Abstract—Humans are capable of accumulating knowledge by sequentially learning different tasks, while neural networks fail to achieve this due to catastrophic forgetting problems. Most current incremental learning methods focus more on tackling catastrophic forgetting for traditional classification networks. Notably, however, embedding networks that are basic architectures for many downstream learning applications also suffer from this problem. Moreover, the most significant difficulty for continual embedding networks is that the relationships between the latent features and prototypes of previous tasks will be destroyed once new tasks have been learned. Accordingly, we propose a novel incremental method for embedding networks, called the disentangled representation translation (DRT) method, to obtain the discriminative class-disentangled features without reusing any samples of previous tasks and while avoiding the perturbation of task-related information. Next, a mask-guided module is specifically explored to adaptively change or retain the valuable information of latent features. This module enables us to effectively preserve the discriminative yet representative features in the disentangled translation process. In addition, DRT can easily and drastically preserve the discriminative class-disentangled features without reusing any samples of previous tasks, while avoiding the perturbation of task-related information. This module allows us to effectively mitigate the catastrophic forgetting problem for embedding networks.

Index Terms—Disentangled representation, embedding network, incremental learning, transfer learning.

I. INTRODUCTION

RECENTLY, deep learning methods [1], [2] develop rapidly and achieve impressive results. Due to the rapid growth of online data and constrained computational resources, incremental learning [3], [4], [5], [6], the goal of which is to gradually learn knowledge from a stream of tasks without forgetting previous tasks, has attracted significant attention in recent years. Under the incremental learning setting, only the data of the current task is available. Thus, in the sequential training process, the important parameters for previous tasks will change rapidly, leading to previously learned knowledge being forgotten; this is referred to as catastrophic forgetting [3], [7].

In an attempt to alleviate catastrophic forgetting, a large number of incremental learning methods have been proposed. These can be divided into three parts: storing training samples of previous tasks [4], [8] that leverage the stored examples with the knowledge of previous tasks and the current data to train the current model, constraining the updating of model parameters [9], [10] that enforce the models of different tasks to have similar outputs when given the same input, and leveraging generative models [11], [12] that generate samples of previous tasks to transfer the incremental learning to supervised learning.

However, the above-mentioned methods aim to tackle catastrophic forgetting in classification networks, which need to add new weights to accommodate for the newly added classes; notably, they are not suitable for embedding networks, which are typically employed for many tasks, including image hashing retrieval [13], [14], [15], zero-shot recognition [16], [17], [18], domain adaptation [19], [20], etc. Unlike traditional classification networks, embedding networks not only can be employed in these tasks, but also can be used for classification when combined with a nearest class mean (NCM) classifier, which successfully complete the inference process by measuring the distance between the test samples and the prototypes.

As for embedding networks, the appearance of catastrophic forgetting is semantic gap, which is shown in Fig. 1. The feature distribution of task $t-1$ is measured precisely after learning the data of task $t-1$. When finished learning the task $t$, the semantic representation in embedding space has been changed, leading the feature distribution of task $t-1$ becomes messed. Additionally, the examples of new classes are added into the embedding spaces, which also disturbs the classification for task $t-1$. Thus, to alleviate catastrophic forgetting in embedding networks, translating the examples from different embedding spaces into a common embedding space is a direct method used to bridge the semantic gap between two embedding spaces of two adjacent tasks. Inspired by this idea, semantic drift compensation (SDC) [21] was
In this article, we propose a novel method of alleviating catastrophic forgetting for embedding networks, dubbed as disentangled representation translation (DRT). By observing the relationships between prototypes and latent features in embedding spaces, we disentangle the latent features into different disentangled embedding spaces. To conduct the classification, we propose a DRT model between two adjacent tasks, which disentangles the latent features into two separate disentangled features and respective classifier to classify the new tasks. Moreover, some methods [24], [25], [26] employ representative learning to obtain the discriminative features of different classes to alleviate catastrophic forgetting. Furthermore, generative adversarial networks have been introduced into many incremental learning methods [11], [12] to generate samples for the previous tasks, achieving incremental learning into traditional supervised learning. SDC [21] introduces the problem of catastrophic forgetting in embedding networks; this approach approximates and compensates for the semantic drift simply after training new tasks. In a departure from these methods, the goal of our approach is to build a better classification space between two adjacent tasks to solve catastrophic forgetting in embedding networks. In addition, we leverage disentangled representation to obtain discriminative class-disentangled features without perturbation of the task-related information.

