Exploring deep learning for intelligent image retrieval

Chen, W.

Chapter 7

Lifelong Image Retrieval via Dual Knowledge Distillation

In Chapters 5-6, we explored incremental learning on fine-grained datasets. However, this is still far from realizing the model's continuous retrieval ability because the images in old categories and new categories are similar semantically. Instead, the images in new categories may have different semantic contents (*i.e.* semantic shifts). For the context of incremental learning, the semantic shifts make the problem of minimizing the forgetting ratio more difficult.

In this chapter, we investigate RQ 4, with a goal of gradually transferring acquired knowledge for any new task while minimizing the forgetting ratio on old tasks. To this end, we propose a Dual Knowledge Distillation (DKD) framework consisting of two professional teachers and a self-motivated student. One teacher is trained on previous datasets and then freezes its parameters. This frozen teacher is responsible for transferring previous knowledge. The other teacher is trained jointly with the student by using samples from the new incoming dataset only. This "on the fly" teacher is responsible for learning new knowledge and acts as an assistant model to improve the student's generalization ability. As the incremental learning proceeds, the semantic drifts between the old and new datasets often weaken the effectiveness of knowledge distillation by the frozen teacher. To mitigate this problem, we leverage the stored statistics in the BatchNorm layers of the frozen teacher to generate representative images of the old datasets.

Keywords

Lifelong image retrieval, Dual knowledge distillation, Data generation, BatchNorm statistics

This chapter is based on the following publication:

- Chen, W., Pu, N., Liu, Y. , Lao, M., Wang, W., Bakker, E. M., Liu L., Tuytelaars, T., and Lew, M.S., "Lifelong Image Retrieval via Dual Knowledge Distillation." submitted to Thirty-Sixth AAAI Conference on Artificial Intelligence (AAAI) (*under review*), 2021.

7.1 Introduction

Image retrieval have been widely explored in the literature since the emergence of deep learning [130, 131, 132, 226]. Typically, existing retrieval works focus on improving the networks' generalization ability and assume that the target dataset is stationary and fixed. This assumption, however, is infeasible for many real-world scenarios, where the environment is non-stationary. To this end, lifelong learning [251] is proposed to make deep networks learn sequential tasks and adapt to streaming data.

The main challenge for lifelong learning systems is to overcome catastrophic forgetting [252]. Knowledge distillation [214] can be used to reduce forgetting, by transferring the learned information from a trained network (*i.e.* teacher) to a new one (*i.e.* student) [212]. It has been researched for various classification-based tasks, including image classification [213], object detection [217], image generation [216]. However, its efficiency on image retrieval is still less studied due to the challenges below.

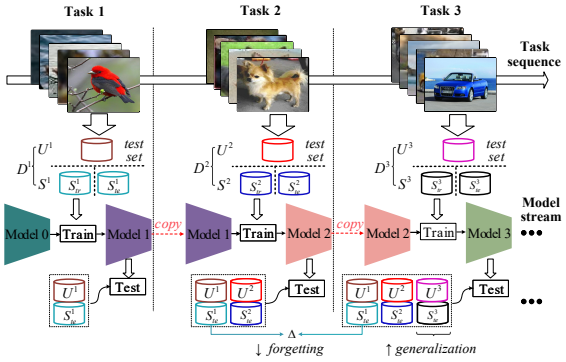

Figure 7.1: Illustration of lifelong image retrieval. A deep model is trained on different sequential datasets $\mathcal{D}^1, \mathcal{D}^2, \mathcal{D}^3, \cdots$. Each dataset is split into a set of seen categories $\mathcal{S}$ and a set of unseen categories $\mathcal{U}$. The semantic difference (*e.g.* birds v.s. cars) results in forgetting when the model is trained on a sequence of task. Thus, the goal is to train the model to minimize the forgetting ratio on the old tasks and simultaneously improve generalization on the new task.

First, a deep model learns to retrieve incrementally on different tasks, and the semantic drifts between the training data lead to tasks that maybe very weakly related, for example the birds, dogs and cars in Figure 7.1. Thus, knowledge distillation cannot effectively prevent the forgetting on streaming data across different tasks. Second, the weak relatedness between tasks results in significant updates of model's parameters when this model learns a new task. Image retrieval is highly sensitive to the matching between features. Thus a small change in the features would have a significant impact on feature matching. The changes in output features make the problem of minimizing forgetting more difficult. Third, conventional knowledge distillation framework pays more attention on preserving the knowledge in the teacher network. This may make it hard to pursue an optimal balance between minimizing the forgetting ratio and improving network's retrieval generalization capacity.

In this chapter, we focus on the three challenges and propose a *Dual Knowledge Distillation* (DKD) framework which includes two professional teachers and a stu-

dent. On the one hand, the first teacher has been trained on previous tasks to transfer old knowledge. To further alleviate the forgetting of the student, we use the statistics stored in the BatchNorm layers of the frozen teacher to generate images used as representatives for the previous datasets. Instead of storing a small budget of exemplars derived from the old data or synthesizing images via training additional generative networks, the representative images can be directly generated from the frozen teacher, without any other operations. On the other hand, the second teacher is trained jointly with the student by using samples from the new task only. This "on-the-fly" teacher acts as an assistant model to improve the student's generalization ability on the new task. Finally, the student can achieve an optimal balance between minimizing the forgetting ratio and improving generalization performance.

7.2 Related Work

Lifelong learning a.k.a. incremental learning, has been explored in image classification [213], object detection [217], image generation [216], and image retrieval [37, 218] *etc*The methods can be broadly divided into three methodologies: network architecture-based [230], memory replay-based [221, 231], and regularization-based methods [213, 227]. Knowledge distillation is one of the regularization-based methods, which can be performed on either the final classifier or the intermediate layers. The key is to minimize the differences between the teacher and the student, which can be characterized by cross-entropy [214], L1 loss [216], L2 loss [253], Gramian matrix [238], and KL-divergence [214]. Multi-teacher knowledge distillation methods have been explored [237]. The ensemble of multiple teachers, *e.g.* by averaging their responses, can provide more powerful prior information for supervising the student. In this chapter, we propose a dual knowledge distillation framework which includes two professional teachers for transferring both old and new knowledge information.

Metric learning has been explored broadly for image retrieval [130, 131, 132, 226]. Given binary indicator information for samples (*i.e.* positive or negative), deep networks learn an embedding space for the features which should be verified as positive pairs or negative pairs [254]. To date, the mainstream methods train deep networks on the seen classes of a fixed dataset and then their generalization performance are validated on the unseen classes of this dataset. Therefore, metric learning for image retrieval focuses on the forward transfer [230], *i.e.* transferring a positive influence to improve the performance on future unseen data. Nevertheless, these methods do not consider the negative backward transfer issue (*i.e.* catastrophic forgetting). Therefore, we explore lifelong image retrieval, with the goal to reduce forgetting and simultaneously improve generalization ability.

BatchNorm statistics utilization. The statistics stored in the BatchNorm layers of a pre-trained model are used for data-free knowledge distillation [255] and data-free model compression [256]. These statistics are relevant to the statistical characteristics of the datasets trained in the past. They have been used as a reference to generate images. For instance, Yin *et al.* [255] introduced Adaptive DeepInversion (ADI) which is a feature map regularizer based on BatchNorm statistics that enables image synthesis from random noise. The generated images have similar semantics to the images of ImageNet. The images generated in [255] depend on optimizing the gradients computed from cross-entropy loss based on the given class labels. This is not directly applicable to lifelong image retrieval because (1) the order of given class labels may affect the softmax-based probabilities of a classifier as the tasks are added sequentially; (2) lifelong image retrieval tasks do not depend on softmax-based probabilities to perform. Instead, we apply a clustering loss to generate images.

