DeepCollaboration: Collaborative Generative and Discriminative Models for Class Incremental Learning

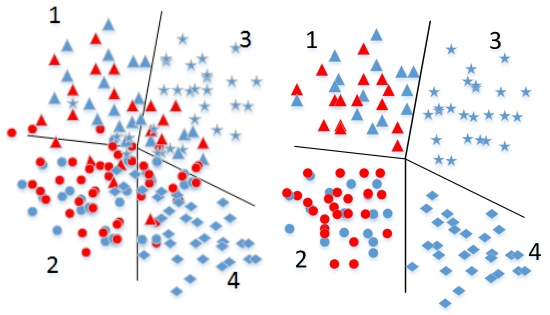
Bo Cui[1,3], **Guyue Hu**[1,3], **Shan Yu**[1,2,4]
[1]Brainnetome Center and National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences
[2]CAS Center for Excellence in Brain Science and Intelligence Technology, Chinese Academy of Sciences
[3]School of Artificial Intelligence, University of Chinese Academy of Sciences
[4]School of Future Technology, University of Chinese Academy of Sciences
{bo.cui, guyue.hu, shan.yu}@nlpr.ia.ac.cn

Abstract

An important challenge for neural networks is to learn incrementally, i.e., learn new classes without catastrophic forgetting. To overcome this problem, generative replay technique has been suggested, which can generate samples belonging to learned classes while learning new ones. However, such generative models usually suffer from increased distribution mismatch between the generated and original samples along the learning process. In this work, we propose DeepCollaboration (D-Collab), a collaborative framework of deep generative and discriminative models to solve this problem effectively. We develop a discriminative learning model to incrementally update the latent feature space for continual classification. At the same time, a generative model is introduced to achieve conditional generation using the latent feature distribution produced by the discriminative model. Importantly, the generative and discriminative models are connected through bidirectional training to enforce cycle-consistency of mappings between feature and image domains. Furthermore, a domain alignment module is used to eliminate the divergence between the feature distributions of generated images and real ones. This module together with the discriminative model can perform effective sample mining to facilitate incremental learning. Extensive experiments on several visual recognition datasets show that our system can achieve state-of-the-art performance.

Introduction

Deep neural networks trained end-to-end have achieved encouraging results on supervised learning tasks (Krizhevsky, Sutskever, and Hinton 2012; He et al. 2016). Typically all the samples and labels must be available for training these systems. However, intelligent systems should be able to learn continually, in other words, sequentially learn new tasks without access to the past data, while preserving the knowledge learned from old tasks. For incremental representation learning, the new task means classifying samples from new classes together with old classes. Catastrophic forgetting (McCloskey and Cohen 1989) is the core problem with class incremental learning. The networks are severely biased by the data of new classes as the past data are not available during training. So the classification accuracy for old classes may quickly deteriorate. To alleviate this problem, most of previous approaches need to store exemplars for old classes and utilize knowledge distillation technique. Storing past samples results in increasing memory cost as new classes come. To handle this, generative replay technique has been proposed (Shin et al. 2017). Using standard deep generative model, generative replay generates pseudo-samples for old classes and mixes them with new samples for further incremental learning. But this method suffers from the distribution shift between the generated data and original samples. Typically the generated samples are visually quite different from natural images. What's more, large distribution divergence exists between the generated samples and real images in feature space, as illustrated in Figure 1. If using them di-



(a) GAN (b) D-Collab

Figure 1: (a) Problem of generative replay (Shin et al. 2017) based incremental learning: when new classes 3 and 4 come, raw training data for class 1 and 2 are not available. Using samples generated from a GAN (Goodfellow et al. 2014) with distribution shift from raw data for training, will result in mis-classfication. (b) Our D-Collab model can dramatically alleviate the domain shift between the generated data and raw samples, meanwhile drop the performance-harming generated samples. Red: features of generated samples; Blue: features of raw samples; Black line: decision boundaries for the incremental learning task; Triangle, Circle, Star and Diamond indicate four different classes, respectively.

