Multi-Domain Multi-Task Rehearsal for Lifelong Learning

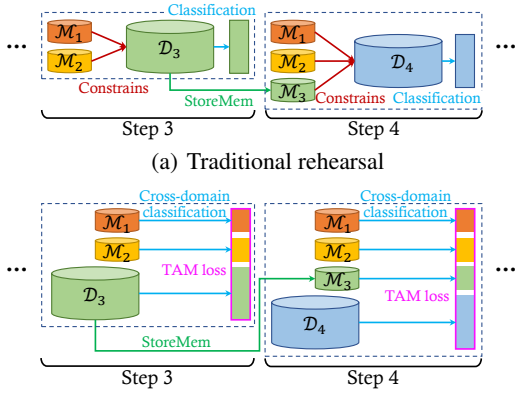
Fan Lyu[1], Shuai Wang[1], Wei Feng[1*], Zihan Ye[2], Fuyuan Hu[2] and Song Wang[1, 3]
[1]Colledge of Intelligence and Computing, Tianjin University
[2]School of Electronic & Information Engineering, Suzhou University of Science and Technology
[3]Department of Computer Science and Engineering, University of South Carolina
{fanlyu, wangshuai201909, wfeng}@tju.edu.cn, {zihanye@post, fuyuanhu@mail}.usts.edu.cn, songwang@cec.sc.edu

Abstract

Rehearsal, seeking to remind the model by storing old knowledge in lifelong learning, is one of the most effective ways to mitigate catastrophic forgetting, *i.e.*, biased forgetting of previous knowledge when moving to new tasks. However, the old tasks of the most previous rehearsal-based methods suffer from the unpredictable domain shift when training the new task. This is because these methods always ignore two significant factors. First, the Data Imbalance between the new task and old tasks that makes the domain of old tasks prone to shift. Second, the Task Isolation among all tasks will make the domain shift toward unpredictable directions; To address the unpredictable domain shift, in this paper, we propose Multi-Domain Multi-Task (MDMT) rehearsal to train the old tasks and new task parallelly and equally to break the isolation among tasks. Specifically, a two-level angular margin loss is proposed to encourage the intra-class/task compactness and inter-class/task discrepancy, which keeps the model from domain chaos. In addition, to further address domain shift of the old tasks, we propose an optional episodic distillation loss on the memory to anchor the knowledge for each old task. Experiments on benchmark datasets validate the proposed approach can effectively mitigate the unpredictable domain shift.

Introduction

Lifelong learning, also known as continual learning and incremental learning, aims to continually learn new knowledge from a sequence of tasks over a lifelong time. In contrast to traditional supervised learning, the lifelong setting helps machine learning work like a more realistic human learning by acquiring a new skill quickly with new training data. All the while, *catastrophic forgetting* (French 1999; Kirkpatrick et al. 2017) is the main challenge for lifelong learning, which happens when the learner forgets the knowledge of old tasks while learning a new task. To seek a balance between the old tasks and the new task, many methods have been proposed to handle the catastrophic forgetting in recent years. Following (De Lange et al. 2019), their methods can be categorized into *Rehearsal* (Lopez-Paz and Ranzato 2017; Chaudhry et al. 2018b; Guo et al. 2019), *Regularization* (Li and Hoiem 2016; Chaudhry et al. 2018a; Dhar et al. 2019) and *Parameter Isolation* (Mallya, Davis, and Lazebnik 2018; Yoon et al. 2017). Regularization-based and parameter isolation-based methods store no data from old tasks and highly rely on extra regularizers or architectures, resulting in their lower performance than the rehearsal-based methods. Rehearsal-based methods store a small number of samples in the training set, the model will retrain the saved data when training the new task to avoid forgetting.



(a) Traditional rehearsal

(b) MDMT rehearsal

Figure 1: (a) Traditional rehearsal-based methods construct single-task learning architecture for the new task (data from training set $\mathcal{D}$) and treat the old tasks (data from memory $\mathcal{M}$) as the constraints of its training. (b) The proposed MDMT rehearsal-based method trains old tasks and new task equally and keep tasks from isolation via TAM loss.

At each step of lifelong learning (see Fig. 1(a)), the most existing rehearsal-based methods (Rebuffi et al. 2017; Lopez-Paz and Ranzato 2017; Chaudhry et al. 2018b; Guo et al. 2019) focus on training the new task while treating the stored data from old tasks as the constraints to preserve their performance. However, the old tasks in these methods may suffer from *unpredictable domain shift* that arises from two significant factors in the lifelong learning process: 1) The *Data Imbalance* between old and new task. The shrinkage of training data of old tasks leads to their domains will be prone to shift that manifests as the catastrophic forgetting. 2) The *Task Isolation* among all tasks (old and new), which makes such domain shift toward unpredictable directions and the boundary between any two tasks may become weak.

To address the unpredictable domain shift, in this paper, we propose a Multi-Domain Multi-Task (MDMT) Rehearsal

*Corresponding Author.

method inspired by the multi-domain multi-task learning (Yang and Hospedales 2014) that considers both multiple tasks *w.r.t.* multiple domains and trains them equally. Specifically, as shown in Fig. 1(b), we first retrain the old tasks along with new task training parallelly rather than setting them as the constraints. We separate all these tasks by a Cross-Domain Softmax, which extends the softmax for each isolated task by combining the logits of all other seen tasks and separates them from each other. Then, to further alleviate the unpredictable domain shift, we propose to leverage a Two-level Angular Margin (TAM) loss to encourage the intra-class/task compactness and the inter-class/task discrepancy on the basis of Cross-Domain Softmax. In addition, we present an optional Episodic Distillation (ED) loss on all buffer memories for old tasks that suppress the domain shift by storing the latent representations of each sample in memories. We evaluate our MDMT rehearsal on four popular lifelong learning datasets for image classification and achieve new state-of-the-art performance. The experimental results show the proposed MDMT rehearsal can significantly mitigate the unpredictable domain shift. Our contributions are three-fold: (1) We propose a Multi-Domain Multi-Task Rehearsal method for lifelong learning, which parallelly and equally trains the old and new tasks and separate them by a Cross-Domain Softmax function. (2) We propose a Two-level Angular Margin (TAM) loss for lifelong learning to further boost the Cross-Domain Softmax for the sake of intra-class/task compactness and the inter-class/task discrepancy. (3) We build an optional Episodic Distillation loss to reduce the domain shift in lifelong progress.