B. Disentangled Representation

The concept of disentangled representation refers to a learning pattern that decouples confused data into two separate parts. The advantage of disentangled representation is that it exploits the implicit or explicit knowledge of the data, and it is applied to many semisupervised methods. However, due to the lack of available labeled datasets, it is necessary to train the model to exploit the important factors of data in an unsupervised manner. Based on the variational autoencoder (VAE) [27], $\beta$-VAE [28] is effective and stable for unsupervised disentangling that requires a needs the reconstruction mechanism to obtain satisfactory disentangled quality. Subsequently, InfoGAN [29] is proposed to capture more discriminative disentangled representation by leveraging the mutual information between a small subset of latent variables and the observations. The Factor-VAE [30] penalizes the total correlation of the aggregated posterior. Moreover, the DIP-VAE variants [31] match the moments of the aggregated
to 0. For the embedding network, the objective function is triplet loss, following [32], which is denoted as follows:

$$\mathcal{L}_{emb} = \mathcal{L}_{tri}.$$  (2)

After the training process of task \( t \) is complete, we employ an NCM classifier to conduct classification in the embedding space, which is also denoted as the classification space of task \( t \). The classification process can be described as follows:

$$c^*_t = \arg \min_{c \in C} \text{dist}(z_j, u_c)$$  (3)

$$u_c = \frac{1}{n_c} \sum_i [y_i = c] z_i$$  (4)

where \( n_c \) represents the number of training samples for class \( c \), while \([y_i = c]\) is 1 if \( y_i = c \), and 0 otherwise. In addition, \( u_c \) represents the prototype of class \( C \) and is added into the prototype memory, which is the mean feature and contains the discriminative information of class \( c \).

When task \( t+1 \) arrives, the embedding network is fine-tuned on the new dataset \( D^{t+1} \), causing the parameters of the embedding network and the embedding space to change. We refer to the prototype of the previous class as \( u_{c^{t-1}} \) \((t > s)\), which is the mean feature for class \( c^t \) in the embedding space of task \( t-1 \). The trained model \( \theta_{t+1} \) is then employed to compute the prototypes of new classes and project the images into the embedding space to achieve the classification of all learned classes. This sequential learning pattern is denoted as embedding fine-tuning (E-FT).

C. Incremental Embedding Learning With Disentangled Representation

The reason why catastrophic forgetting occurs in embedding networks is that fine-tuning the trained network will change the important parameters and embedding spaces for the previous tasks. Thus, the prototypes of precious classes computed in previous classification spaces are employed in the current embedding space, which will destroy the relationship between the prototypes \( u_{c^{t-1}} \) and the examples \( z^*_t \). To alleviate catastrophic forgetting in embedding networks, the simplest and most effective approach is to estimate and compensate for the semantic gap between adjacent embedding spaces. In addition, the features from different embedding spaces will inescapably introduce task-related information, which leads the representation of class-related information variously and disrupt the classification. Thus, we employ disentangled representation to obtain class-disentangled features while avoiding the perturbation of task-disentangled features.

To obtain the disentangled representation, we construct the translation networks between two adjacent tasks. After the training stage of task \( t \) is complete, the latent features \( \tilde{z}^t \) and \( z^*_t \) are extracted from the embedding network \( \theta^{t-1} \) and \( \theta' \) with the same input image \( x^t \), respectively. As shown in Fig. 2, \( \tilde{z}^t \) and \( z^*_t \) represent the same class in different embedding spaces, while \( \tilde{z}^t \) and \( u_{c^{t-1}} \) represent different classes in the same embedding space. Thus, the three latent features contain all four important factors between the two embedding spaces: current class-related information, previous
To help preserve the discriminative information, respectively.

Class-Disentangled Space

\[ z_i^t \]

\[ m_i^{t-1} \]

\[ \tilde{m}_i^{t-1} \]

\[ o_i^t \]

\[ \tilde{o}_i^t \]

\[ m_i^{t-1} \]

\[ \tilde{m}_i^{t-1} \]

\[ o_i^t \]

\[ \tilde{o}_i^t \]

\[ m_i^{t-1} \]

\[ \tilde{m}_i^{t-1} \]

\[ o_i^t \]

\[ \tilde{o}_i^t \]

\[ m_i^{t-1} \]

\[ \tilde{m}_i^{t-1} \]

\[ o_i^t \]

\[ \tilde{o}_i^t \]

\[ m_i^{t-1} \]

\[ \tilde{m}_i^{t-1} \]

\[ o_i^t \]

\[ \tilde{o}_i^t \]

\[ m_i^{t-1} \]

\[ \tilde{m}_i^{t-1} \]

\[ o_i^t \]

\[ \tilde{o}_i^t \]

\[ m_i^{t-1} \]

\[ \tilde{m}_i^{t-1} \]

\[ o_i^t \]

\[ \tilde{o}_i^t \]

\[ m_i^{t-1} \]

\[ \tilde{m}_i^{t-1} \]

\[ o_i^t \]

\[ \tilde{o}_i^t \]

\[ m_i^{t-1} \]

\[ \tilde{m}_i^{t-1} \]

\[ o_i^t \]

\[ \tilde{o}_i^t \]

\[ m_i^{t-1} \]

\[ \tilde{m}_i^{t-1} \]

\[ o_i^t \]

\[ \tilde{o}_i^t \]

\[ m_i^{t-1} \]

\[ \tilde{m}_i^{t-1} \]

\[ o_i^t \]

\[ \tilde{o}_i^t \]

\[ m_i^{t-1} \]

\[ \tilde{m}_i^{t-1} \]

\[ o_i^t \]

\[ \tilde{o}_i^t \]

\[ m_i^{t-1} \]

\[ \tilde{m}_i^{t-1} \]

\[ o_i^t \]

\[ \tilde{o}_i^t \]

\[ m_i^{t-1} \]

\[ \tilde{m}_i^{t-1} \]

\[ o_i^t \]