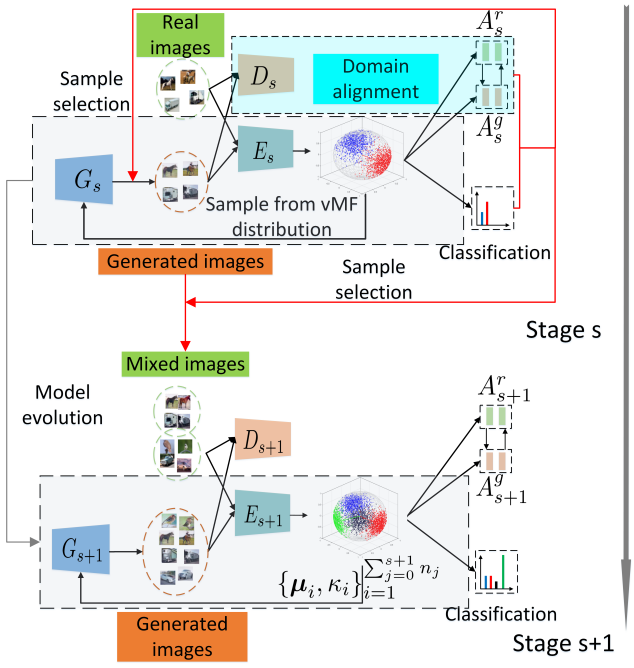


Figure 2: System overview. We propose to use collaborative generative and discriminative models $M = \{E, G, D, A^r, A^g\}$ for incremental learning. The whole system consists of three parts, the discriminative model E, the generative model G, and the domain alignment module $\{D, A^r, A^g\}$. E and G are connected with incrementally updated vMF distributions. They evolve as a whole along the learning stages while the domain classifier D and feature adaptors A^r, A^g are re-initialized at every stage. For stage $s+1$ a model M_{s+1} is trained using real images of class $s+1$ and generated samples from model M_s, to classify $n_0 + ... + n_{s+1}$ classes. The domain alignment module $\{D, A^r, A^g\}$ from stage s can also perform sample selection collaborated with E.

rectly for incremental learning, the system cannot classify accurately on all classes.

In this paper, we propose DeepCollaboration, a collaborative framework of deep generative and discriminative models to solve the problem. A discriminative learning model is designed to incrementally update the feature embedding space. Meanwhile a generative model is proposed to achieve conditional generation using the latent feature distribution from the discriminative model. Then through bidirectional training the two models are connected and collaborate with each other. The resulted cycle-consistency between feature domain and image domain will alleviate the domain shift problem. As can be seen in Figure 1, we illustrate the benefits of our designs.

In summary, the contributions of this work include: (1) To address the problem of distribution shift in generative replay systems for incremental representation learning, we propose a novel method to connect deep generative and discriminative models and train them end-to-end in a collaborative way. A domain alignment module is further introduced to alleviate the divergence between generated samples and real ones. (2) We propose an effective and efficient sample mining method for generated samples through the collaboration of discriminative model and domain alignment module. (3) We experimentally demonstrate that the DeepCollaboration model can achieve state-of-the-art performance on standard class incremental learning tasks, with a large margin above naive generative replay method.