Related Work

Lifelong Learning

In contrast to static machine learning (He et al. 2016; Deng et al. 2018; Lyu et al. 2019; Lyu, Feng, and Wang 2020), Lifelong Learning (Ring 1998; Thrun 1998) seeks to improve the self-learning ability of the machine that continually learns new knowledge. The previous solutions to the catastrophic forgetting (French 1999; Kirkpatrick et al. 2017) in recent years can be categorized into *regularization-based*, *parameter isolation-based* and *rehearsal-based* methods (De Lange et al. 2019). Regularization-based methods (Li and Hoiem 2016; Chaudhry et al. 2018a; Dhar et al. 2019) store no data but explore extra regularization terms in the loss function to consolidate previous knowledge. Parameter isolation-based methods (Mallya, Davis, and Lazebnik 2018; Yoon et al. 2017) freeze the task-specific parameters and grow new branches for new tasks to bring in new knowledge. Rehearsal-based methods store some knowledge of old tasks to remind the model and often achieve better performance. Existing methods can be categorized into three groups. 1) by saving the raw data (Rehearsal, *e.g.*, image) (Rebuffi et al. 2017; Lopez-Paz and Ranzato 2017; Chaudhry et al. 2018b; Guo et al. 2019), the model can retrain the saved data along with the current training; 2) by saving the latent features for selected samples (Latent-rehearsal) (Pellegrini et al. 2019), the model slows down learning at the layers below the rehearsal layer and leaves the layers above free to learn at full pace; 3) by building generative model to synthesize data (Pseudo-rehearsal) (Shen et al. 2020; van de Ven and Tolias 2018; Lesort et al. 2019), the knowledge can be saved as parameters rather than data. In this paper, we only consider the native rehearsal by storing raw data in image classification.

Multi-domain Multi-task Learning

Multi-domain learning (Nam and Han 2016; Tang and Jia 2020) refers to sharing information about the same problem across different contextual domains, while multi-task learning (Lin et al. 2019; Sener and Koltun 2018) addresses sharing information about different problems in the same domain. By considering both multiple domains and multiple tasks, Multi-domain multi-task (MDMT) learning was first proposed in (Yang and Hospedales 2014), and has been applied to classification (Peng and Dredze 2016) and semantic segmentation (Fourure et al. 2017), *etc.*. The common solution to MDMT problem is to construct parallel data streams and seek to build the correlations among tasks. Here, we explain why we decide to formulate the lifelong learning problem into a MDMT learning problem. 1) By storing some samples of a task into memory, MDMT learning can significantly train them together, which helps mitigate the task isolation in the traditional rehearsal-based lifelong learning. 2) MDMT learning can help suspending the domain shift to some extent by making classifiers perceive each other.

Margin Loss And Distillation Loss

The margin based Softmax explicitly adds a margin to each logit to improve feature discrimination. L-Softmax (Liu et al. 2016) and SphereFace (Liu et al. 2017) add multiplicative angular margin to squeeze each class. CosFace (Wang et al. 2018b,a) and ArcFace (Deng et al. 2019) add additive cosine margin and angular margin, respectively, for easier optimization. Based on ArcFace, we propose a Two-level Angular Margin loss to guarantee both inter-class/task compactness and intra-class/task discrepancy. The knowledge distillation (Hinton, Vinyals, and Dean 2015) transfers the knowledge about smoothed probability distribution of the output layer of the teacher network to the student network. Inspired by this, we propose to build distillation loss between the old and new models on old tasks by storing the latent representation of stored data.

Methodology

Multi-domain Multi-task Rehearsal

Suppose there are T different tasks with respect to datasets $\{\mathcal{D}_1, \cdots, \mathcal{D}_T\}$. For the t-th dataset (task), $\mathcal{D}_t = \{(x_{t,1}, y_{t,1}), \cdots, (x_{t,N_t}, y_{t,N_t})\}$, where $x_{t,i} \in \mathcal{X}_t$ is the i-th input data, $y_{t,i} \in \mathcal{Y}_t$ is the corresponding label and N_t is the number of samples. $\mathcal{D}_t$ can be split into a training set $\mathcal{D}_t^{\text{trn}}$ and a testing set $\mathcal{D}_t^{\text{tst}}$, and we denote $\mathcal{D}_t$ as $\mathcal{D}_t^{\text{trn}}$ in our presentation for simple denotation. Lifelong learning aims at learning a predictor $f_t : \mathcal{X}_k \rightarrow \mathcal{Y}_k,\ k \in \{1, \cdots, t\}$, which can predict tasks that have been learned at any time. The rehearsal-based lifelong learning (Rebuffi et al. 2017; Lopez-Paz and Ranzato 2017; Riemer et al. 2018; Chaudhry

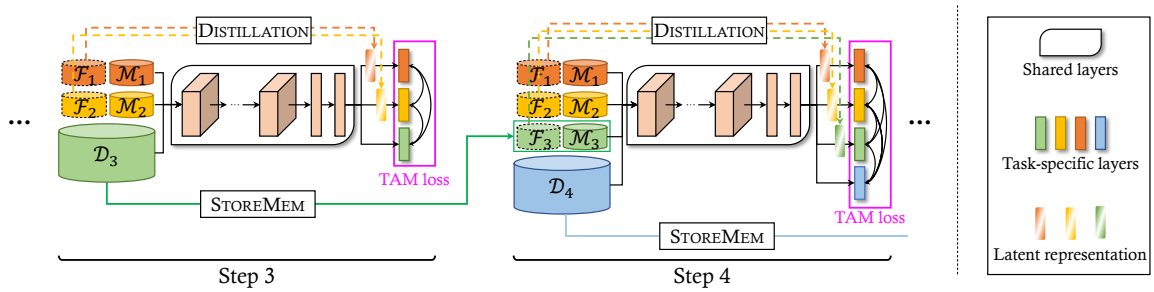


Figure 2: Training procedure of the proposed MDMT rehearsal based lifelong learning. At each step, a small number of samples will be saved into memory $\mathcal{M}$ and the corresponding latent representations will be saved into $\mathcal{F}$. TAM loss guarantee the intra-class/task compactness and inter-class/task discrepancy. Episodic Distillation loss helps further to reduce the domain shift of the old tasks. The dashed elements mean the optional operation.

et al. 2018b; Guo et al. 2019) builds a memory buffer $\mathcal{M}_k \subset \mathcal{D}_k$ with small-size for each previous task k, *i.e.*, $|\mathcal{M}_k| \ll |\mathcal{D}_k|$. Following (Lopez-Paz and Ranzato 2017), when training a task $t \in \{1, \cdots, T\}$, for all $\mathcal{M}_k$ that $k < t$, the rehearsal-based lifelong learning can be modeled as a single objective optimizing problem:

$$\begin{aligned} \arg\min_{\theta,\theta_t} \quad & \ell(f_\theta, f_{\theta_t}, \mathcal{D}_t), \\ \text{s.t.} \quad & \ell(f_\theta, f_{\theta_k}, \mathcal{M}_k) \le \ell(f_\theta^{t-1}, f_{\theta_k}^{t-1}, \mathcal{M}_k), \quad \forall k < t, \end{aligned} \tag{1}$$

where ℓ is the empirical loss. θ is the shared parameter across all tasks while θ_k and θ_t are the task-specific parameters. The constraints above are designed to prevent the performance degradation of previous tasks. Then, the problem can be reduced to find an optimal gradient that benefits all tasks. To inspect the increase in old tasks' loss, (Lopez-Paz and Ranzato 2017; Chaudhry et al. 2018b; Guo et al. 2019) compute the angle between the gradient of each old task and the proposed gradient update on the current task.