\[ \tilde{o}_i^t \]

\[ m_i^{t-1} \]

\[ \tilde{m}_i^{t-1} \]

\[ o_i^t \]

\[ \tilde{o}_i^t \]

\[ m_i^{t-1} \]

\[ \tilde{m}_i^{t-1} \]

\[ o_i^t \]

\[ \tilde{o}_i^t \]

\[ m_i^{t-1} \]

\[ \tilde{m}_i^{t-1} \]

\[ o_i^t \]

\[ \tilde{o}_i^t \]

\[ m_i^{t-1} \]

\[ \tilde{m}_i^{t-1} \]

\[ o_i^t \]

\[ \tilde{o}_i^t \]

\[ m_i^{t-1} \]

\[ \tilde{m}_i^{t-1} \]

\[ o_i^t \]

\[ \tilde{o}_i^t \]

\[ m_i^{t-1} \]

\[ \tilde{m}_i^{t-1} \]

\[ o_i^t \]

\[ \tilde{o}_i^t \]

\[ m_i^{t-1} \]

\[ \tilde{m}_i^{t-1} \]

\[ o_i^t \]

\[ \tilde{o}_i^t \]

\[ m_i^{t-1} \]

\[ \tilde{m}_i^{t-1} \]

\[ o_i^t \]

\[ \tilde{o}_i^t \]

\[ m_i^{t-1} \]

\[ \tilde{m}_i^{t-1} \]

\[ o_i^t \]

\[ \tilde{o}_i^t \]

\[ m_i^{t-1} \]

\[ \tilde{m}_i^{t-1} \]

\[ o_i^t \]

\[ \tilde{o}_i^t \]

\[ m_i^{t-1} \]

\[ \tilde{m}_i^{t-1} \]

\[ o_i^t \]

\[ \tilde{o}_i^t \]

\[ m_i^{t-1} \]

\[ \tilde{m}_i^{t-1} \]

\[ o_i^t \]

\[ \tilde{o}_i^t \]

\[ m_i^{t-1} \]

\[ \tilde{m}_i^{t-1} \]

\[ o_i^t \]

\[ \tilde{o}_i^t \]

\[ m_i^{t-1} \]

\[ \tilde{m}_i^{t-1} \]

\[ o_i^t \]

\[ \tilde{o}_i^t \]

\[ m_i^{t-1} \]

\[ \tilde{m}_i^{t-1} \]

\[ o_i^t \]

\[ \tilde{o}_i^t \]

\[ m_i^{t-1} \]

\[ \tilde{m}_i^{t-1} \]

\[ o_i^t \]

\[ \tilde{o}_i^t \]

\[ m_i^{t-1} \]

\[ \tilde{m}_i^{t-1} \]

\[ o_i^t \]

\[ \tilde{o}_i^t \]

\[ m_i^{t-1} \]

\[ \tilde{m}_i^{t-1} \]

\[ o_i^t \]

\[ \tilde{o}_i^t \]

\[ m_i^{t-1} \]

\[ \tilde{m}_i^{t-1} \]

\[ o_i^t \]

\[ \tilde{o}_i^t \]

\[ m_i^{t-1} \]

\[ \tilde{m}_i^{t-1} \]

\[ o_i^t \]

\[ \tilde{o}_i^t \]

\[ m_i^{t-1} \]

\[ \tilde{m}_i^{t-1} \]

\[ o_i^t \]

\[ \tilde{o}_i^t \]

\[ m_i^{t-1} \]

\[ \tilde{m}_i^{t-1} \]

\[ o_i^t \]

\[ \tilde{o}_i^t \]

\[ m_i^{t-1} \]

\[ \tilde{m}_i^{t-1} \]

\[ o_i^t \]

\[ \tilde{o}_i^t \]

\[ m_i^{t-1} \]

\[ \tilde{m}_i^{t-1} \]

\[ o_i^t \]

\[ \tilde{o}_i^t \]

\[ m_i^{t-1} \]

\[ \tilde{m}_i^{t-1} \]

\[ o_i^t \]

\[ \tilde{o}_i^t \]

\[ m_i^{t-1} \]

\[ \tilde{m}_i^{t-1} \]

\[ o_i^t \]

\[ \tilde{o}_i^t \]

\[ m_i^{t-1} \]

\[ \tilde{m}_i^{t-1} \]

\[ o_i^t \]

\[ \tilde{o}_i^t \]

\[ m_i^{t-1} \]

\[ \tilde{m}_i^{t-1} \]

\[ o_i^t \]

\[ \tilde{o}_i^t \]

\[ m_i^{t-1} \]

\[ \tilde{m}_i^{t-1} \]

\[ o_i^t \]

\[ \tilde{o}_i^t \]

\[ m_i^{t-1} \]

\[ \tilde{m}_i^{t-1} \]

\[ o_i^t \]

\[ \tilde{o}_i^t \]

\[ m_i^{t-1} \]

\[ \tilde{m}_i^{t-1} \]

\[ o_i^t \]

\[ \tilde{o}_i^t \]

\[ m_i^{t-1} \]

\[ \tilde{m}_i^{t-1} \]

\[ o_i^t \]

\[ \tilde{o}_i^t \]
D. Mask-Guided Translation Network

In the translation process, the discriminative information would be lose, which will destroy the ability of prototypes to represent each of the classes. To preserve more discriminative and representative features, we design a mask-guided translation network to achieve the disentangling and reconstruction, which can be named as “MGT.”