Related Work

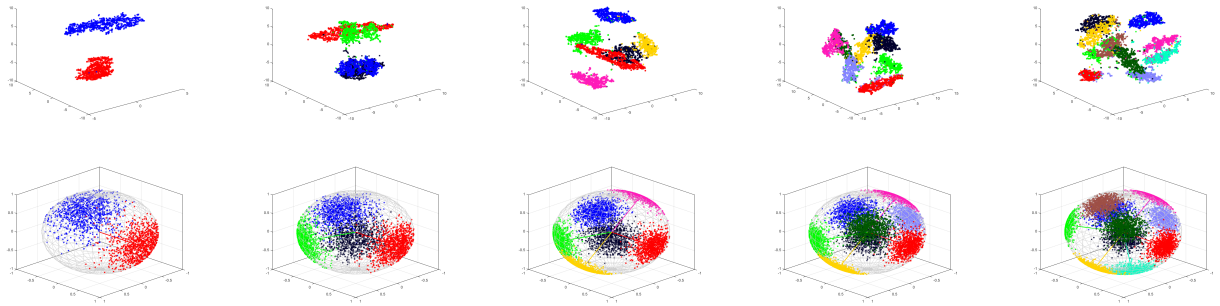
Overcoming Catastrophic Forgetting

In recent years overcoming catastrophic forgetting in DNNs has drawn much attention from researchers. Different approaches aim to solve the problem in different task settings. For general settings, regularization-based methods are the main-stream. EWC (Kirkpatrick et al. 2017) employed Fisher information guided regularization technique to protect the most important weights for past tasks from drastic changes, and make the less important weights plastic to adapt for new tasks. Different from EWC, SI (Zenke, Poole, and Ganguli 2017) computed the per-synapse consolidation strength over learning trajectory in an online fashion, instead of computing synaptic importance offline. A novel orthogonal weight modification algorithm, OWM, was proposed to enable the weights of a network only be modified in the directions orthogonal to the subspace spanned by all previously learned tasks (Zeng et al. 2019). These general methods are not designed specifically for incremental representation learning, and regularization-based methods like EWC typically cannot generalize well on convolutional networks.

Class Incremental Learning

Among methods for class incremental learning, LwF (Li and Hoiem 2016) was the first to use knowledge distillation method (Hinton, Vinyals, and Dean 2015). Using only images of newest classes to train the network, the knowledge distillation loss encourages the outputs of new network to approximate the outputs of older one. However the performance of LwF deteriorated after incremental learning of several stages. To enhance the performance, iCaRL (Rebuffi et al. 2017) further stored selected exemplars of old classes. While using combination of classification loss and knowledge distillation loss, iCaRL performed nearest mean classification during testing. As a result, iCaRL achieved very good incremental learning accuracies. EEiL (Castro et al. 2018) improved iCaRL by learning the classifier and the features jointly, in an end-to-end fashion.

Knowledge distillation methods for class incremental learning need to store past samples to achieve good performance. To be more memory efficient, generative replay methods have been proposed. Shin et al. proposed to use class-conditional GAN to replay the samples for old classes (Shin et al. 2017), however the method was only tested on digits datasets. FearNet (Kemker and Kanan 2018) utilized a brain-inspired dual-memory system to replay the past features, while the feature extractor was pre-trained on large



(a) Stage 0 (b) Stage 1 (c) Stage 2 (d) Stage 3 (e) Stage 4

Figure 3: Latent feature distributions on MNIST dataset of cGAN (top) and our D-Collab model (bottom). At every stage 2 new classes are added. The dimensions of features are reduced to 3 using t-SNE (Van Der Maaten 2014).

datasets and fixed. Xiang et al. extended the framework of FearNet to convolutional networks (Xiang et al. 2019), but the parameters of most convolution layers were still fixed. Some recent replay-based methods (Iscen et al. 2020; Hayes et al. 2020; Liu et al. 2020) were also designed to replay features. Our D-Collab model can replay images and achieve similar classification accuracies with iCaRL on end-to-end class incremental (representation) learning, without storing past samples.

The DeepCollaboration Framework

Problem Definition and System Overview

We define the class incremental learning task as follows. There are totally s sequential stages in the learning process. At stage 0 a classification model M_0 is trained on dataset X_0 with n_0 classes. For stage j a model M_j is trained to classify on accumulated $\sum_{m=0}^{j} n_m$ classes also with only the newest dataset X_j. Let us denote the new dataset as $X_j = \{(\mathbf{x}_i; y_i), 1 \leq i \leq N_j, y_i \in [1+\sum_{m=0}^{j-1} n_m, ..., \sum_{m=0}^{j} n_m]\}$ where N_j is the size of the dataset, $\mathbf{x}_i$ and y_i are the image and the label, respectively.