However, such a single objective optimization on the current task for rehearsal-based lifelong learning over-emphasizes the new task while ignoring the difference among tasks. In other words, the old tasks can only play the role of source domain to be transferred into the current training model. The domain of old tasks will significantly shift because of the rectified gradient that the gradient norm of new task is much larger than the old tasks', which may induce the domain overlap.

In contrast, this paper treats the problem as a Multi-Domain Multi-Task (MDMT) learning problem to jointly and equally improve the current task as well as the old tasks:

$$\begin{aligned} \arg_{\theta,\{\theta_1,\cdots,\theta_t\}} \quad & \{\min \ell(f_\theta, f_{\theta_t}, \mathcal{D}_t), \min \ell(f_\theta, f_{\theta_k}, \mathcal{M}_k), \cdots, \\ & \min \ell(f_\theta, f_{\theta_1}, \mathcal{M}_1)\}, \\ \text{s.t.} \quad & d(f_i, f_j) \ge d(f_i^{t-1}, f_j^{t-1}), i, j \in [1, t], i \ne j, \end{aligned} \tag{2}$$

where $f_i = f_\theta(\mathcal{D}_i)$ if $i = t$ and $f_i = f_\theta(\mathcal{M}_i)$ if $i < t$. d means the distance between two domains. For the t tasks w.r.t. datasets $\{\mathcal{D}_1, \cdots, \mathcal{D}_t\}$, a MDMT rehearsal model trains t tasks parallelly and equally. The constraints above mean the domain distance between any two tasks should not be smaller than the model trained on the last task. Note that we only consider the situation that the tasks are irrelevant as the common lifelong learning.

We make two key operations to solve the Eq.(2) efficiently. First, we transform the multi-objective optimization as a single-objective optimization problem by ensembling all these objectives as the traditional solution to multi-task learning (Lin et al. 2019; Sener and Koltun 2018).

$$\arg\min_{\theta} \quad \ell(f_\theta, f_{\theta_t}, \mathcal{D}_t) + \sum_{k=1}^{t-1} \ell(f_\theta, f_{\theta_k}, \mathcal{M}_k), \tag{3}$$

Second, it exists high memory-cost to calculate the distance between any two domains and store old predictors f_θ^{t-1}, but we can do this in a simple yet effective way by extending the softmax function for each task as

$$\ell_k = -\frac{1}{N_k} \sum_{n=1}^{N_k} \log \frac{e^{(W_{y_n}^k)^T x_n + b_{y_n}}}{\sigma_n}, \tag{4}$$

where

$$\sigma_n = \sum_{j=1}^{C_k} e^{(W_j^k)^T x_n + b_j} + \sum_{i=1, i\ne k}^{t} \sum_{j=1}^{C_i} e^{(W_j^i)^T x_n + b_j}. \tag{5}$$

N_k is the batch size for task k and $W_j^k \in \mathbb{R}^d$ denotes the j-th column of the weight $W^k \in \mathbb{R}^{d \times C_k}$ in the last fully-connected layer for task k and C_k is the class number. We name this extension as Cross-Domain Softmax (CDS), which combines the logits from other classifiers and is similar to a native softmax to a classification problem with total $\sum_{k=1}^{t} C_k$ class. Here, we discuss the difference. For MDMT rehearsal, different tasks never share a same classifiers as common classification, *i.e.*, the classifiers for different tasks lack mutual perception. By combining the logits form other tasks, the tasks can perceive and separate from each other. The previous methods update the model by the optimal gradient that highly rely on the angle between the gradients of old and new tasks. In contrast, we directly obtain the hybrid gradient for the shared layers by ensembling the gradients from the new task and old tasks as $\tilde{g} \leftarrow \sum_{k=1}^{t} g_k$.

We compare our MDMT rehearsal with several well-known rehearsal-based lifelong works:

iCaRL (Rebuffi et al. 2017) saves small number of samples to make the model not to forget old class, but they classify

Algorithm 1 MDMT rehearsal based lifelong learning.

Procedure TRAIN(f_θ, $f_{\theta_{1:T}}$, $\{\mathcal{D}_1^{\text{trn}}, \cdots, \mathcal{D}_T^{\text{trn}}\}$)
  $\mathcal{M}, \mathcal{F} \leftarrow \{\}, \{\}$
  for $t = 1$ **to** T **do**
    for $(\mathbf{x}, y) \in \mathcal{D}_t^{\text{trn}}$ **do**
      $g, g_1 \leftarrow \nabla_\theta \ell(f_\theta(\mathbf{x}, t), y)$
      if $t = 1$ **then**
        $\tilde{g} \leftarrow g$
      else
        $g^{\text{ref}}, g_{1:t-1} \leftarrow \nabla_\theta \ell(f_\theta, f_{\theta_{1:t-1}}, \mathcal{M})$
        $g^{\text{ref}} \leftarrow g^{\text{ref}} + \nabla_\theta \tilde{\ell}(f_\theta, \mathcal{F}_{\text{ref}})$
        $\tilde{g} \leftarrow g + g^{\text{ref}}$
      end if
      $\theta \leftarrow \theta - \text{StepSize} \cdot \tilde{g}$
      $\theta_{1:t} \leftarrow \theta_{1:t} - \text{StepSize} \cdot g_{1:t}$
    end for
    $\mathcal{M}, \mathcal{F} \leftarrow$ STOREMEM($\mathcal{M}, \mathcal{F}, \mathcal{D}_t^{\text{trn}}, f_\theta$)
  end for

Procedure STOREMEM($\mathcal{M}$, $\mathcal{F}$, $\mathcal{D}$, f)
  for $i = 1$ **to** $|\mathcal{M}|/T$ **do**
    $(\mathbf{x}, y) \sim \mathcal{D}$
    $\mathcal{M} \leftarrow \mathcal{M} + (\mathbf{x}, y)$
    $\mathcal{F} \leftarrow \mathcal{F} + f(\mathbf{x})$
  end for
  Return $\mathcal{M}$, $\mathcal{F}$

Procedure EVAL(f_θ, $f_{\theta_{1:T}}$, $\{\mathcal{D}_1^{\text{tst}}, \cdots, \mathcal{D}_T^{\text{tst}}\}$)
  $a \leftarrow 0 \in \mathbb{R}^T$
  for $t = 1$ **to** T **do**
    $a_t \leftarrow 0$
    for $(\mathbf{x}, y) \in \mathcal{D}_t^{\text{tst}}$ **do**
      $a_t \leftarrow a_t + \text{Accuracy}(f_{\theta_t}(f_\theta(\mathbf{x}, t)), y)$
    end for
    $a_t \leftarrow a_t / |\mathcal{D}_t^{\text{tst}}|$
  end for
  Return a

samples by the nearest prototype, which is not suitable for task-incremental lifelong learning because the task-specific parameters are ignored.