For the disentangling process, the input feature \( z \) is projected by a network with two branches: one for the class-disentangled feature and the other for the task-disentangled feature, as shown in Fig. 3. For the class-disentangled feature, the translation network is also a network with two branches: one for generating the class feature \( v_{\text{class}} \) and the other for the guided mask \( g_{\text{class}} \). \( v_{\text{class}} \) is mapped by two fully connected layers, while \( g_{\text{class}} \) is projected by two fully connected layers and a softmax operation. The disentangled operation can be described as follows:

\[
m = v_{\text{class}} \ast g_{\text{class}} + z \ast (1 - g_{\text{class}}).
\]

(16)

The mask \( g_{\text{class}} \) is employed to guide the disentangling process to preserve more discriminative and representative prototypes. The same translation operation is employed for the process of the disentangling task.

As for the reconstruction process, the reconstruction network is also a two-branch network, with one branch for the class-disentangled feature and the other for the task-disentangled feature, as shown in Fig. 4. The class-disentangled feature \( m \) and task-disentangled feature \( o \) are projected and fused by two-branch mask-guided networks. The reconstruction operation can be denoted as follows:

\[
z_{\text{recon}} = m \ast g_{\text{re-class}} + o \ast g_{\text{re-task}}.
\]

(17)

E. Training and Inference

In the training stage, the embedding model is trained sequentially on different tasks, the training process of which is presented in Fig. 2. The training processes of the embedding network and disentangled translation network are iterative. Following the embedding network training process, which is regularized by \( L_{\text{emb}} \), we first add the prototypes of new classes into prototype memory. Then, we train the disentangled translation network to update the prototypes of both the new classes and old classes, which is regularized by \( L_{\text{tran}} \). The training process of incremental embedding learning with DRT is summarized in Algorithm 1: here, DRT comprises \( G_{t,\text{class}} \), \( G_{t,\text{task}} \), \( G_{t-1,\text{class}} \), and \( G_{t-1,\text{task}} \), whose parameters are optimized together.

In the testing stage, we first map all testing samples into the original classification space as latent features. Subsequently, all latent features of testing samples and class prototypes are projected into the class-disentangled space. The prototypes of previous classes are mapped by \( G_{t-1,\text{class}} \), and the prototypes of current task and all testing samples are mapped by \( G_{t,\text{class}} \). Finally, we can use an NCM classifier for classification, which is defined as follows:

\[
c_j^* = \arg \min_{c \in C} \text{dist}(m_j, u_c).
\]

(18)
IV. EXPERIMENTS

In this section, we describe the datasets, evaluation metrics, and implementation details. The experiments, which conduct comparisons with several state-of-the-art incremental methods, are presented with the goal of verifying the effectiveness of our proposed method. Finally, we conduct three groups of ablation studies to verify the effectiveness of different modules.

1) Datasets: CUB-200-2011 (CUB) [33], Flowers-102 (FLO) [34], Caltech-101 (CAL) [35] and CIFAR100 [36] are selected as the datasets for evaluating the performance of the methods. CUB, which contains 200 classes and 11788 images, is a popular fine-grained dataset for many embedding tasks, such as zero-shot learning, few-shot learning, and fine-grained classification. Flowers-102 consists of 102 flower categories and is also a fine-grained dataset. Caltech-101 consists of 9146 images belonging to 101 widely varied categories. CIFAR100, which contains 100 classes and 60000 images, is the typical dataset for class-incremental learning, and has been applied by many incremental methods.

2) Implementation Details: Following the experimental settings of [21], we select the ResNet-18 network, pretrained on ImageNet [40], and the ResNet-32 network [41] without pretraining, as the backbone networks for CUB and CIFAR100, respectively. In the training stage, the images are resized to 256 × 256 for CUB, FLO and CAL, and 32 × 32 for CIFAR100. Subsequently, the images of CUB, FLO, and CAL are randomly cropped and flipped. The epoch, learning rate and batch size are set to 50, 1e−5, and 32, respectively, for the four datasets. The dimension of latent features in embedding space is set to 512. In addition, all models are implemented using Pytorch and optimized via the Adam optimizer [42].

As for the disentangled translation network, both the translation network and recognition network are two-branch networks, each branch of which contains two fully connected layers, and the dimension of the hidden layer is 1024. The epochs and batch sizes are set to 150 and 128 for CUB, FLO, and CAL, as well as 70 and 128 for CIFAR100. In addition, the learning rate is set to 0.002 and the model is optimized via Adam. γ, β, δ, ρ, and η are set to 1000, 100, 100, 100, and 0.1 for CUB, FLO, and CAL, and 200, 100, 100, 100, and 0.1 for CIFAR100.