7.3 The Lifelong Image Retrieval Problem

Preliminary. To perform image retrieval, a dataset $\mathcal{D}$ is split into a training set $\mathcal{D}_{tr}$ and a testing set $\mathcal{D}_{te}$. A deep network $f(\cdot, \boldsymbol{\theta})$ is trained on $\mathcal{D}_{tr}$ to learn representations $\boldsymbol{F} = f(X, \boldsymbol{\theta})$ by using a certain objective function. To date, ranking loss has been widely used as a constraint to train the network f. Taking the triplet loss as an example, each ground-truth label in $\mathcal{D}_{tr}$ is used to mine a positive x_p, a hard negative x_n, and an anchor image x_a. The network f is trained to learn a feature space, where the distance between x_n and x_a denoted by $D(x_a, x_n) = ||f(x_a; \boldsymbol{\theta}) - f(x_n; \boldsymbol{\theta})||_2^2$ is pushed away by a margin $\delta > 0$ from $D(x_a, x_p)$:

$$L_{triplet}(x_a, x_p, x_n) = \max\Big(\delta + D(x_a, x_p) - D(x_a, x_n), 0\Big) \tag{7.1}$$

Problem definition. We use the triplet loss as a basic constraint to train a model to perform tasks incrementally. The flowchart is illustrated in Figure 7.1. Each task t corresponds to the training of a whole dataset $\mathcal{D}^t$ (*e.g.* birds). During the t^{th} task, dataset $\mathcal{D}^t$ is split into a set of seen categories $\mathcal{S}^t$ and a set of unseen categories $\mathcal{U}^t$. For the seen part, $\mathcal{S}^t$ includes n_s categories, *i.e.* $\mathcal{S}^t = \{(X^c, y^c) | c = 1, 2, \cdots, n_s\}$, each class c includes a different amount of images $|X^c|$ sharing the same label y^c. The $\mathcal{S}^t$ part is further split into a training set and a testing set. Likewise, the unseen part $\mathcal{U}^t$ includes n_u categories, all of which are used to evaluate the model's generalization ability, similar to the general practice in metric learning for image retrieval. For lifelong image retrieval, suppose a deep model has been trained sequentially on the training sets $\mathcal{S}^1, \mathcal{S}^2, \cdots, \mathcal{S}^t$ (current task t). On the one hand, it is required that the trained model is able to minimize the forgetting ratios on the previous tasks $\mathcal{S}^1, \mathcal{S}^2, \cdots, \mathcal{S}^{t-1}$ and $\mathcal{U}^1, \mathcal{U}^2, \cdots, \mathcal{U}^{t-1}$, thereby retaining its retrieval capacity on the previous datasets $\mathcal{D}^1, \mathcal{D}^2, \cdots, \mathcal{D}^{t-1}$. On the other hand, it is required that the

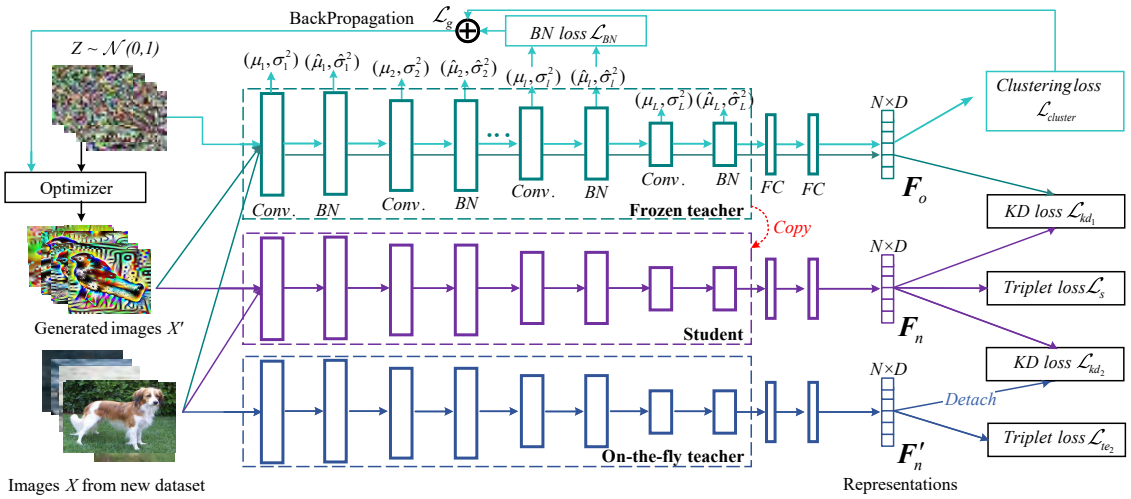

Figure 7.2: The dual knowledge distillation (DKD) framework. The stored statistics in the BatchNorm layers of the frozen teacher are used to generate representative images, optimized by the L_g. The on-the-fly teacher is initialized its parameters differently from the frozen teacher and trained jointly with the student by using L_{te_2}. For clarity, the ReLU activation function and pooling layers are not depicted.

trained model achieves good accuracy on the seen part $\mathcal{S}^t$ and, more importantly, generalizes well on the unseen part $\mathcal{U}^t$ of current dataset $\mathcal{D}^t$.

7.4 Dual Knowledge Distillation

To minimize the forgetting ratio and simultaneously improve generalization performance, we propose a dual knowledge distillation (DKD) framework which includes two teachers and a student, as shown in Figure 7.2. In the following, we will introduce each component in more detail.

7.4.1 Knowledge distillation by frozen teacher

Prior to training task t, a teacher model has been trained on the previous task $(t-1)$ and has its parameters fixed. Training the student on the new task t leads to a negative backward transfer which may degrade the performance of preceding tasks [230]. Knowledge distillation by using the frozen teacher $f_{te_1}^{t-1}$ can prevent this degradation. As shown in Figure 7.2, knowledge distillation by using the frozen teacher is performed on the embedded D-dimension features from the fully-connected layers, formulated as $\boldsymbol{F}_o = f_{te_1}^{t-1}(X^c, \boldsymbol{\theta}_{te_1}^{t-1}) \in \mathbb{R}^{N\times D}$, where N is the size of a mini-batch. Likewise, the feature representations from the student f_s^t are $\boldsymbol{F}_n = f_s^t(X^c, \boldsymbol{\theta}_s^t) \in \mathbb{R}^{N\times D}$. As suggested in [245, 257, 258], semantically similar inputs produce similar patterns on features extracted from the frozen teacher and the student. Therefore, we adopt the Gram matrix with a kernel function to measure the feature correlations:

$$G_o^{(i,j)} = \mathcal{K}(F_o^i, F_o^j);\ G_n^{(i,j)} = \mathcal{K}(F_n^i, F_n^j) \tag{7.2}$$

$\mathcal{K}(\cdot)$ refers to inner product, *i.e.*, $\mathcal{K}(F^i, F^j) = <F^i, F^j>$. Each entry (i, j) in $\boldsymbol{G} \in \mathbb{R}^{N\times N}$ represents the correlations of the same activation ($i = j$) or these between different activations ($i \neq j$). We use KL-divergence to measure the difference between $\boldsymbol{G}_o$ and $\boldsymbol{G}_n$, normalized by a Softmax function $\sigma(\cdot)$. Thus, the knowledge distillation loss by the frozen teacher $f_{te_1}^{t-1}$ is formulated as L_{kd_1}, weighted by a factor λ_{kd_1}:

$$L_{kd_1} = \lambda_{kd_1} \sum^{N} KL\Big(\sigma(\boldsymbol{G}_o), \sigma(\boldsymbol{G}_n)\Big) \tag{7.3}$$

7.4.2 Representative data generation

When the student learns task t, the performance degradation of preceding tasks is prevented by using the KL-divergence in Eq. 7.3. However, when the student is trained incrementally on the data with large semantic drifts (*e.g.* birds and cars in Figure 7.1), L_{kd_1} cannot effectively prevent the degradation by transferring more previously learned information. To overcome this problem, we use the stored statistics in BatchNorm layers to generate samples as representatives for the previous tasks. Representative data generation is performed by the frozen teacher itself, instead of selecting exemplars from these already-trained datasets.