GEM/A-GEM (Lopez-Paz and Ranzato 2017; Chaudhry et al. 2018b) propose to solve forgetting by finding the optimal gradient that saves the old tasks from being corrupt, and they focus on training the new task with single objective optimization while ignore the domain shift of old tasks.

ER (Chaudhry et al. 2019a) extends Experience Replay (Rolnick et al. 2019) for reinforcement lifelong learning and be proven better than A-GEM. However, they never consider the relations among all tasks, which makes the domains of old task may significantly shift.

PRD (Hou et al. 2018) proposes to treat lifelong learning as a multi-task learning problem and proposes to build a distillation module with one saved CNN expert as teacher for each old task. Differently, we would like to build a MDMT rehearsal that leverage the expanded softmax without saving many extra models.

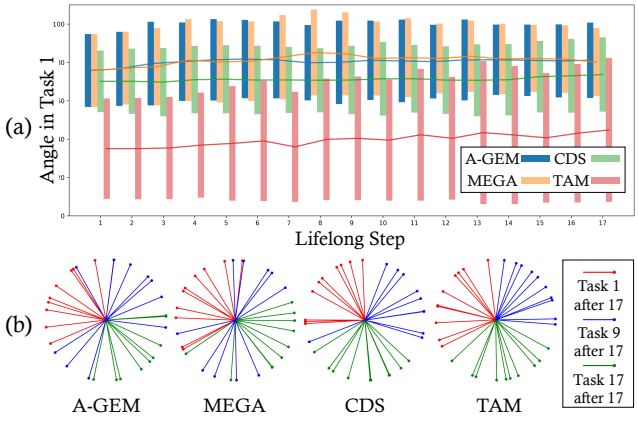


Figure 3: On Permuted MNIST, (a) the changes of angle range between feature and the target weight center of task 1 along the lifelong learning; (b) the angulars relations of class centers of task 1, 9 and 17 after trained on task 17.

Two-level Angular Margin Loss

The proposed MDMT rehearsal helps to jointly and equally train the new task and retrain the old tasks, making all tasks perceive each other. Nonetheless, the softmax loss is not efficient enough because it does not explicitly encourage intra-class compactness and inter-class discrepancy, in coping with which, large margin based softmax is widely used in recent discriminative problems (Deng et al. 2019; Liu et al. 2016). However, these methods cannot be directly applied to MDMT rehearsal based lifelong learning because these methods place the large margin only to single task and can not be applied to multiple tasks scenario.

In this paper, we propose two levels margin, *i.e.*, *class level* and *task level*, on softmax for each task (Eq. (4)). Our work is based on the popular large margin based softmax method Arcface (Deng et al. 2019) where the large margin is added to the angle between weight and feature, which has been proven effective and efficient. Specifically, Arcface deletes the bias and transforms the logit fed into the softmax as $W_j^T x_i = \|W_j\| \|x_i\| \cos\theta_j$, where θ_j is the angle between the weight W_j and the feature x_i, then an angular margin m is placed between different classes

$$\ell = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{e^{s\cdot\cos(\theta_{y_i}+m)}}{e^{s\cdot\cos(\theta_{y_i}+m)} + \sum_{j=1,j\neq y_i}^{n} e^{s\cdot\cos\theta_j}}, \quad (6)$$

where the individual weight $||W_j||$ is fixed to 1 by l_2 normalization and the embedding feature $||x_i||$ is fixed to s by l_2 normalization and rescale. The normalization on features and weights makes the predictions only depend on the angle between them. Such a geodesic distance margin between the sample and centers makes the prediction gain more intra-class compactness and inter-class discrepancy.

Based on Eq. (6), we propose our Two-level Angular Margin (TAM) loss for the task $k \in [1, t]$

$$\ell_k = -\frac{1}{N_k}\sum_{n=1}^{N_k} \log \frac{e^{s\cdot\cos\left((\theta^k_{y_n}+m^{\text{c}})+m^{\text{t}}\right)}}{\sigma_n}, \quad (7)$$

where

$$\sigma_n = e^{s\cdot\cos\left((\theta^k_{y_i}+m^{\mathrm{c}})+m^{\mathrm{t}}\right)} + \sum_{j=1,j\neq y_i}^{C_k} e^{s\cdot\cos(\theta^k_j+m^{\mathrm{t}})} + \sum_{i=1,i\neq k}^{t}\sum_{j=1}^{C_i} e^{s\cdot\cos\theta^i_j}. \tag{8}$$

In Eq. (7), we add class-level margin m^{c} and task-level margin m^{t} on the angular. m^{c} is similar to m in Eq. (6), which controls the intra-task class compactness and discrepancy (Deng et al. 2019). m^{t} controls the task compactness and discrepancy, which ensures the knowledge of each task not to mix up with others.

As shown in Fig. 3, the proposed TAM loss produces two advantages for MDMT rehearsal based lifelong learning. First, TAM helps the model to better discriminate into a task. Although the CDS has a better angle between feature and its target weight, TAM loss even reduce the angle to a smaller than CDS, which expresses the effect of m_{c}. Second, TAM loss mitigates the domain overlap caused by the domain shift by forcing tasks to separate. We can also see that for the angles among weights center, TAM loss can significantly separate old and new tasks, which expresses the effect of m_{t}. However, it is still difficult to omit the domain shift because of the extreme data imbalance between old tasks and new task. Thus, we construct an optional Episodic distillation loss for the MDMT rehearsal based lifelong process.