3) Baseline Methods:

a) E-FT: As described in Section III-A.

b) E-LwF [9]: The goal of embedding-learning without forgetting (E-LwF) is to enforce the output $z_i^t$ of the current embedding model to be similar to the output $z_i^{t−1}$ of the former embedding model when given the same input, which is denoted as follows:

$$L_{LwF} = \|z_i^t - z_i^{t−1}\|_2$$  \hspace{1cm} (19)

where $\|\|$ refers to the Frobenius norm.

c) E-EWC [22]: The goal of embedding-elastic weight consolidation (E-EWC) is to preserve the optimal parameters of the previous task without changes during the training process of the current task, the objective function of EWC is

$$L_{EWC} = \sum_{p} \frac{1}{2} F_{pt}^{-1} (\theta_p^t - \theta_p^{t−1})^2$$  \hspace{1cm} (20)

where $F_{pt}^{-1}$ is denoted as the Fisher information matrix, which is computed after the previous task $t−1$. In addition, the summation is conducted over all parameters $\theta_p$ of the network.

d) E-MAS [8]: The goal of embedding-memory aware synapses (E-MAS) is to compute the sensitivity of the predicted output function when the parameter changes, which can be denoted as follows:

$$L_{MAS} = \sum_{p} \frac{1}{2} \Omega_p (\theta_p^t - \theta_p^{t−1})^2$$  \hspace{1cm} (21)

where $\Omega_p$ is estimated with reference to the sensitivity of the squared $L_2$ norm of the function output to their changes.

These losses can be added directly to the objective loss of embedding networks to prevent forgetting while continually learning tasks, as follows:

$$\mathcal{L} = \mathcal{L}_{emb} + \phi \mathcal{L}_C$$  \hspace{1cm} (22)

where $\mathcal{L} \in \{ L_{W F}, EWC, MAS \}$, $\phi$ represents the trade-off between the embedding network loss and the losses, which are set to 1, 1e7, and 1e6, respectively.

e) SDC [21]: The goal of SDC is to approximate and compensate for the semantic drift of prototypes between two adjacent tasks, which can be flexibly applied via several incremental learning methods to further improve the performance. All results are implemented and measured in the same experimental environment.

4) Evaluation Metric: We employ the average incremental accuracy and average forgetting to evaluate the performance of the different methods. $a_{k,j} \in [0, 1]$ is denoted as the accuracy of the $j$th task ($j < k$) after the training stage of task $k$ is complete. The average incremental accuracy of task $k$ is defined as $A_k = (1/k) \sum_{j=1}^{k} a_{k,j}$. Average forgetting is defined to estimate the forgetting of previous tasks. The forgetting for the $j$th task is $f_j = \max_{l \in \{1,...,j−1\}} (a_{l,j} − a_{k,j})$, $\forall j < k$. The average forgetting after the training stage of task $k$ is written as $F_k = (1/k − 1) \sum_{j=1}^{k−1} f_j$.

A. Results and Analysis

To evaluate the effectiveness of our method, we conduct experiments on CUB, FLO, CAL, and CIFAR100.

Authorized licensed use limited to: XIDIAN UNIVERSITY. Downloaded on March 02,2024 at 02:08:28 UTC from IEEE Xplore. Restrictions apply.
Tables I and II represent the average incremental accuracy on the CUB, CIFAR100, FLO, and CAL datasets with six- task scenarios. The base datasets consist of 50, 50, 52, and 51 classes, respectively, for CUB, CIFAR100, FLO, and CAL. Compared with the results of E-FT, other incremental methods obtain better performance, which proves the existence of catastrophic forgetting in embedding networks. Both SDC and DRT can be equipped with regularization-based incremental methods to further improve the performance; moreover, DRT obtains a more impressive improvement on the CUB and CIFAR100 datasets. On the CUB dataset, DRT obtains 59.4%, 60.4%, 60.2%, and 59.0%, representing improvements of 7.1%, 5.8%, 2.7%, and 2.7% compared with SDC when equipped with E-FT, E-LwF, E-EWC, and E-MAS. On the CIFAR100 dataset, DRT obtains 20.3%, 47.0%, 47.1%, and 47.3%, representing improvements of 11.8%, 20.9%, 20.7%, and 20.5% compared with SDC when equipped with E-FT, E-LwF, E-EWC, and E-MAS.
Fig. 5. Comparison of the average incremental accuracy and forgetting with an 11-task setting on (a) and (b) CUB and (c) and (d) CIFAR100 datasets.

Fig. 6. Comparison of the average incremental accuracy and forgetting with an average ten-task setting on (a) and (b) CUB and (c) and (d) CIFAR100 datasets.
WEI et al.: INCREMENTAL EMBEDDING LEARNING WITH DRT

Fig. 7. t-sne results on CIFAR100. (a) E-FT. (b) E-FT+DRT. (c) E-LwF. (d) E-LwF+DRT. (e) E-EWC. (f) E-EWC+DRT. (g) E-MAS. (h) E-MAS+DRT.