Suppose the frozen teacher includes L convolutional layers, each of which is followed by a BatchNorm layer, as shown in Figure 7.2. Each BatchNorm layer l includes channel-wise running means $\hat{\mu}_l$ and running variances $\hat{\sigma}_l^2$. Prior to training the student, a batch of Gaussian noise Z with random class labels Y' are fed into the teacher. Outputs of each convolutional layer l of the teacher are used to compute the batch means μ_l and batch variances σ_l^2. Similar to [255], we define a BN loss L_{BN} to measure the difference between the stored statistics and the current statistics of Z:

$$L_{BN} = \lambda_{BN} \sum_{l=1}^{L} \Big(\|\mu_l(Z) - \hat{\mu}_l\|_2^2 + \|\sigma_l^2(Z) - \hat{\sigma}_l^2\|_2^2 \Big) \tag{7.4}$$

Different from ADI in [255] which is limited only from the classification networks. We apply a K-means clustering loss $L_{cluster}$, together with L_{BN} to optimize Z. Given a mini-batch of N noise tensors with K classes, containing P tensors of each given class, the mean M_k for a class $k \in K$ is defined as $M_k = \frac{1}{P}\sum_{p=1}^{P} f_{te_1}^{t-1}(z_{k_p}, \boldsymbol{\theta}_{te_1}^{t-1})$, where z_{k_p} is a sample from the tensors Z. The number of clusters is set to the number of classes in tensors Z (*i.e.* K classes). We cluster features of Z via computing intra-class and inter-class distances. Specifically, for a given class $k \in K$, a set of intra-class distances d_k^{intra} is formulated as $\{||f_{te_1}^{t-1}(z_{k_p}, \boldsymbol{\theta}_{te_1}^{t-1}) - M_k||_2|\}$, where $p = 1, 2, ..., P$ and the number of elements in d_k^{intra} is equal to P. Likewise, a set of

inter-class distances d_k^{inter} is computed according to all other $(N-P)$ samples from k'_p classes ($k'_p \in K$ and $k'_p \neq k$). Clustering all the elements in d_k^{intra} and d_k^{inter} leads to a low training efficiency. Instead, we mine the hardest samples in these distance sets. For d_k^{intra}, we mine the sample that lies farthest from its class mean M_k. For d_k^{inter}, we mine the sample that lies closest from the considered class mean M_k. For all K classes, we use a clustering loss $L_{cluster}$ to regularize the inter-class variations to become larger than the intra-class variations by a margin $\Delta > 0$:

$$L_{cluster} = \lambda_{cluster} \sum_{k=1}^{K} \max\Big(\Delta + \max_{P} d_k^{intra} - \min_{N-P} d_k^{inter}, 0\Big) \tag{7.5}$$

Afterwards, the loss $L_g = L_{BN} + L_{cluster}$ is used to optimize Z based on the frozen teacher $f_{te_1}^{t-1}(\cdot, \boldsymbol{\theta}_{te_1}^{t-1})$ to generate representative images X' of the previous task $(t-1)$, *i.e.* $X' \leftarrow \underset{z \in Z}{\operatorname{argmin}} \sum(L_g; \boldsymbol{\theta}_{te_1}^{t-1})$. Images X' and class labels Y' can be used to build a mixed dataset $X_{mix} = X \cup X'$. X belongs to the origin training set in D^t. The mixed labels are $Y_{mix} = Y \cup Y'$. In this case, the mixed data are fed into the frozen teacher $f_{te_1}^{t-1}$ to transfer richer previous knowledge to the student.

7.4.3 Self-motivated learning on the mixed data

At the start of task t, the parameters of the student are copied from the frozen teacher, as shown in Figure 7.1. The self-motivated learning for the student is important for guaranteeing the performance on the current task t, as can be seen from the results for *Case 4* in Table 7.6. Consistent to the training scheme for the frozen teacher, we employ the triplet loss in a similar form as Eq. 7.1 to train the student.

$$L_s = \lambda_s \sum^{N} L_{triplet}\Big(f_s^t(x'_a), f_s^t(x'_p), f_s^t(x'_n)\Big) \tag{7.6}$$

Note that the anchor, positive, and negative images (x'_a, x'_p, x'_n) are from the mixed dataset X_{mix} according to the mixed labels Y_{mix} in each training session.

7.4.4 Auxiliary distillation by on-the-fly teacher

During training, the student needs to learn new information and simultaneously protect previous knowledge. However, knowledge distillation from the mixed data using the frozen teacher is a strong regularization by the time it reaches the student, making the student be prone to remembering previous knowledge but having lower generalization on the new task t, as demonstrated by *Case 2* in Table 7.6. As a result, an optimal balance between reducing forgetting and improving generalization is hard to achieve. Therefore, we propose a second teacher $f_{te_2}^t$ which is trained together with the student. Its parameters $\boldsymbol{\theta}_{te_2}^t$ are initialized differently from these

of the frozen teacher and the student. This teacher is constrained by a triplet loss L_{te_2}:

$$L_{te_2} = \lambda_{te_2} \sum^{N} L_{triplet}\Big(f^t_{te_2}(x_a), f^t_{te_2}(x_p), f^t_{te_2}(x_n)\Big) \tag{7.7}$$

For L_{te_2}, the training images (x_a, x_p, x_n) are mined only from $S^t = \{(X^c, y^c)|c = 1, 2, \cdots, n_s\}$ of the dataset D^t, rather than the mixed data X_{mix}, see Figure 7.2. The on-the-fly teacher is designed to transfer new information to the student to improve its generalization ability. Thus, an auxiliary knowledge distillation loss L_{kd_2} is defined as:

$$\begin{aligned} &L_{kd_2} = \lambda_{kd_2} \sum^{N} KL\Big(\sigma(\boldsymbol{G}'_n), \sigma(\boldsymbol{G}_n)\Big) \\ &\text{where } G'_n = \mathcal{K}(F'_n, F'_n),\ F'_n = f^t_{te_2}(x, \boldsymbol{\theta}^t_{te_2}); \\ &G_n = \mathcal{K}(F_n, F_n),\ F_n = f^t_s(x, \boldsymbol{\theta}^t_s); x \in X \end{aligned} \tag{7.8}$$

Note that during training the gradients computed from L_{kd_2} are *detached* for the on-the-fly teacher. This operation can guarantee the on-the-fly teacher to be fully dedicated to capturing new information from the new dataset $\mathcal{D}^t$.

Full objective. When training with dataset $\mathcal{D}^t$ on task t, together with the generated images, the DKD framework is running by using the full objective function:

$$L = \lambda_s L_s + \lambda_{kd_1} L_{kd_1} + \lambda_{kd_2} L_{kd_2} + \lambda_{te_2} L_{te_2} \tag{7.9}$$

7.5 Experiments

7.5.1 Dataset splits

Our experimental methodology involves using sequences of two tasks and sequences of three tasks in a roughly similar way as the recent lifelong learning research [259]. We perform experiments on three datasets: Caltech-UCSD Birds (CUB-200) [224], Stanford-Dogs [223], and Stanford-Cars [260].