Episodic Distillation

In this paper, we propose a simple yet effective solution to further mitigate the domain shift for old task named Episodic Distilllation (ED) loss. The main role the ED loss played is to reduce the feature distribution change along with the lifelong process as far as possible. First, apart from the sampled training data stored in memory, *i.e.*, $\mathcal{M}_k = \{(x_{k,1}, y_{k,1}), \cdots, (x_{k,|M_k|}, y_{k,|M_k|})\} \subset \mathcal{D}_k$, we also store the corresponding latent representations when they are first trained, denoted as $\mathcal{F}_k = \{\mathbf{f}_{k,1}, \cdots, \mathbf{f}_{k,|M_k|}\}$. Then, we train the model with an updated objective:

$$\arg\min_{\theta} \quad \ell(f_\theta, f_{\theta_t}, \mathcal{D}_t) + \sum_{k=1}^{t-1}\left[\ell(f_\theta, f_{\theta_k}, \mathcal{M}_k) + \tilde{\ell}(f_\theta, \mathcal{F}_k)\right], \tag{9}$$

where

$$\tilde{\ell}(f_\theta, \mathcal{F}_k) \triangleq \frac{1}{N_k}\sum_i \tilde{\ell}_i(f_\theta(\mathbf{x}_{k,i}), \mathbf{f}_{k,i}). \tag{10}$$

$\tilde{\ell}_i$ is the ED loss that can be in many formats, and we choose the Mean Square Error (MSE). By training with Eq. (9) in each step, we can ease the shift effectively.

ED loss is an optional loss function and builds extra memory buffers to save the latent representation for each sample in memories. The extra memory buffers do increase the memory cost to some extent, but still very small in compared with the whole training set. In our implementation, we save the representation from the fc layer before the last one, which is a vector with length from 256 to 2048 for different network. That means the cost of the representation memory is even smaller than the data memory.

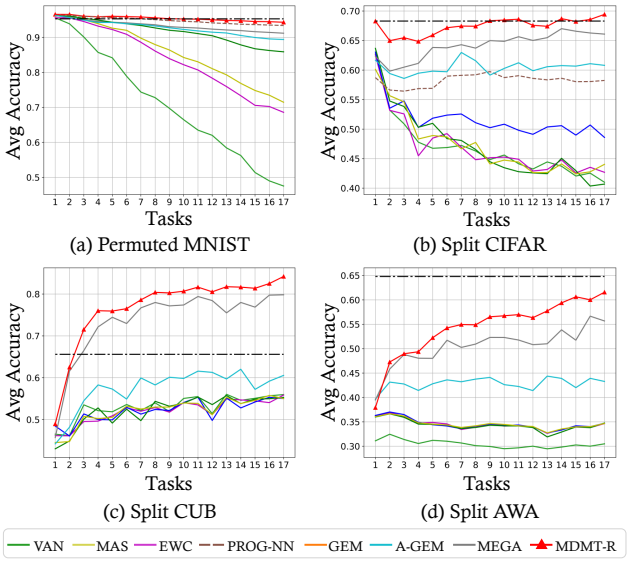


Figure 4: Average accuracy trend (from A_1 to A_T) on four datasets in the lifelong process.

Total Algorithm

We follow A-GEM (Chaudhry et al. 2018b) that unite memory of all old tasks for efficient training. Let $\mathcal{M} = \cup_{k<t}\mathcal{M}_k$ and $\mathcal{F} = \cup_{k<t}\mathcal{F}_k$ be the united data and representation memory for old tasks. For each step, we will sample a batch of data from the united memory. In this way, the previous tasks will be optimized by an average gradient instead of all gradients for previous tasks, which speeds up the training.

We show the detailed process in Algorithm 1 including training and evaluation procedure. First, the storage of memory feature in StoreMem to be as the anchor of old task in current task training. Second, the gradient to be updated depends not only the gradient on old and current tasks using TAM loss, but the gradient on feature difference by ED loss. The evaluation procedure is similar with the previous works.

Experiments

Experimental Settings

We evaluate the proposed method on four image recognition datasets. (1) *Permuted MNIST.* (Kirkpatrick et al. 2017): this is a variant of standard MNIST dataset of handwritten digits with 20 tasks. Each task has a fixed random permutation of the input pixels which is applied to all the images of that task. (2) *Split CIFAR.* (Zenke, Poole, and Ganguli 2017): this dataset consists of 20 disjoint subsets of CIFAR-100 dataset (Krizhevsky, Hinton et al. 2009), where each subset is formed by randomly sampling 5 classes without replacement from the original 100 classes. (3) *Split CUB.* (Chaudhry et al. 2018b): the CUB dataset (Wah et al. 2011) is split into 20 disjoint subsets by randomly sampling 10 classes without replacement from the original 200 classes. (4) *Split AWA.* (Chaudhry et al. 2018b): this dataset consists

Method	Permuted MNIST				Split CIFAR			
	$A_{\mathrm{T}}(\%)$	F_{T}	LCA_{10}	LTR	$A_{\mathrm{T}}(\%)$	F_{T}	LCA_{10}	LTR
Joint	95.30	-	-	-	68.30	-	-	-
VAN	47.55 ± 2.37	0.52 ± 0.026	0.259 ± 0.005	5.375 ± 0.194	40.44 ± 1.02	0.27 ± 0.006	0.309 ± 0.011	2.613 ± 0.174
EWC	68.68 ± 0.98	0.28 ± 0.010	0.276 ± 0.002	3.292 ± 0.135	42.67 ± 4.24	0.26 ± 0.039	0.336 ± 0.010	2.493 ± 0.427
MAS	70.30 ± 1.67	0.26 ± 0.018	0.298 ± 0.006	-	42.35 ± 3.52	0.26 ± 0.030	0.332 ± 0.010	-
RWalk	85.60 ± 0.71	0.08 ± 0.007	$\mathbf{0.319} \pm 0.003$	-	42.11 ± 3.69	0.27 ± 0.032	0.334 ± 0.012	-
MER	-	-	-	-	37.27 ± 1.68	$\mathbf{0.03} \pm 0.030$	0.051 ± 0.101	-
GEM	89.50 ± 0.48	0.06 ± 0.004	0.230 ± 0.005	-	61.20 ± 0.78	0.06 ± 0.007	0.360 ± 0.007	-
A-GEM	89.32 ± 0.46	0.07 ± 0.004	0.277 ± 0.008	0.716 ± 0.048	61.28 ± 1.88	0.09 ± 0.018	0.350 ± 0.013	0.643 ± 0.124
ER	90.47 ± 0.14	0.03 ± 0.001	0.184 ± 0.004	0.367 ± 0.013	63.97 ± 1.30	0.06 ± 0.006	0.349 ± 0.105	0.451 ± 0.333
MEGA	91.21 ± 0.10	0.05 ± 0.001	0.283 ± 0.004	0.524 ± 0.017	66.12 ± 1.94	0.06 ± 0.015	$\mathbf{0.375} \pm 0.012$	0.356 ± 0.114
MDMT-R	$\mathbf{94.33} \pm 0.04$	$\mathbf{0.02} \pm 0.000$	0.298 ± 0.003	$\mathbf{0.247} \pm 0.009$	$\mathbf{69.20} \pm 1.60$	0.04 ± 0.010	0.334 ± 0.008	$\mathbf{0.283} \pm 0.099$

