXiv:1711.10563*.
[43] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*.
[44] A. Krizhevsky *et al.*, "Learning multiple layers of features from tiny images," Univ. Toronto, Toronto, ON, Canada, Tech. Rep., 2009. [Online]. Available: https://bibbase.org/network/publication/krizhevsky-hinton-learningmultiplelayersoffeaturesfromtinyimages-2009
[45] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.
[46] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[47] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2579–2605, 2008.



Kunchi Li received the B.S. degree from Nankai University, Tianjin, China, in 2007, and the M.E. degree from the Harbin Institute of Technology, Shenzhen, China, in 2019. He is currently pursuing the Ph.D. degree with the Institute of Automation, Chinese Academy of Sciences (CASIA). His main research interests include machine learning and deep learning.

Jun Wan (Senior Member, IEEE) received the B.S. degree from the China University of Geosciences, Beijing, China, in 2008, and the Ph.D. degree from the Institute of Information Science, Beijing Jiaotong University, Beijing, in 2015. Since January 2015, he has been a Faculty Member with the National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences (CASIA), China, where he currently works as an Associate Professor. His main research interests include computer vision and machine learning. He is an Area and Session Chair of ICME 2021. He served as the Co-Editor for special issues in the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE and the IEEE TRANSACTIONS ON BIOMETRICS, BEHAVIOR, AND IDENTITY SCIENCE. He is an Associate Editor of the *IET Biometrics* from 2020 to 2022.

Shan Yu received the B.S. and Ph.D. degrees in biology from the University of Science and Technology of China, Hefei, China, in 2000 and 2005, respectively. From 2005 to 2014, he conducted postdoctoral research with the Max-Planck Institute of Brain Research, Germany (2005–2008), and the National Institute of Mental Health, USA (2008–2014). After that, he joined the Institute of Automation, Chinese Academy of Sciences (CASIA). He is currently a Professor with the Brainnetome Center and the National Laboratory of Pattern Recognition (NLPR), CASIA. He has authored or coauthored more than 30 peer-reviewed articles in neuroscience and other interdisciplinary fields at leading journals, such as the *Nature Machine Intelligence*, the *Journal of Neuroscience*, and *eLife*. His current research interests include neuronal information processing, brain-inspired computing, and brain–machine interface.