24.3%, and 8.9% compared with SDC when equipped with E-FT, E-LwF, E-EWC, and E-MAS. On the FLO dataset, DRT obtains 83.8%, 82.3%, 84.6%, and 82.4%, representing improvements of 1.3%, 0.5%, 0.4%, and 1.8% compared with SDC when equipped with E-FT, E-LwF, E-EWC, and E-MAS. On the CAL dataset, DRT obtains 85.3%, 84.3%, 85.6%, and 84.3%, representing improvements of 0.2%, 0.2%, 0.6%, and 0.7% compared with SDC when equipped with E-FT, E-LwF, E-EWC, and E-MAS. These results prove that DRT is a better method for bridging the semantic gap between two tasks when compared with SDC and can easily be combined with existing regularization-based incremental methods (such as EWC, LwF, or MAS) to further alleviate catastrophic forgetting in embedding networks.

Table III represents the final accuracies for different tasks on CUB and CIFAR100 datasets. It is obvious that our method achieves better performance in the first several tasks compared with other methods, which proves the effectiveness of our method for alleviating catastrophic forgetting. The reason for the bad performances in the final several tasks is negative transfer. This phenomenon means our method preserves sufficient knowledge of previous tasks and influences the learning for current task, which is common for incremental learning. Compared with the little drop in the final several tasks, the improvement in the first several tasks is impressive, which leads to the better average recognition accuracy.

In addition, the average incremental accuracy and average forgetting of different incremental methods on CUB and CIFAR100, which is set up as an 11-task scenario, are represented in Fig. 5. The number of classes in the base dataset is set to 50 for CUB and CIFAR100. By examining the average incremental accuracy, we can determine that E-EWC+DRT and E-MAS+DRT obtain the best performance on CUB and CIFAR100, respectively, when compared with other methods. As for the average forgetting, our method suffers from less forgetting than other methods on both the CUB and CIFAR100 datasets, which demonstrates the capability of our method to prevent forgetting. In addition, when equipped with the same regularization-based incremental learning method, DRT performs better than SDC on both CUB and CIFAR100 under two evaluation metrics; this proves that DRT is a better method for approximating and compensating for the semantic gap between two adjacent classification spaces.

To further prove the effectiveness of our method, we set up an experiment with an average ten-task scenario on CUB and CIFAR100, the results of which are presented in Fig. 6. Under this setting, the base dataset exerts only a very small influence on the performance. As for the average incremental accuracy, we note that E-FT+DRT and E-LwF+DRT obtain the best performance on CUB and CIFAR100, respectively, when compared with other methods. As for the average forgetting, our method alleviates catastrophic forgetting more effectively, which further proves the superiority of our method.

Compared with other state-of-the-art incremental methods (EWC [8], MAS [8], synaptic intelligence (SI) [37],

<table>
<thead>
<tr>
<th>Method</th>
<th>Task Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>E-FT</td>
<td>11.2</td>
</tr>
<tr>
<td>EWC [22]</td>
<td>17.3</td>
</tr>
<tr>
<td>E-LwF [9]</td>
<td>16.8</td>
</tr>
<tr>
<td>SI [37]</td>
<td>23.5</td>
</tr>
<tr>
<td>Rwalk [38]</td>
<td>21.4</td>
</tr>
<tr>
<td>LwF [9]</td>
<td>23.8</td>
</tr>
<tr>
<td>DMC [39]</td>
<td>37.8</td>
</tr>
<tr>
<td>E-LwF+DRT</td>
<td>46.6</td>
</tr>
<tr>
<td>E-EWC+DRT</td>
<td>44.4</td>
</tr>
<tr>
<td>E-MAS+DRT</td>
<td>45.3</td>
</tr>
</tbody>
</table>
TABLE V

<table>
<thead>
<tr>
<th>Method</th>
<th>CUB</th>
<th>CIFAR100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base</td>
<td>E-FT 57.5</td>
<td>E-FT 7.8</td>
</tr>
<tr>
<td>+DR</td>
<td>E-LwF 58.4</td>
<td>E-LwF 42.7</td>
</tr>
<tr>
<td>+MGT</td>
<td>E-EWC 57.6</td>
<td>E-EWC 42.8</td>
</tr>
<tr>
<td>+DR+MGT</td>
<td>E-MAS 56.8</td>
<td>E-MAS 43.7</td>
</tr>
</tbody>
</table>

TABLE VI

<table>
<thead>
<tr>
<th>Regularization</th>
<th>CUB</th>
<th>CIFAR100</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{L}_{\text{all}}$</td>
<td>E-FT 58.5</td>
<td>E-FT 11.8</td>
</tr>
<tr>
<td>$\mathcal{L}_{\text{task-class}}$</td>
<td>E-LwF 58.4</td>
<td>E-LwF 45.7</td>
</tr>
<tr>
<td>$\mathcal{L}_{\text{task}}$</td>
<td>E-EWC 58.9</td>
<td>E-EWC 44.4</td>
</tr>
<tr>
<td>$\mathcal{L}_{\text{rec}}$</td>
<td>E-MAS 58.9</td>
<td>E-MAS 44.6</td>
</tr>
</tbody>
</table>

Fig. 8. Parameter sensitivity experiment of (a) and (f) $\gamma$, (b) and (g) $\beta$, (c) and (h) $\delta$, (d) and (i) $\rho$, and (e) and (j) $\eta$ on CUB and Cifar100 datasets.