We propose to use collaborative deep generative and discriminative models $M = \{E, G, D, A^r, A^g\}$ to solve this problem. The whole system consists of three parts, the discriminative model E, the generative model G, and the domain alignment module $\{D, A^r, A^g\}$. The discriminative model E is in fact an encoder which can perform representation learning and create good feature embeddings for classification. The generative model aims to generate pseudo-images $\hat{X}$ at stage $j+1$ to serve as the training data for old classes. To solve the distribution mismatch problem, the discriminative model E_j and the generative model G_j are connected through a unified feature distribution design and bidirectional joint training. In addition, the domain alignment module is introduced. The discriminator D_j and adaptors $\{A_j^r, A_j^g\}$ try to minimize the discrepancy between generated samples and real ones in image domain and feature domain, respectively. After appropriate training of the collaborative models, the domain alignment module can also serve as a sample selection module for stage $j+1$, providing high-quality samples to M_{j+1} with G_j. The overall framework is shown in Figure 2.

So the collaboration of the generative and discriminative models are three-fold. Firstly, E and G are collaborated to achieve joint discriminative learning and conditional generation, and can evolve as a whole to achieve incremental learning. Secondly, all models in M are collaborated to reduce the domain distribution shift between generated images and raw images, in both the image space and feature space. Thirdly, models in stage j are collaborated to perform sample selection for current stage, and stage $j+1$.

The Discriminative Model

Using a discriminative model E we can get the feature embedding of an image: $\mathbf{f} = E(\mathbf{x})$. Many representation learning approaches normalize their features to be on the unit hypersphere (Wang and Isola 2020). L_2 normalization eliminates the influence of magnitudes of features, making directions of features the major component to optimize. Here we use a mixture of von Mises-Fisher (vMF) distributions (Banerjee et al. 2005) to model the feature distribution of images explicitly. For a d dimensional feature vector $\mathbf{f} = [f_1, ..., f_d]$ ($\|\mathbf{f}\|_2 = 1$), the density function of the vMF distribution is defined as: $q(\mathbf{f}|\boldsymbol{\mu}, \kappa) = C_d(\kappa) exp(\kappa \boldsymbol{\mu}^T \mathbf{f})$, and $\|\boldsymbol{\mu}\|_2 = 1$. Here $\boldsymbol{\mu}$ is the mean vector and κ is the concentration parameter (with $\kappa \geq 0$). $C_d(\kappa)$ is the normalization constant. $C_d(\kappa) = \kappa^{d/2-1}/(2\pi)^{d/2} I_{d/2-1}(\kappa)$, where $I_v(*)$ is the modified Bessel function of the first kind with order v. The shape of the vMF distribution depends on the value of the concentration parameter κ. The larger value of κ is, the more strongly the distribution is concentrated to the mean direction. In contrary, for low values of κ, the l_2 normalized features are more uniformly distributed on the hypersphere. The two parameters can be effectively estimated through maximum likelihood estimates (Banerjee et al. 2005) .

Given a training set with n classes, in this probability space, a sample $\mathbf{x}$ with feature $\mathbf{f}$ is assigned to class c with

the following normalized probability:

$$P(c|\mathbf{f}, \{\kappa_i, \boldsymbol{\mu}_i\}_{i=1}^n) = \frac{C_d(\kappa_c) exp(\kappa_c \boldsymbol{\mu}_c^T \mathbf{f})}{\sum_{i=1}^n C_d(\kappa_i) exp(\kappa_i \boldsymbol{\mu}_i^T \mathbf{f})}. \quad (1)$$

Then for all samples in $X = \{(\mathbf{x}_j; y_j), 1 \leq j \leq N\}$, the discriminative loss is defined as:

$$\mathcal{L}_{dis} = -\frac{1}{N} \sum_{j=1}^{N} log \frac{C_d(\kappa_{y_j}) exp(\kappa_{y_j} \boldsymbol{\mu}_{y_j}^T \mathbf{f}_j)}{\sum_{i=1}^n C_d(\kappa_i) exp(\kappa_i \boldsymbol{\mu}_i^T \mathbf{f}_j)}. \quad (2)$$

Note that for new classes in stage $s+1$, new distributions are learned from scratch, while old distributions are adapted using the saved statistics from stage s. We show this in Figure 3. Thus the discriminative model is expanded by storing new class mean vectors and estimated concentration parameters. During testing each sample will be classified to a class which has the highest probability computed using equation (1).