Method	Split CUB				Split AWA			
	$A_{\mathrm{T}}(\%)$	F_{T}	LCA_{10}	LTR	$A_{\mathrm{T}}(\%)$	F_{T}	LCA_{10}	LTR
Joint	65.60	-	-	-	64.80	-	-	-
VAN	53.89 ± 2.00	0.13 ± 0.020	0.292 ± 0.008	0.976 ± 0.215	30.35 ± 2.81	0.04 ± 0.013	0.214 ± 0.008	0.202 ± 0.090
EWC	53.56 ± 1.67	0.14 ± 0.024	0.292 ± 0.009	1.021 ± 0.210	33.43 ± 3.07	0.08 ± 0.021	0.257 ± 0.011	0.675 ± 0.214
MAS	54.12 ± 1.72	0.13 ± 0.013	0.293 ± 0.008	-	33.83 ± 2.99	0.08 ± 0.022	0.257 ± 0.011	-
RWalk	54.11 ± 1.71	0.13 ± 0.013	0.293 ± 0.009	-	33.63 ± 2.64	0.08 ± 0.023	0.258 ± 0.011	-
PI	55.04 ± 3.05	0.12 ± 0.026	0.292 ± 0.010	-	33.86 ± 2.77	0.08 ± 0.022	0.259 ± 0.011	-
A-GEM	61.82 ± 3.72	0.08 ± 0.021	0.302 ± 0.011	0.456 ± 0.174	44.95 ± 2.97	0.05 ± 0.014	0.287 ± 0.012	0.178 ± 0.082
ER	73.63 ± 0.52	$\mathbf{0.01} \pm 0.005$	0.265 ± 0.004	$\mathbf{0.001} \pm 0.001$	54.27 ± 4.05	$\mathbf{0.02} \pm 0.030$	0.293 ± 0.009	0.014 ± 0.015
MEGA	80.58 ± 1.94	$\mathbf{0.01} \pm 0.017$	0.311 ± 0.010	0.002 ± 0.002	54.28 ± 4.84	0.05 ± 0.040	$\mathbf{0.305} \pm 0.015$	0.070 ± 0.114
MDMT-R	$\mathbf{84.27} \pm 1.63$	$\mathbf{0.01} \pm 0.015$	$\mathbf{0.337} \pm 0.013$	0.017 ± 0.014	$\mathbf{61.56} \pm 3.36$	$\mathbf{0.02} \pm 0.027$	0.298 ± 0.008	$\mathbf{0.002} \pm 0.002$

Table 1: Comparison with different state-of-the-arts. The numbers are averaged across 5 runs using a different seed each time.

m_{t}	m_{c}	ED	$A_{\mathrm{T}}(\%)$	F_{T}	LCA_{10}	LTR
-	-	-	65.44 ± 1.13	0.052 ± 0.006	$\mathbf{0.371} \pm 0.008$	0.377 ± 0.076
-	-	✓	66.44 ± 2.22	0.050 ± 0.009	0.370 ± 0.014	0.307 ± 0.066
0.0	0.0	-	67.15 ± 2.02	0.053 ± 0.012	0.353 ± 0.006	0.411 ± 0.097
0.1	0.0	-	67.49 ± 1.55	$\mathbf{0.049} \pm 0.010$	0.354 ± 0.005	0.369 ± 0.096
0.0	0.01	-	67.45 ± 1.09	0.059 ± 0.008	0.354 ± 0.005	0.483 ± 0.064
0.1	0.01	-	67.68 ± 1.72	0.052 ± 0.008	0.350 ± 0.007	0.390 ± 0.062
0.4	0.01	-	67.28 ± 0.97	0.053 ± 0.012	0.347 ± 0.006	0.394 ± 0.106
0.4	0.05	-	66.68 ± 1.23	0.063 ± 0.005	0.333 ± 0.005	0.473 ± 0.077
0.4	0.1	-	64.97 ± 1.13	0.084 ± 0.009	0.324 ± 0.008	0.680 ± 0.086
0.1	0.01	✓	$\mathbf{68.64} \pm 1.35$	0.059 ± 0.016	0.334 ± 0.008	$\mathbf{0.297} \pm 0.103$

Table 2: Ablation study on Split CIFAR.

of 20 subsets of the AWA dataset (Lampert, Nickisch, and Harmeling 2009). Each subset is constructed by sampling 5 classes with replacement from a total of 50 classes and the same class can appear in different subsets.

We leverage four existing metrics to evaluate the performance and catastrophic forgetting. (1) *Average Accuracy* ($A_t \in [0, 1]$) after the model has been trained continuously done till task $t \in \{1, \cdots, T\}$. In particular, A_T is the average accuracy on all the tasks after the last task has been learned. (2) *Forgetting Measure* (Chaudhry et al. 2018a) ($F_t \in [-1, 1]$) is the average forgetting after the model has been trained continuously with all the mini-batches for task $t \in \{1, \cdots, T\}$. (3) *Learning Curve Area* (Chaudhry et al. 2018a). ($LCA \in [0, 1]$) is the area of the convergence curve for any average b-shot performance after the model has been trained for all the T tasks, where $b \in [0, \beta]$. (4) *Long-Term Remembering* (Guo et al. 2019). ($LTR \geq 0$) LTR quantifies the accuracy drop on each task relative to the accuracy just right after the task has been learned. The detailed descriptions and the formulas can be shown in the supplementary materials.

Following the previous works (Lopez-Paz and Ranzato 2017; Chaudhry et al. 2018b; Guo et al. 2019), for Permuted MNIST we adopt a standard fully-connected network with two hidden layers, where each layer has 256 units with ReLU activation. For Split CIFAR we use a reduced ResNet18 (He et al. 2016). For Split CUB and Split AWA, we use a standard ResNet18.

Comparison With The State-of-the-arts

We compare the proposed method with the state-of-the-art methods including EWC (Kirkpatrick et al. 2017), MAS (Aljundi et al. 2018), RWalk (Chaudhry et al. 2018a), PI (Zenke, Poole, and Ganguli 2017), GEM (Lopez-Paz and Ranzato 2017), MER (Riemer et al. 2018), ER (Chaudhry et al. 2019b) A-GEM (Chaudhry et al. 2018b) and MEGA (Guo et al. 2019). Specifically, EWC, MAS, RWalk and PI are regularization-based methods that prevent the important weights from changing too much. GEM, MER, ER, AGEM and MEGA are rehearsal-based methods that rectifies the gradient guided by the stored data. VAN is a single supervised model trained continuously on the sequence of tasks. We also compare with the baseline that jointly trains all datasets with different classifiers together.