- *CUB-200* includes 11,788 images of 200 classes. We select 150 classes (8,822 images) as the seen set $\mathcal{S}$ and use the remaining 50 classes as unseen set $\mathcal{U}$ (2,966 images). For the seen set, we select ∼60% of each class for training (5,274 images), while the remaining ∼40% (3,548 images) are used to evaluate the forgetting ratio.

- *Stanford-Dogs* includes 20,580 images of 120 classes. We select 100 classes (17,028 images) as the seen set and use the remaining 20 classes as unseen set $\mathcal{U}$ (3,552 images). For the seen set, we select ∼80% of each class for training (13,063 images), while the remaining ∼20% (3,965 images) are for testing.

- *Stanford-Cars* includes 16,185 images of 196 classes. We select 160 classes (10,038 images) as the seen set and use the remaining 36 classes as unseen set $\mathcal{U}$ (3,040 images). For the seen set, we select ∼80% images of each class for training (10,038 images), while the remaining ∼20% (3,107 images) are used at test.

7.5.2 Training details

We utilize the pre-trained Google Inception with BatchNorm as a backbone net. The on-the-fly teacher is always initialized from the pre-stored parameters learned from ImageNet before training each task. Following the practice in [131, 226], the final retrieval features are 512-D. The model is trained for 1500 epochs on the first dataset to get the initial frozen teacher. The training is constrained by the triplet loss with a margin $\delta = 0.5$ as given in Eq. 7.1, optimized by the Adam optimizer with a learning rate of 1×10^{-6} and a batch size of 32, The fully-connected layers for dimension reducing are updated with a learning rate of 1×10^{-5}. Representative images are generated by using Eqs. 7.4 and 7.5 where factors λ_{BN} and $\lambda_{cluster}$ are set to 0.01 and 0.1, respectively. Δ in Eq. 7.5 is set to 1.0. The image generation process is optimized by an additional Adam optimizer with a learning rate of 0.5. The factors λ_s, λ_{te_2}, λ_{kd_1}, and λ_{kd_2} in Eq. 7.9 are set to 1, 1, 80, 20, respectively. We include the main steps of the Dual Knowledge Distillation (DKD) framework in Algorithm 2. Before training each task, the student initializes its parameters from the frozen teacher. Differently, the on-the-fly teacher is always initialized from the pre-stored parameters of Google Inception learned from the ImageNet. In addition, its fully-connected layers are initialized randomly. Image generation process is performed prior to training the student model. The whole framework is trained in an end-to-end manner.

7.5.3 Performance evaluation

Baseline. To the best of our knowledge, there is no prior work for lifelong image retrieval performed on different datasets. We build the sequential fine-tuning (SFT) method as a baseline, which is performed by using a triplet loss as defined in Eq 7.1. We compare 3 knowledge distillation methods, including L_1 loss [216], L_2 loss [217], and maximum mean discrepancy loss (L_{mmd} in short) [37]. We claim the work of incremental fine-grained image retrieval [37] is less challenging than ours because the new data and old are from the same dataset in [37]. Similar to [259], we use the joint training on the training sets of 3 datasets as the *upper-bound* reference for all compared methods.

Metrics. We evaluate the performance of seen set s and that of unseen set u by using the standard performance metric *Recall@K* (*i.e.* *R@K*). The evaluation for u is the same as the one widely explored in deep metric learning [130, 131, 132, 226] which aims at demonstrating the generalization ability. The evaluation for s aims

Algorithm 2: Dual Knowledge Distillation (DKD) framework

Input:
Frozen teacher $f_{te_1}^{t-1}(\cdot, \boldsymbol{\theta}_{te_1}^{t-1})$ has been trained on the previous task $t-1$;
New training images $X \in \mathbb{R}^{N\times H\times W\times 3}$ and labels $Y \in \mathbb{R}^{N\times 1}$ on the training set of $\mathcal{S}^t$ on the current dataset $\mathcal{D}^t$;
Initialization:
$\boldsymbol{\theta}_s^t = \boldsymbol{\theta}_{te_1}^{t-1}$ *//Copied the frozen teacher as the initial student;*
$\boldsymbol{\theta}_{te_2}^t \leftarrow$ Google Inception *//Initialize on-the-fly teacher;*
Random noise tensor $Z \in \mathbb{R}^{N\times H\times W\times 3}$;
Random labels $Y' \in \mathbb{R}^{N\times 1}$ for input noise Z *//Include K classes in total;*
Iterations *Iter* of image generation; Training epochs *Epoch*; Mini-batch size N;
Optimizer with a learning rate lr_1;
Training:
For $iter = 0$ to *Iter*
$\boldsymbol{F}(Z) = f_{te_1}^{t-1}(Z, \boldsymbol{\theta}_{te_1}^{t-1}) \in \mathbb{R}^{N\times D}$ *//Features to calculate cluster means, inter-class distance sets, and intra-class distance sets;*
$L_{BN} = \sum_{l=1}^{L}\Big(\|\mu_l(Z) - \hat{\mu}_l\|_2^2 + \big\|\sigma_l^2(Z) - \hat{\sigma}_l^2\big\|_2^2\Big)$ *//BN loss in Eq. 7.4;*
$L_{cluster} = \sum_{k=1}^{K}\max\Big(\Delta + \max_{P} d_k^{intra} - \min_{N-P} d_k^{inter}, 0\Big)$ *//Clustering loss in Eq. 7.5;*
$X' \leftarrow \underset{Z}{\operatorname{argmin}} \sum\Big((L_{BN} + L_{cluster}); \boldsymbol{\theta}_{te_1}^{t-1}\Big)$ *//Using the optimizer with lr_1;*
End for
For $epoch = 0$ to *Epoch*
$X_{mix} = X \cup X'$, $Y_{mix} = Y \cup Y'$ *//Build a mixed dataset via data concatenation;*
$\boldsymbol{F}_o = f_{te_1}^{t-1}(X_{mix}, \boldsymbol{\theta}_{te_1}^{t-1}) \in \mathbb{R}^{2N\times D}$ *//$2N \times D$-dim features from the frozen teacher;*
$\boldsymbol{F}_n = f_s^t(X_{mix}, \boldsymbol{\theta}_s^t) \in \mathbb{R}^{2N\times D}$ *//$2N \times D$-dim features from the student;*
$\boldsymbol{F}_n' = f_{te_2}^t(X, \boldsymbol{\theta}_{te_2}^t) \in \mathbb{R}^{N\times D}$ *//$N \times D$-dim features from the on-the-fly teacher;*
$L_{kd_1} = \mathrm{KL}(\boldsymbol{F}_o, \boldsymbol{F}_n)$ *//Knowledge distillation from the frozen teacher in Eq. 7.3;*
$L_s = \mathrm{Triplet}(\boldsymbol{F}_n, Y_{mix})$ *//Triplet loss from the student in Eq. 7.6;*
$L_{te_2} = \mathrm{Triplet}(\boldsymbol{F}_n', Y)$ *//Triplet loss from the on-the-fly teacher in Eq. 7.7;*
$L_{kd_2} = \mathrm{KL}(\boldsymbol{F}_n', \{\boldsymbol{F}_n\}_{n=1,\ldots,N})$ *//Knowledge distillation in Eq. 7.8;*
$L = L_s + L_{kd_1} + L_{kd_2} + L_{te_2}$ *//Weighted full loss function in Eq. 7.9;*
End for
Output: The optimized student model $f_s^t(\cdot, \boldsymbol{\theta}_s^t)$.

to analyze the forgetting ratio of a considered model. Similar to [261], we use the harmonic mean H of s and u to evaluate the trained model, which the most important metrics for each task.

$$H = \frac{2 \times s \times u}{s + u} \tag{7.10}$$

Results. We consider the two-task scenario and three-task scenario. For the two-task scenario, we use CUB-200 as the first task, and consider the task sequences: CUB-200 → Stanford-Dogs and CUB-200 → Stanford-Cars. The results are reported in Tables 7.1 and 7.2. For the three-task scenario, we randomly select a task sequence starting with CUB-200: CUB-200 → Stanford-Dogs → Stanford-Cars. The results

are reported in Table 7.3. For clarity, we report the Recall@1 results.