The Generative Model

For replay-based incremental learning, a generative model is needed to do the inverse mapping of a discriminative model: $\hat{\mathbf{x}} = G(\mathbf{f})$. GAN (Goodfellow et al. 2014) and VAE (Kingma and Welling 2014) are the mostly used deep generative models. There are also works combined the advantage of these two models, e.g, (Larsen et al. 2016). As we use von Mises-Fisher (vMF) distributions to model the distributions of the latent space explicitly, we can connect our discriminative model with these generative models efficiently.

Connect Discriminative Model with Conditional GAN The conditional GAN (cGAN) (Mirza and Osindero 2014) consists of a class-conditional generator $G(\mathbf{z}, c)$ associated with a class-conditional discriminator $D(\mathbf{x}, c)$. Typically c is the (discrete) class label and sampled from the categorical distribution P_π, and $\mathbf{z}$ is random noise sampled from normal distribution. The generator and discriminator are trained to optimize the following adversarial objective:

$$\begin{aligned} \mathcal{L}_{GAN}(G, D) = \mathop{\mathbb{E}}_{c \sim P_\pi} \Bigg[\mathop{\mathbb{E}}_{\mathbf{x} \sim \pi_c} [log D(\mathbf{x}, c)] + \\ \mathop{\mathbb{E}}_{\mathbf{z} \sim \mathcal{N}(0,I)} log(1 - D(G(\mathbf{z}, c), c))] \Bigg]. \end{aligned} \quad (3)$$

Here we propose to use the class centers $\{\boldsymbol{\mu}_i\}_{i=1}^n$ of the learned mixture distributions from the discriminative model as the conditional signals. These mean vectors are informative and explicitly connected to the feature distributions. Also the random noise $\mathbf{z}$ will be sampled from the vMF distribution with mean vector $\boldsymbol{\mu}$. Note that we already have the encoder E that can classify the real and generated samples to corresponding classes, so the generator G and domain classifier D can be trained using the following loss:

$$\begin{aligned} \mathcal{L}_{GAN}(G, D) = \mathop{\mathbb{E}}_{c \sim P_\pi} \Bigg[\mathop{\mathbb{E}}_{\mathbf{x} \sim \pi_c} [log D(\mathbf{x})] + \\ \mathop{\mathbb{E}}_{\mathbf{z} \sim vMF(0, \kappa_c)} log(1 - D(G(\mathbf{z} + \boldsymbol{\mu}_c)))] \Bigg]. \end{aligned} \quad (4)$$

Connect Discriminative Model with VAE Different from GAN, the VAE model first uses encoder E to map images to the latent space: $\mathbf{f} = E(\mathbf{x})$, then recovers the raw data through a generator $\hat{\mathbf{x}} = G(\mathbf{f})$. The training objective for VAE consists of two parts, a reconstruction loss and a KL loss. The reconstruction loss can be l_1 loss or l_2 loss between the recovered images and raw data. While KL loss represents the divergence between the generated and expected distributions of feature space. According to (Hashimoto et al. 2018), the KL divergence between two vMF distributions, $vMF_1(\boldsymbol{\mu}_q, \kappa)$ and $vMF_2(\boldsymbol{\mu}_c, \kappa)$, is:

$$\begin{aligned} KL(vMF_1 || vMF_2) &= \mathbb{E}[\kappa \mathbf{f}^T \boldsymbol{\mu}_q - \kappa \mathbf{f}^T \boldsymbol{\mu}_c] \\ &= \kappa \mathbb{E}[\mathbf{f}]^T \boldsymbol{\mu}_q - \kappa \mathbb{E}[\mathbf{f}]^T \boldsymbol{\mu}_c \\ &= C_\kappa (1 - \boldsymbol{\mu}_q^T \boldsymbol{\mu}_c), \end{aligned} \quad (5)$$

where the two vMFs denote the posterior $q_E(\mathbf{f}|\mathbf{x})$ and the prior $p_G(\mathbf{f})$ accordingly, and C_κ denotes the constant integrated in this equation. This means for our vMF distribution, the solution for KL loss is equivalent to minimizing the Euclidean distance between the features and the mean vector $\boldsymbol{\mu_c}$ of the corresponding class c. Then for all samples in $X = \{(\mathbf{x}_j; y_j), 1 \leq j \leq N\}$, the KL loss is defined as:

$$\mathcal{L}_{KL} = \frac{C_\kappa}{2N} \sum_{j=1}^{N} \|\mathbf{f}_j - \boldsymbol{\mu}_{y_j}\|_2^2. \quad (6)$$