TABLE VII

<table>
<thead>
<tr>
<th>Class-Mask</th>
<th>Task-Mask</th>
<th>CUB</th>
<th>CIFAR100</th>
</tr>
</thead>
<tbody>
<tr>
<td>E-FT+DRT</td>
<td>58.5</td>
<td>13.1</td>
<td>11.8</td>
</tr>
<tr>
<td>E-LwF+DRT</td>
<td>58.4</td>
<td>19.7</td>
<td>45.7</td>
</tr>
<tr>
<td>E-EWC+DRT</td>
<td>58.4</td>
<td>15.6</td>
<td>44.4</td>
</tr>
<tr>
<td>E-MAS+DRT</td>
<td>57.5</td>
<td>16.7</td>
<td>46.3</td>
</tr>
</tbody>
</table>

TABLE VIII

<table>
<thead>
<tr>
<th>CAL</th>
<th>FLO</th>
<th>AVG</th>
</tr>
</thead>
<tbody>
<tr>
<td>E-FT</td>
<td>45.9</td>
<td>90.6</td>
</tr>
<tr>
<td>E-FT+DRT</td>
<td>49.7</td>
<td>87.7</td>
</tr>
<tr>
<td>E-LwF+DRT</td>
<td>77.1</td>
<td>79.3</td>
</tr>
<tr>
<td>E-EWC+DRT</td>
<td>77.2</td>
<td>77.3</td>
</tr>
<tr>
<td>E-MAS+DRT</td>
<td>79.6</td>
<td>70.5</td>
</tr>
</tbody>
</table>

Table IX

Remanian Walk for Incremental Learning (Rwalk) [38], LwF [9], deep model consolidation (DMC) [39], which do not store the samples of previous tasks, our method obtains the best results with different numbers of tasks on CIFAR100, which is represented in Table IV. Our method obtains improvements of 8.8% and 13.2% based on these methods, when CIFAR100 is divided into 10 and 20 tasks; this proves that our methods can effectively alleviate catastrophic forgetting without storing the samples of previous tasks.

We also provide the qualitative results to prove the effectiveness of our method, shown in Fig. 7. There are the feature distributions of fifteen classes belonging to the base task in the final classification spaces. SDC does not change the feature distributions, but updates the class prototypes in the incremental processes. Thus, the feature distributions of SDC with E-FT/E-LwF/E-EWC/E-MAS are the same as the incremental methods. It is obvious that our method can obtain more discriminative representation of these classes without relearning these classes. This phenomenon indicates that our method can bridge the semantic gaps between tasks and alleviate catastrophic forgetting effectively.

B. Ablation Study

We conduct several groups of ablation experiments to study the effectiveness of our method.
TABLE IX  
AVERAGE ACCURACIES FOR DIFFERENT TASKS ON THE CUB DATASET BASED ON ALEXNET AND VGG16

<table>
<thead>
<tr>
<th>Method</th>
<th>AlexNet</th>
<th>VGG16</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>T1</td>
<td>T2</td>
</tr>
<tr>
<td>E-FT</td>
<td>56.1</td>
<td>47.6</td>
</tr>
<tr>
<td>E-FT+SDC</td>
<td>56.1</td>
<td>48.1</td>
</tr>
<tr>
<td>E-FT+DRT</td>
<td>56.1</td>
<td>51.7</td>
</tr>
<tr>
<td>E-LwF</td>
<td>56.1</td>
<td>49.9</td>
</tr>
<tr>
<td>E-LwF+SDC</td>
<td>56.1</td>
<td>49.4</td>
</tr>
<tr>
<td>E-LwF+DRT</td>
<td>56.1</td>
<td>50.2</td>
</tr>
<tr>
<td>E-EWC</td>
<td>56.1</td>
<td>51.1</td>
</tr>
<tr>
<td>E-EWC+SDC</td>
<td>56.1</td>
<td>51.4</td>
</tr>
<tr>
<td>E-EWC+DRT</td>
<td>56.1</td>
<td>51.8</td>
</tr>
<tr>
<td>E-MAS</td>
<td>56.1</td>
<td>48.9</td>
</tr>
<tr>
<td>E-MAS+SDC</td>
<td>56.1</td>
<td>49.1</td>
</tr>
<tr>
<td>E-MAS+DRT</td>
<td>56.1</td>
<td>49.5</td>
</tr>
</tbody>
</table>

The results of our basic model with different modules incrementally added are present in Table V. The basic model is the embedding network equipped with specific incremental learning methods, such as E-LwF, E-EWC, and E-MAS, while the translation networks are two-layer fully-connected networks with the loss of $L_{\text{align}}$. Based on the base model, we add a disentangled representation module and mask-guided translation module, denoted as “DR” and “MGT,” respectively. When adding “DR” and “MG” into the base model, the model performance further improves. These improvements prove that the disentangled representation module can capture the distribution of different classes, while the mask-guided translation module can preserve more discriminative and representative latent features in different classification spaces.

As shown in Table VI, we next conduct an ablation study to evaluate the effectiveness of different objective functions, when the translation network is mask-guided. We can clearly determine that all four losses have a positive influence on alleviating catastrophic forgetting in embedding networks on two popular datasets. When all three losses are added, our method achieves the best performance, which proves that these objective functions can cooperate with each other to improve the performance.