(1) Two-task evaluation. As shown in Tables 7.1 and 7.2, three experimental comparisons are reported. Compared to the reference, fine-tuning on the Stanford-Dogs and Stanford-Cars achieves a Recall@1 of 78.0% and 77.5% of H on the second task, respectively, while fine-tuning suffers from forgetting on the first task. If "one-teacher" knowledge distillation methods are performed, the student suffers less from forgetting. However, the improvements on the first task are limited due to the semantic drifts. When BatchNorm statistics are used to address this limitation, we observe that the students regularized by different methods are both prone to remembering the first task but degrading their generalization ability on the second task. This is caused by the strong regularization from the frozen teacher, together with the representative images. If the on-the-fly teacher is used (*i.e.* "DKD + BN statistics"), the generalization performance on the second task is improved or even surpass that from the baseline. For instance, on sequence "CUB-200 $\rightarrow$ Stanford-Dogs" in Table 7.1, when knowledge distillation in the DKD framework is realized by using KL-divergence in Eqs. 7.3 and 7.8, the overall Recall@1 reaches to 80.0%, higher than the 78.0% of the baseline. This demonstrates the efficiency of the auxiliary distillation. At the same time, the student suffers from the minimal degradation on the first task, with a Recall@1 of 67.0%, compared to the 68.7% of the reference. Likewise, on sequence "CUB-200 $\rightarrow$ Stanford-Cars" in Table 7.2, the student has a Recall@1 of 60.7% compared to 67.7% of the reference. This larger difference is caused by the different distributions between training data of Stanford-Dogs and that of Stanford-Cars.

(2) Three-task evaluation. When three tasks are performed incrementally, the student trained on the final task is tested on the previous two datasets. The results are reported in Table 7.3. Specifically, the generalization performance of the DKD framework on the last task (*i.e.* on Stanford-Cars) is close to or even surpasses the reference performance of joint training (*i.e.* 78.1% and 77.8%). Compared to the two-task scenario, training on the sequence of three tasks leads to more forgetting on the preceding tasks due to the accumulated semantic drifts, especially for the first task. We compare the forgetting ratios of the compared methods on CUB-200. As depicted in Figure 7.3, the initial model is converged at 1500 training epochs on CUB-200, with Recall@1=74.8% on seen set and Recall@1=61.6% on unseen set. We observe that the SFT method degrades performance significantly. Training on the sequence of three tasks also causes forgetting on the unseen set, as shown in Figure 7.3(b). In comparison, the proposed DKD reduces the degradation greatly and is closer to the upper-bound reference.

(3) Evaluation of the on-the-fly teacher. Due to the gradients detach operation, the on-the-fly teacher learns the new task, only being regularized by the term L_{te_2} in Eq. 7.7. We follow the setup of the two-task scenario in Table 7.1, and

Table 7.1: Recall@K (K=1) comparison (%) of s and u for the sequence "CUB-200 → Stanford-Dogs". "KD" represents that one frozen teacher is used for knowledge distillation only. For all cases, the student is regularized by triplet loss only. "KL-divergence" denotes that the knowledge is transferred by using Eq. 7.3. The best balanced results are highlighted in boldface.

		CUB-200 → Stanford-Dogs					
		Test on CUB-200			Test on Stanford-Dogs		
		s	u	H	s	u	H
	Recall@K	K=1	K=1	K=1	K=1	K=1	K=1
Baseline	FT [226]	56.0	47.5	51.4	72.2	84.9	78.0
KD	L_1 loss [216]	52.1	47.4	49.6	71.1	78.7	74.7
	L_{mmd} loss [37]	62.3	52.2	56.8	73.3	85.3	78.9
	L_2 loss [217]	60.5	49.9	54.7	73.7	85.0	78.9
	KL-divergence	62.2	52.1	56.7	73.6	85.0	78.9
KD + BN statistics	L_1 loss [216]	72.0	60.7	65.9	49.8	76.8	60.4
	L_{mmd} loss [37]	73.1	61.7	66.9	49.7	76.3	60.2
	L_2 loss [217]	72.5	62.3	67.0	49.4	75.5	59.7
	KL-divergence	73.5	63.8	68.3	60.0	80.3	68.7
DKD + BN statistics	L_1 loss [216]	64.1	53.3	58.2	74.3	84.8	79.2
	L_{mmd} loss [37]	68.6	60.1	64.1	73.8	85.9	79.4
	L_2 loss [217]	71.7	61.1	66.0	72.1	85.2	78.1
	KL-divergence	**72.0**	**62.7**	**67.0**	**74.4**	**86.5**	**80.0**
Reference	Joint training	74.1	64.1	68.7	74.5	86.7	80.1

Table 7.2: Recall@K (K=1) comparison (%) of s and u for the sequence "CUB-200 → Stanford-Cars". "KD" represents that one frozen teacher is used for knowledge distillation only. For all cases, the student is regularized by triplet loss only. "KL-divergence" denotes that the knowledge is transferred by using Eq. 7.3. The best balanced results are highlighted in boldface.

		CUB-200 → Stanford-Cars					
		Test on CUB-200			Test on Stanford-Cars		
		s	u	H	s	u	H
	Recall@K	K=1	K=1	K=1	K=1	K=1	K=1
Baseline	FT [226]	41.8	38.4	40.0	74.9	80.2	77.5
KD	L_1 loss [216]	43.9	37.1	40.2	72.6	79.2	75.8
	L_{mmd} loss [37]	46.4	39.2	42.5	75.4	79.0	77.2
	L_2 loss [217]	44.5	38.4	41.2	74.7	80.2	77.4
	KL-divergence	45.0	40.5	42.6	74.3	80.6	77.3
KD + BN statistics	L_1 loss [216]	58.7	50.8	54.5	68.4	75.4	71.7
	L_{mmd} loss [37]	64.5	57.2	60.6	64.3	73.6	68.6
	L_2 loss [217]	63.4	56.0	59.5	69.9	76.4	73.0
	KL-divergence	64.5	57.1	60.6	69.8	78.5	73.9
DKD + BN statistics	L_1 loss [216]	54.9	45.4	49.7	73.3	80.6	76.8
	L_{mmd} loss [37]	52.0	**63.8**	57.3	72.7	79.5	76.0
	L_2 loss [217]	57.2	49.9	53.3	74.1	80.4	77.1
	KL-divergence	**64.6**	57.3	**60.7**	**74.6**	**83.5**	**78.8**
Reference	Joint training	72.1	63.8	67.7	77.5	82.2	79.8

report the performance of the on-the-fly teacher under the training sequence: CUB-200 → Stanford-Dogs. Since this teacher is specific for transferring newly-learned information of a new dataset, we only report its performance on the second task (*i.e.*