To encourage the output of the generator to match the input, we use an l_1 loss between the output and the input image as the reconstruction loss, defined as:

$$\mathcal{L}_1^{image} = \frac{1}{N} \sum_{j=1}^{N} \|G(E(\mathbf{x}_j)) - \mathbf{x}_j\|_1. \quad (7)$$

Cyclic Consistency in Training

After the initial training for encoder E we get the feature distributions for raw images. With these vMF distributions we can sample and train the GAN model, in other words G and D will be trained in an adversarial way. To eliminate the domain mismatch between the generated images and the real images, inspired by the recent work on image-to-image translation (Zhu et al. 2017a), we propose to use bi-directional training to enforce the cycle consistency.

The first cycle consistency enables a model to reconstruct raw images from the latent vectors with high quality, i.e., we try to get $G(E(\mathbf{x})) \rightarrow \mathbf{x}$. The system can be trained in a joint manner with D connected to G to improve the quality of generated images. The objective is

$$\begin{aligned} E^* G^* D^* = arg \min_{G,E} \max_D \mathcal{L}_{GAN}(G, D, E) \\ + \lambda \mathcal{L}_1^{image}(G, E) + \lambda_{KL} \mathcal{L}_{KL}(E). \end{aligned} \quad (8)$$

The other cycle consistency encourages the latent representation of generated images to be similar with the feature vectors sampled from: $E(G(\mathbf{f})) \rightarrow \mathbf{f}$. We try to optimize the following objective:

$$G^* D^* E^* = arg \min_{G,E} \max_D \mathcal{L}_{GAN}(G, D) + \lambda_{dis} \mathcal{L}_{dis}(G, E). \quad (9)$$

Algorithm 1 Training of D-Collab model

1: **Input:** Sequence of image set $\{X_0, X_1, ...X_{s-1}\}$.
2: **Output:** Incrementally learned models $M = \{E, G, D, A^r, A^g\}$.
3: Train E using X_0 based on equation (2), calculate statistics $\{\boldsymbol{\mu}_i, \kappa_i\}_{i=1}^{n_0}$ of normalized embeddings for initial classes, and get test results at the same time.
4: Initialize D for current stage. Sample in learned distributions, train G and D using loss in equation (4).
5: Train E, G and D jointly in a cycle consistent way using objectives in equation (8) and (9), two losses for generated samples are weighted using equation (14) and (15).
6: Initialize A^r and A^g. Further train E, G, A^r, A^g jointly using equation (13), where domain alignment losses in equation (10, 11, 12) are added to the training process.
7: **for** $j = 1$ **to** $s - 1$ **do**
8: Generate and select samples using M_{j-1}. Form batches after sample selection using acceptance rate in equation (16), together with new data X_j. Train E to get the new feature distributions $\{\boldsymbol{\mu}_i, \kappa_i\}_{i=1}^{\sum_{m=0}^{j} n_m}$, and perform classification on test set.
9: Repeat step 4-6 using mixed images and updated distributions.
10: **end for**

The Domain Adaptors

As discussed above, bi-directional training of E and G is used to alleviate the domain shift between generated images and real images. However distribution mismatch may still exist in feature space. Domain-invariant features should be able to be translated from one domain to the other (Li et al. 2019). To this end, we propose to train feature adaptors A^r and A^g separately which are applied to features of real and generated images, $\mathbf{f}_r$ and $\mathbf{f}_g$, to translate the features from one domain to the other. We use domain adversarial loss to train them, with D^g and D^r respectively:

$$\min_{A^r}\max_{D^g}\mathcal{L}_g = \mathbb{E}[logD^g(\mathbf{f}_g)] + \mathbb{E}[log(1 - logD^g(A^r(\mathbf{f}_r)))], \tag{10}$$

$$\min_{A^g}\max_{D^r}\mathcal{L}_r = \mathbb{E}[logD^r(\mathbf{f}_r)] + \mathbb{E}[log(1 - logD^r(A^g(\mathbf{f}_g)))]. \tag{11}$$