First, as shown in Tab. 1, the quantitative results of the proposed method outperform other state-of-the-arts. For A_T, the performances of our method show the superiority on all four datasets. This indicates the less forgetting on old

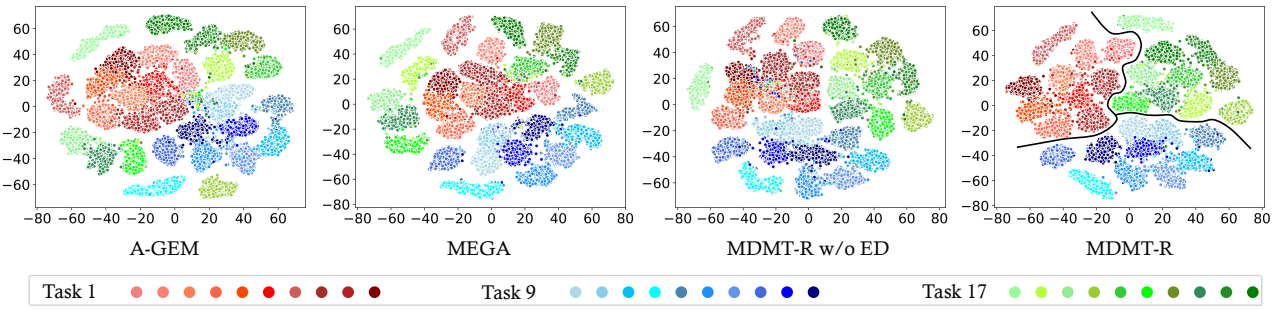


Figure 5: Final t-SNE of the features extracted from task 1, 9 and 17 on Permuted MNIST after the training on task 17.

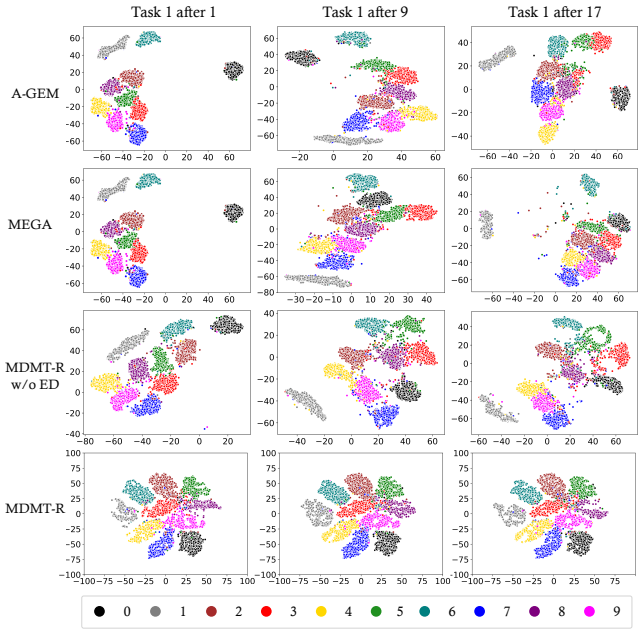


Figure 6: t-SNE of the features from task 1 on Permuted MNIST after the lifelong learning on task 1, 9 and 17.

tasks and better learning on new tasks through the lifelong training by reducing unpredictable domain shift. F_T evaluates the fine-grained batch-level forgetting on all tasks and never cares the Acc value. We get good F_T except on Split CIFAR with slight worse (0.04 vs. 0.03) than MER. MER has a better F_T but poor A_T because it adopts a complex meta learning strategy. For LCA_{10}, it evaluates the training speed on the first 10 training batches for each task, our method has the best LCA_{10} only on Split CUB. This is because the TAM and ED losses may slow the early training to mitigate domain overlap, but the following training will be improved significantly. LTR focuses on long-term remembering and our method outperforms other methods on these datasets except Split CUB. We think this is because the dataset CUB contains similar classes of birds, which means less impact of TAM and ED losses because of similar representations. In Fig. 4, we show the average accuracy trends in the continual process (from A_1 to A_T), which also indicate the better performance of the MDMT-R.

In Tab. 2, we then analyze the importance of the main components including TAM and ED loss on Split CIFAR. The first row is the results with only vanilla softmax. By adding ED loss, the average accuracy gets a little improvement. By adding TAM loss, the performance obtains larger gains, and we select the best m_t and m_c as the hyperparameters where $m_t = 0$ and $m_c = 0$ means the Cross-Domain Softmax. By adding both TAM and ED loss, we obtain a dramatic improvement in performance compared to the vanilla softmax and the state-of-the-art methods, which means the TAM and ED loss can significantly reduce the forgetting.

Domain Shift Observation

In this section, we would like to show some observations of domain shift using t-distributed Stochastic Neighbor Embedding (t-SNE) (Maaten and Hinton 2008) on Permuted MNIST. First, in order to intuitively reflect the task relation of the proposed method during the training process, we visualize the final feature distribution, *i.e.*, trained after the task 17, of task 1, 9 and 17 in Fig. 5. A-GEM and MEGA cannot guarantee the task boundaries, which means generating some mix area and makes the task easy to misclassify. The proposed MDMT rehearsal separates each class in three task while obtain explicit task boundary, which means the proposed method is able to encourage the intra-class/task compactness and inter-class/task discrepancy. As shown in Fig. 6, we also show the domain shift of task 1 after the model trained on task 1, 9 and 17, respectively. The previous methods A-GEM and MEGA cannot reduce the domain shift at all, which makes them sustainable to forget. The proposed MDMT rehearsal method can significantly mitigate the unpredictable domain shift. Without ED loss, our MDMT rehearsal still gets some unpredictable domain shift (such as task 1 after 1 and 9) because of the shrink of training data.

Conclusion

In this paper, we address catastrophic forgetting by considering the unpredictable domain shift of old tasks in the training sequence. To this end, we proposed a Multi-Domain Multi-Task rehearsal method, which effectively makes all tasks perceive each other. Then we proposed a Two-level Angular Margin loss to further encourage the intra-class/task compactness and inter-class/task discrepancy. Finally, an optional Episodic Distillation loss was proposed to mitigate domain shift. We have tested the proposed approach on four image classification benchmark datasets. Extensive experiments show the superiority of our approach over state-of-the-art methods.

Acknowledgments

This work was supported by the Natural Science Foundation of China (Nos. 62072334, 61671325, 61876121, 61672376 and U1803264) and Jiangsu Provincial Key Research and Development Program (No. BE2017663). The authors would like to thank constructive and valuable suggestions for this paper from the experienced reviewers and AE.

References

Aljundi, R.; Babiloni, F.; Elhoseiny, M.; Rohrbach, M.; and Tuytelaars, T. 2018. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Chaudhry, A.; Dokania, P. K.; Ajanthan, T.; and Torr, P. H. 2018a. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Chaudhry, A.; Ranzato, M.; Rohrbach, M.; and Elhoseiny, M. 2018b. Efficient Lifelong Learning with A-GEM. In *International Conference on Learning Representations*.