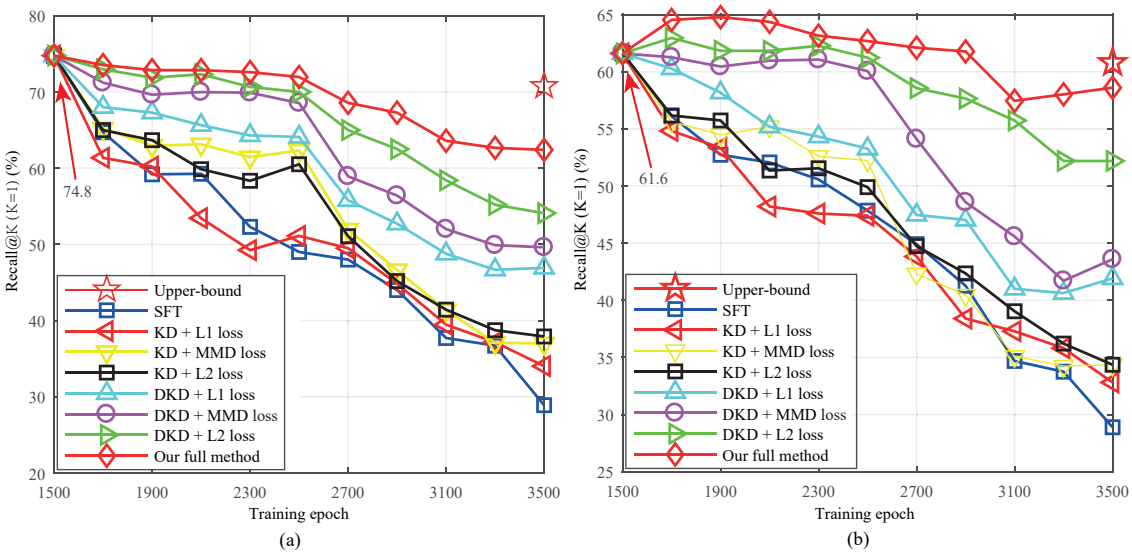

Figure 7.3: The performance degradation evaluation on the CUB-200 dataset: (a) on the seen set; and (b) on the unseen set.

Stanford-Dogs), which are shown in Table 7.4. The "Student model" refers to the model trained by our DKD. We observe that this on-the-fly teacher achieves good generalization performance on the new task.

(4) Evaluation of the generated images. One benefit of using BatchNorm layers is that the representative images can be directly generated using the frozen teacher, without any other operations or additional generative networks. For evaluation, we select the generated images by using the frozen teacher trained on CUB-200, evaluated by using the inception score [262] and Fréchet Inception Distance (FID) [263]. The origin images are chosen randomly from previous 70 classes (4076 images) on CUB-200. These class labels are used to generate equal representative images. As shown in Table 7.5, these results demonstrate that the efficacy of loss terms L_{BN} and $L_{cluster}$ for generating images. Moreover, several generated birds images for the CUB-200 dataset are visualized in Figure 7.4. The generated representative images for the Stanford-Dogs dataset are listed in Figure 7.5). As required by lifelong image retrieval, this student needs to remember previously learned knowledge and capture new information on the new dataset (*i.e.* Stanford-Dogs). As a result, the images generated by this trained student model share some properties for Birds images and Dog images. Similarly, the representative images generated for the Stanford-Cars dataset are shown in Figure 7.6. We observe that these representative images show more semantics for the Cars images. The reason is that the student is prone to learning new information on the Stanford-Cars dataset. Furthermore, the image generation process on the CUB-200 dataset is illustrated in Figure 7.7. The initial input is random Gaussian noise, which is optimized iteratively until $Iter = 2000$, as can be seen in Algorithm 2.

(5) Ablation study. We perform an ablation analysis of the proposed method,

Table 7.3: Recall@K (K=1) comparison (%) of s and u on three datasets. The results are reported when the model is trained on Stanford-Cars and then tested backward on the previous two datasets. ‡ refers to BatchNorm statistics are used for enhancing the knowledge distillation using the frozen teacher only. Likewise, ‡ refers to BatchNorm statistics are used to enhance the frozen teacher. The best balanced results are highlighted in boldface.

	CUB-200 → Stanford-Dogs → Stanford-Cars								
	Test on CUB-200			Test on Stanford-Dogs			Test on Stanford-Cars		
	s	u	H	s	u	H	s	u	H
Recall@K	K=1	K=1	K=1	K=1	K=1	K=1	K=1	K=1	K=1
SFT [226]	28.9	28.1	28.5	40.6	63.3	49.5	72.6	78.1	75.3
KD+L_1 loss [216]	34.0	32.8	33.4	44.5	68.3	53.9	71.8	79.3	75.4
KD+L_{mmd} loss[37]	37.0	34.4	35.7	46.1	69.7	55.5	72.0	76.9	74.4
KD+L_2 loss[217]	37.9	34.4	36.1	43.8	67.8	53.2	74.9	80.8	77.7
KD+ KL div.	37.3	34.3	35.7	45.9	69.1	55.2	71.9	80.6	76.0
KD+L_1 loss†	69.7	58.5	63.6	44.2	74.2	55.4	37.9	58.1	45.9
KD+L_{mmd} loss†	70.7	60.8	65.4	47.9	76.1	58.8	40.3	58.4	47.7
KD+L_2 loss†	70.9	62.3	66.3	53.8	79.8	64.3	40.2	58.3	47.6
KD+KL div.†	71.1	65.6	68.2	55.8	80.2	65.8	40.8	58.9	48.2
DKD+L_1 loss‡	46.9	41.9	44.3	59.5	77.8	67.4	74.1	80.8	77.3
DKD+L_{mmd}‡	49.6	43.6	46.4	58.7	77.4	66.8	71.5	78.9	75.0
DKD+L_2 loss‡	54.1	52.2	53.1	58.8	78.6	67.3	**75.1**	80.8	77.9
DKD+KL div.‡	**62.4**	**58.6**	**60.5**	**67.4**	**84.3**	**74.9**	73.2	**83.7**	**78.1**
Joint training	71.5	62.5	66.7	71.2	83.3	76.8	74.3	81.6	77.8

Table 7.4: Evaluation for the on-the-fly teacher on the second task.

	CUB-200 → Stanford-Dogs		
	Test on Stanford-Dogs		
	s	u	H
Recall@K	K=1	K=1	K=1
Fine-tuning	72.2	84.9	78.0
Student model	74.4	86.5	80.0
On-the-fly teacher	74.6	86.3	80.0
Joint training	74.5	86.7	80.1

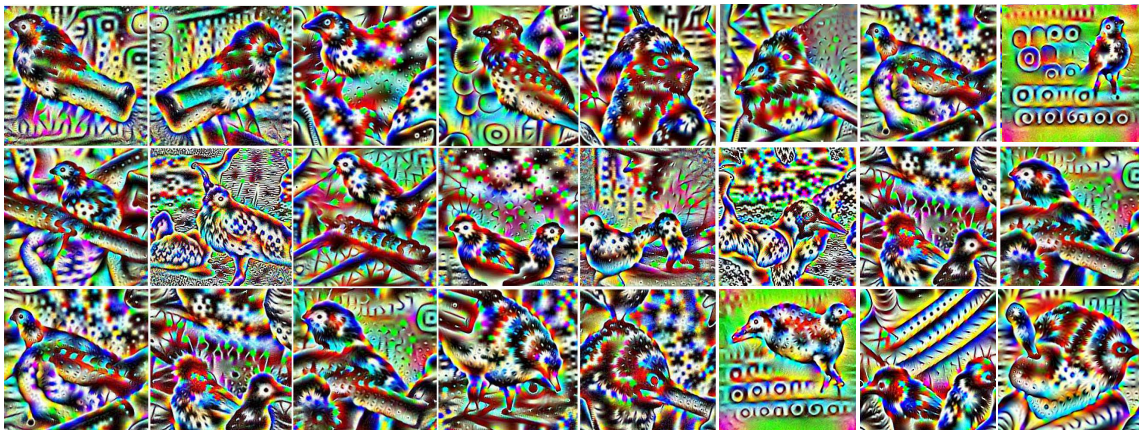

Figure 7.4: The generated representative images for CUB-200.