As shown in Fig. 8, we perform an experiment to discuss the influence of the hyperparameters on CUB and CIFAR100, $\gamma$, $\beta$, $\delta$, $\rho$, and $\eta$, which is set up as a six-task scenario. The best performance is achieved when $\gamma$, $\beta$, $\delta$, $\rho$, and $\eta$ are set to 1000, 100, 100, 100, and 0.1 for CUB, and 200, 100, 100, 100, and 0.1 for CIFAR100. These results clearly show that the average incremental accuracy increases with the increase of hyperparameters before the peak performance of average incremental accuracy is achieved.

To prove the effectiveness of class-mask, we design an ablation study, whose results are shown in Table VII. It is notable that our method obtains better performance with class-mask than the performance with task-mask, proving the class-mask information containing more discriminative class-level information compared with the task-mask information.

To further prove the effectiveness of our method, we design the dataset-incremental learning experiment based on CAL and FLO datasets, whose results are shown in Table VIII. The result of E-FT proves that catastrophic forgetting still exists in dataset-incremental learning problem. Our method achieves better performance compared with E-FT, which indicates our method still can alleviate catastrophic forgetting for dataset-incremental learning problem.

To prove the flexibility of our proposed method, we combine our method with other frameworks, such as AlexNet [43] and VGG16 [44], whose results are shown in Table IX. When compared with other methods, our method obtains the best performance based on two frameworks, which indicates the superiority of our method.

V. CONCLUSION AND FUTURE WORKS

In this article, we propose a novel method designed to alleviate catastrophic forgetting for embedding networks. To estimate and compensate for the semantic gap between two adjacent classification spaces, we construct a disentangled translation model that projects the latent features of the embedding space into class-disentangled space and task-disentangled space, in which the class-disentangled features from two domains are precisely aligned and measured. We then design a mask-guided translation network to preserve more discriminative and representative latent features in the translation process. In addition, our proposed method can be flexibly combined with other incremental learning methods to further improve the performance of alleviating catastrophic forgetting. Experiments show that our method outperforms other methods by a large margin on four benchmark datasets. To alleviate catastrophic forgetting further, we will capture the relationships between the prototypes and the examples. In addition, storing few examples to improve the performance also should be considered in the future works.

REFERENCES


Da Chen received the B.S. degree in optic science from Xidian University, Xi’an, China, in 2011, the M.S. degree in computer science from Halmstad University, Halmstad, Sweden, in 2013, and the Ph.D. degree in computer science from the University of Bath, Bath, U.K., in 2018. He is currently a Researcher with the Alibaba Group, Beijing, China. His research interests include machine learning, computer vision, and multimodal analysis.

Yuhong Li received the bachelor’s degree in software engineering from Xidian University, Xi’an, China, in 2010, and the Ph.D. degree from the Department of CIS, University of Macau, Macau, China, in 2016. He is currently a Senior Staff Engineer with the Security Department, Alibaba Group, Beijing China. He has authored more than 20 papers on top-tier conferences and journals, including the International Conference on Very Large Data Bases (VLDB), the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), the ACM International Conference on Information and Knowledge Management (CIKM), the IEEE Transactions on Visualization and Computer Graphics (TVCG), and the International Conference on Extending Database Technology (EDBT). His current research interests include AI for social good, multimodal understanding, self-supervised learning on big data, and edge computing.

Dr. Li is a recipient of the Stars of Tomorrow (Award of Excellent Intern), Microsoft Research Asia, Technology Talent of Xidian University, and AliStar.

Xu Yang (Member, IEEE) received the B.E. and Ph.D. degrees in electronic and information engineering from Xidian University, Xi’an, China, in 2016 and 2021, respectively. He is currently a Lecturer with the School of Electronic Engineering, Xidian University. His research interests include computer vision and machine learning.

Cheng Deng (Senior Member, IEEE) received the B.E., M.S., and Ph.D. degrees in signal and information processing from Xidian University, Xi’an, China, in 2001, 2006, and 2009, respectively. He is currently a Full Professor with the School of Electronic Engineering, Xidian University. He has authored or coauthored more than 100 scientific articles at top venues, including IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON CYBERNETICS, the Neural Information Processing Systems (NeurIPS), the International Conference on Machine Learning (ICML), the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), the IEEE International Conference on Computer Vision (ICCV), the Association for the Advancement of Artificial Intelligence (AAAI), the International Joint Conferences on Artificial Intelligence (IJCAI), and the Knowledge Discovery and Data Mining (KDD). His research interests include computer vision, pattern recognition, and information hiding.

Dacheng Tao (Fellow, IEEE) is currently the President of the JD Explore Academy, Beijing, China, and a Senior Vice President of JD.com. He is also an Advisor and the Chief Scientist of the Digital Science Institute, The University of Sydney, Darlington, NSW, Australia. He mainly applies statistics and mathematics to artificial intelligence and data science, and his research is detailed in one monograph and over 200 publications in prestigious journals and proceedings at leading conferences.

Dr. Tao is a fellow of the Australian Academy of Science, the American Association for the Advancement of Science (AAAS), and ACM. He received the 2015 Australian Scopus-Eureka Prize, the 2018 IEEE ICDM Research Contributions Award, and the 2021 IEEE Computer Society McCluskey Technical Achievement Award.