Chaudhry, A.; Rohrbach, M.; Elhoseiny, M.; Ajanthan, T.; Dokania, P. K.; Torr, P. H.; and Ranzato, M. 2019a. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486* .

Chaudhry, A.; Rohrbach, M.; Elhoseiny, M.; Ajanthan, T.; Dokania, P. K.; Torr, P. H. S.; and Ranzato, M. 2019b. On Tiny Episodic Memories in Continual Learning. *arXiv preprint arXiv:1902.10486* .

De Lange, M.; Aljundi, R.; Masana, M.; Parisot, S.; Jia, X.; Leonardis, A.; Slabaugh, G.; and Tuytelaars, T. 2019. Continual learning: A comparative study on how to defy forgetting in classification tasks. *arXiv preprint arXiv:1909.08383* .

Deng, C.; Wu, Q.; Wu, Q.; Hu, F.; Lyu, F.; and Tan, M. 2018. Visual grounding via accumulated attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7746–7755.

Deng, J.; Guo, J.; Xue, N.; and Zafeiriou, S. 2019. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Dhar, P.; Singh, R. V.; Peng, K.-C.; Wu, Z.; and Chellappa, R. 2019. Learning without memorizing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Fourure, D.; Emonet, R.; Fromont, E.; Muselet, D.; Neverova, N.; Trémeau, A.; and Wolf, C. 2017. Multi-task, multi-domain learning: application to semantic segmentation and pose regression. *Neurocomputing* 251: 68–80.

French, R. M. 1999. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences* 3(4): 128–135.

Guo, Y.; Liu, M.; Yang, T.; and Rosing, T. 2019. Learning with Long-term Remembering: Following the Lead of Mixed Stochastic Gradient. *arXiv preprint arXiv:1909.11763* .

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the Knowledge in a Neural Network. *Computer Science* 14(7): 38–39.

Hou, S.; Pan, X.; Change Loy, C.; Wang, Z.; and Lin, D. 2018. Lifelong learning via progressive distillation and retrospection. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A. A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences* .

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images. Technical report, Citeseer.

Lampert, C. H.; Nickisch, H.; and Harmeling, S. 2009. Learning to detect unseen object classes by between-class attribute transfer. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*.

Lesort, T.; Gepperth, A.; Stoian, A.; and Filliat, D. 2019. Marginal replay vs conditional replay for continual learning. In *International Conference on Artificial Neural Networks*.

Li, Z.; and Hoiem, D. 2016. Learning Without Forgetting. In *European Conference on Computer Vision*.

Lin, X.; Zhen, H.-L.; Li, Z.; Zhang, Q.-F.; and Kwong, S. 2019. Pareto Multi-Task Learning. In *Advances in Neural Information Processing Systems*.

Liu, W.; Wen, Y.; Yu, Z.; Li, M.; Raj, B.; and Song, L. 2017. SphereFace: Deep Hypersphere Embedding for Face Recognition. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*.

Liu, W.; Wen, Y.; Yu, Z.; and Yang, M. 2016. Large-margin softmax loss for convolutional neural networks. In *ICML*.

Lopez-Paz, D.; and Ranzato, M. 2017. Gradient episodic memory for continual learning. In *Advances in neural information processing systems*.

Lyu, F.; Feng, W.; and Wang, S. 2020. vtGraphNet: Learning weakly-supervised scene graph for complex visual grounding. *Neurocomputing* 51–60.

Lyu, F.; Wu, Q.; Hu, F.; Wu, Q.; and Tan, M. 2019. Attend and imagine: Multi-label image classification with visual attention and recurrent neural networks. *IEEE Transactions on Multimedia* 1971–1981.

Maaten, L. v. d.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9(2605): 2579–2605.

Mallya, A.; Davis, D.; and Lazebnik, S. 2018. Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Nam, H.; and Han, B. 2016. Learning multi-domain convolutional neural networks for visual tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Pellegrini, L.; Graffieti, G.; Lomonaco, V.; and Maltoni, D. 2019. Latent replay for real-time continual learning. *arXiv preprint arXiv:1912.01100* .

Peng, N.; and Dredze, M. 2016. Multi-task domain adaptation for sequence tagging. *arXiv preprint arXiv:1608.02689* .

Rebuffi, S.-A.; Kolesnikov, A.; Sperl, G.; and Lampert, C. H. 2017. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2001–2010.

Riemer, M.; Cases, I.; Ajemian, R.; Liu, M.; Rish, I.; Tu, Y.; and Tesauro, G. 2018. Learning to learn without forgetting by maximizing transfer and minimizing interference. *arXiv preprint arXiv:1810.11910* .

Ring, M. B. 1998. CHILD: A first step towards continual learning. In *Learning to learn*, 261–292. Springer.

Rolnick, D.; Ahuja, A.; Schwarz, J.; Lillicrap, T.; and Wayne, G. 2019. Experience replay for continual learning. In *Advances in Neural Information Processing Systems*.

Sener, O.; and Koltun, V. 2018. Multi-task learning as multi-objective optimization. In *Advances in Neural Information Processing Systems*.

Shen, G.; Zhang, S.; Chen, X.; and Deng, Z.-H. 2020. Generative Feature Replay with Orthogonal Weight Modification for Continual Learning. *arXiv preprint arXiv:2005.03490* .

Tang, H.; and Jia, K. 2020. Discriminative Adversarial Domain Adaptation. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*.

Thrun, S. 1998. Lifelong learning algorithms. In *Learning to learn*. Springer.

van de Ven, G. M.; and Tolias, A. S. 2018. Generative replay with feedback connections as a general strategy for continual learning. *arXiv preprint arXiv:1809.10635* .

Wah, C.; Branson, S.; Welinder, P.; Perona, P.; and Belongie, S. 2011. The caltech-ucsd birds-200-2011 dataset. Technical report, California Institute of Technology.

Wang, F.; Cheng, J.; Liu, W.; and Liu, H. 2018a. Additive Margin Softmax for Face Verification. *IEEE Signal Process. Lett.* .

Wang, H.; Wang, Y.; Zhou, Z.; Ji, X.; Gong, D.; Zhou, J.; Li, Z.; and Liu, W. 2018b. CosFace: Large Margin Cosine Loss for Deep Face Recognition. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*.

Yang, Y.; and Hospedales, T. M. 2014. A Unified Perspective on Multi-Domain and Multi-Task Learning. In *International Conference on Learning Representations*.

Yoon, J.; Yang, E.; Lee, J.; and Hwang, S. J. 2017. Lifelong learning with dynamically expandable networks. *arXiv preprint arXiv:1708.01547* .

Zenke, F.; Poole, B.; and Ganguli, S. 2017. Continual learning through synaptic intelligence. *Proceedings of machine learning research* .