Table 7.5: Evaluation of the generated images

	Inception score	FID
Input random noise	0.93±0.01	401
Generated birds images	3.09±0.39	198
Origin birds images	5.24±0.30	0

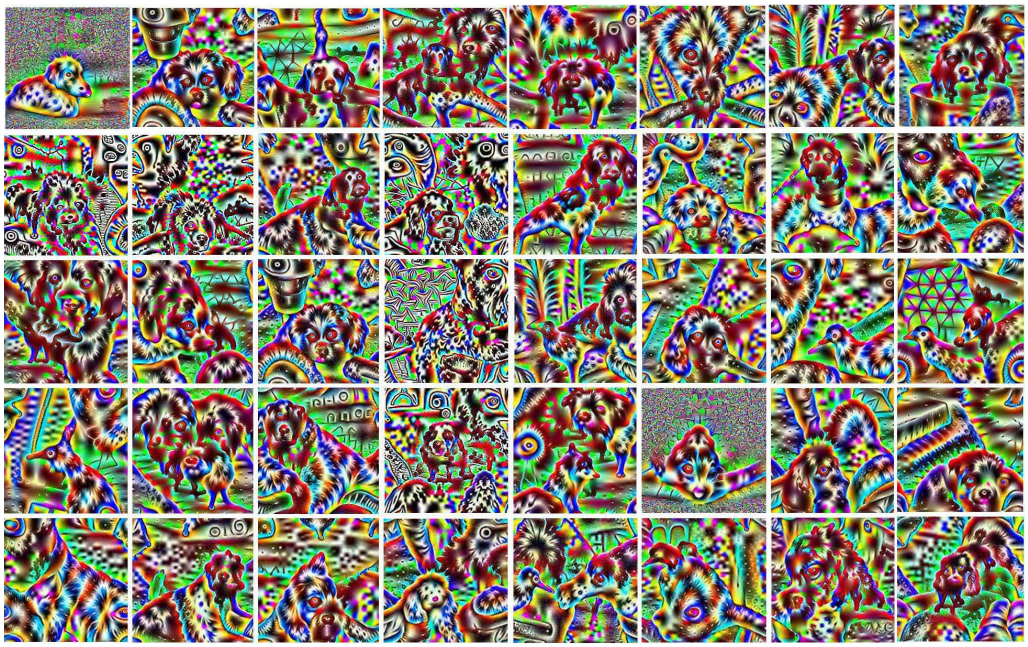

Figure 7.5: The generated images for the Stanford-Dogs dataset.

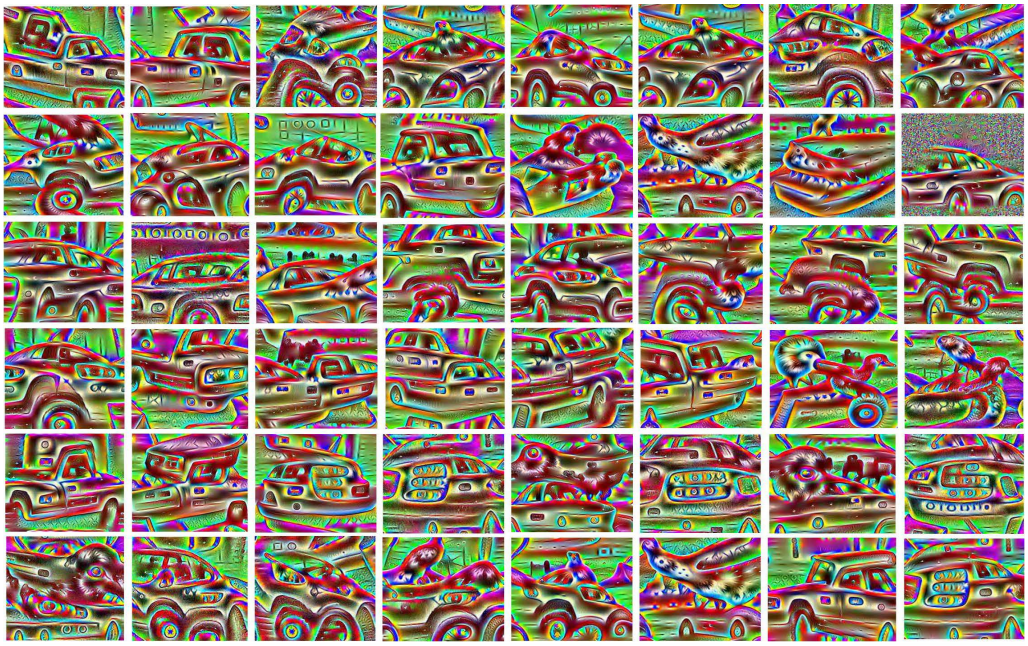

Figure 7.6: The generated images for the Stanford-Cars dataset.

as shown in Table 7.6. Consistent to previous experiments, we use the sequence of two tasks: CUB-200 → Stanford-Dogs. We build the fine-tuning method as a

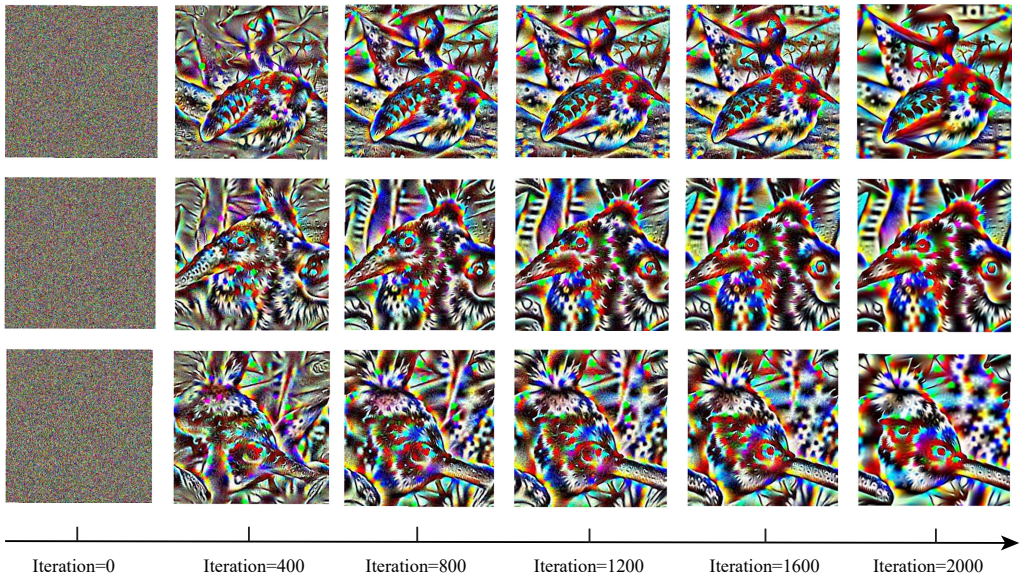

Figure 7.7: Illustration of image generation process on the CUB-200 dataset.

Baseline by using L_s only. As noted, the baseline model suffers from forgetting on the first task. **Case 1** is the knowledge distillation using L_{kd_1} from the frozen teacher only. As a result, the previously learned knowledge is transferred to the student (improving R@K=1 from 51.4% to 56.7% on CUB-200). To demonstrate the efficacy of BatchNorm statistics, we study **Case 2** where representative images are generated using $(L_{BN} + L_{cluster})$. Compared to *Case 1*, the student trained under this condition is prone to the first task and has its performance improved from 56.7% to 68.3% significantly, while performance on the second task degrade from 78.9% to 68.7%. **Case 3** is designed for the scenario where the self-motivated student is regularized only by the on-the-fly teacher when learns the second task. Consequently, the student improves on the second task (from 78.0% to 79.6%) and keeps the performance on the first task similar to the Baseline. We explore **Case 4** to study the importance of self-motivated learning of the student, which is regularized by dual knowledge distillation, but without using L_s. As a result, the student remembers the previous knowledge well and has a good generalization accuracy Recall@1 of 76.6% on the second task. Furthermore, **Case 5** refers to the network is regularized by two teachers but without using the BatchNorm statistics to enhance the frozen teacher. Compared to *Case 3*, the student improves its performance on the first task (*e.g.* from 50.8% to 56.9%), while the performance on the second task is kept unchanged. Finally, when the student is self-motivated to learn by using the term L_s, *i.e.* our DKD full method, whose generalization performance is improved from 76.6% in *Case 4* to 80.0% while the performance on the first task is close to the reference.

Table 7.6: Ablation study for lifelong image retrieval on the two-task setup. As defined in Eqs. 7.4 and 7.5, the representative image generation process is constrained by $L_g = L_{BN} + L_{cluster}$.

		CUB-200 → Stanford-Dogs					
		Test on CUB-200			Test on Stanford-Dogs		
		s	u	H	s	u	H
	Recall@K	K=1	K=1	K=1	K=1	K=1	K=1
Baseline	Fine-tuning by using L_s	56.0	47.5	51.4	72.2	84.9	78.0
Case 1	$L_s + L_{kd_1}$	62.2	52.1	56.7	73.6	85.0	78.9
Case 2	$L_s + L_{kd_1} + L_g$	73.5	63.8	68.3	60.0	80.3	68.7
Case 3	$L_s + L_{kd_2} + L_{te_2}$	55.1	47.1	50.8	74.0	86.2	79.6
Case 4	$L_{kd_1} + L_g + L_{kd_2} + L_{te_2}$	73.2	62.4	67.3	69.0	86.1	76.6
Case 5	$L_s + L_{kd_1} + L_{kd_2} + L_{te_2}$	59.7	54.5	56.9	74.0	86.2	79.6
Ours	$L_s + L_{kd_1} + L_g + L_{kd_2} + L_{te_2}$	72.0	62.7	67.0	74.4	86.5	80.0
Reference	Joint training by using L_s	74.1	64.1	68.7	74.5	86.7	80.1

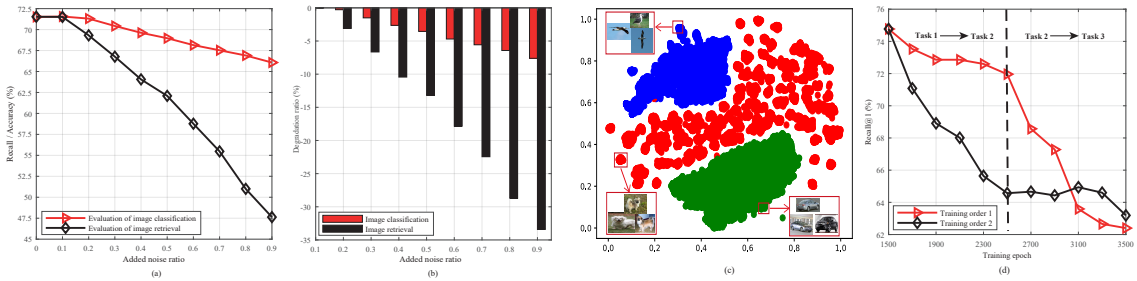

Figure 7.8: Sensitivity comparisons of image classification and image retrieval. (a) Recall rate / classification accuracy; (b) Performance degradation ratios for different noise ratios. (c) Dataset distributions visualization; (d) Performance evolution of two training orders, evaluated on the first task, *i.e.* on the CUB-200 dataset.

7.5.4 Further explorations

(1) Comparison with classification-based tasks. In terms of reducing forgetting, we observe that lifelong image retrieval is more challenging than classification-based tasks that focus on classification probabilities. The classification model is more stable, as long as image features of old data are classified within the range of prior boundaries, whereas image retrieval is sensitive to the matching between features. A small change in features would have a significant impact on feature matching. This makes the problem of minimizing forgetting more difficult. As a demonstration, we build an additional classifier on top of the fully-connected layers and use the LwF method [212] to train under the sequence: CUB-200 → Stanford-Dogs. During testing, we sample Gaussian noise from $\mathcal{N}(0, 0.1)$ and add it to each image, which affects the retrieval features and the final classification probabilities of the same model. We vary the ratio of the Gaussian noise and consider the evolution of retrieval recall and classification accuracy on the seen set of CUB-200. The results are reported in Figure 7.8. As can be seen, image retrieval task is more sensitive than image classification task for same levels of noise distraction.

(2) Training order exploration. We consider the training order 1: CUB-200 → Stanford-Dogs → Stanford-Cars in Table 7.3. To examine the effect of the task training order, we keep starting with CUB-200 and explore the other training order 2: CUB-200 → Stanford-Cars → Stanford-Dogs. We visualize all training samples of three datasets in Figure 7.8(c). For the two training orders, we evaluate the performance on the seen set of the first task (*i.e.* CUB-200) by using the model trained at the end of tasks (*i.e.* Stanford-Cars and Stanford-Dogs). The results are depicted in Figure 7.8(d). In general, the model suffers from performance degradation with respect to these two training orders. Due to the different distributions of datasets, the training order affects the performance greatly. In case of training order 1, the samples from Stanford-Dogs on task 2 are distributed closely to the samples from CUB-200. Therefore, the degradation during the "task 1 → task 2" session is relatively slow. However, the vehicle images from task 3 are distributed farther away from the bird images in task 1, which causes serious degradation during the "task 2 → task 3" session. In contrast, for training order 2, the performance degrades significantly from CUB-200 to Stanford-Cars during the "task 1 → task 2" session and whereas it becomes slow again during the "task 2 → task 3" session.

7.6 Chapter Conclusions

In this chapter, we explored image retrieval in a lifelong scenario and considered reducing catastrophic forgetting and simultaneously improving generalization performance. This goal is achieved by training a dual knowledge distillation framework to transfer previously learned knowledge and newly captured information. We used the stored statistics in the BatchNorm layers of the frozen teacher to generate representatives images to further reduce catastrophic forgetting on preceding tasks. The efficacy of the proposed method was demonstrated by thorough experimental results on three datasets. A limitation of this work is that the semantic drifts between training data in the task sequence still result in significant forgetting. In future work, more efficient approaches need to be investigated to realize lifelong image retrieval without forgetting. Furthermore, it would be very valuable to explore lifelong image retrieval on non-fine grained datasets or practical applications such as commercial shopping and recommendation systems.