The desired property of A^r and A^g is cycle-consistency between the features of generated and real samples. The objective we try to optimize is:

$$\mathcal{L}_c = \mathbb{E}\big[||A^g(A^r(\mathbf{f}_r)) - \mathbf{f}_r||_2^2\big] + \mathbb{E}[||A^r(A^g(\mathbf{f}_g)) - \mathbf{f}_g||_2^2\big]. \tag{12}$$

The overall loss for this training phase is:

$$\mathcal{L}_{ada} = \mathcal{L}_g + \mathcal{L}_r + \mathcal{L}_c + \mathcal{L}_{dis}. \tag{13}$$

Generated Sample Selection

The sample selection is performed at two different phases. The first phase is the joint training phase of D-Collab model (encoder and generator) at stage j. With confidence scores computed from the collaborative models, the system can gradually learn from more confident samples to less confident samples. This helps to stable training process and achieve good performance (Zhang et al. 2018). The second phase is the pseudo-image generation phase at stage $j + 1$. Selecting samples generated by the D-Collab model trained from stage j should take both diversity and confidence into account.

	20	40	60	80	100
Finetuning	81.23	42.29	28.26	22.45	17.23
Frozen	81.45	47.31	33.52	26.89	22.76
LwF	81.74	62.23	50.01	41.50	34.98
iCaRL	81.43	72.19	65.21	59.43	54.38
EEIL	81.51	**74.49**	66.17	59.34	54.22
BiC	81.26	74.35	**67.52**	**62.27**	**57.12**
AE	81.93	68.32	54.17	43.03	36.79
AE-cGAN	81.67	71.46	55.48	45.58	40.65
cGAN	81.82	70.06	58.24	49.03	44.79
Ours	81.55	**74.45**	**67.82**	**62.01**	**56.92**

Table 1: Incremental learning results (accuracy %) on CIFAR-100 dataset with 20 new classes at every stage.

Sampling and Weighting for the Current Stage

We use a core-set based method (Sener and Savarese 2018) to generate initial samples which can cover the distribution of the real data, i.e., sample with high diversity. Let E_c denote the encoder with a classifier, the classification score for generated sample $\hat{\mathbf{x}}$ with assigned label $\hat{y}$ is $E_c(\hat{y}|\hat{\mathbf{x}})$, while the domain score is set to $1 - D(\hat{\mathbf{x}})$. Then at each epoch we select samples with high classification and domain scores. Along with the training process we dynamically adjust the proportion of selected samples and set proper weights for them. For domain adversarial loss, at training epoch n_{epoch} from total n_{total} epoches, acceptance rate is set to $r_d = max(0.2, min(n_{epoch}/n_{total}, 0.8))$, which means the generated samples with highest scores according to this proportion parameter are used for training. If a sample is accepted, then its selection indication $s_d(\hat{\mathbf{x}}) = 1$, otherwise $s_d(\hat{\mathbf{x}}) = 0$. The weight for an accepted sample $\hat{\mathbf{x}}$ is set to $w_d(\hat{\mathbf{x}}) = 2(1 - D(\hat{\mathbf{x}}))$. The weighted domain adversarial loss for N_g generated samples is:

$$\mathcal{L}_{GAN}^w = \frac{1}{N_g}\sum_{j=1}^{N_g} s_d(\hat{\mathbf{x}}_j) w_d(\hat{\mathbf{x}}_j)\mathcal{L}_{GAN}. \tag{14}$$

Similarly, for discriminative loss, acceptance rate is set to $r_e = max(0.2, min(n_{epoch}/n_{total}, 0.6))$. The weight for an accepted sample $\hat{\mathbf{x}}$ with feature $\hat{\mathbf{f}} = E(\hat{\mathbf{x}})$ is set to $w_e(\hat{\mathbf{x}}) = E_c(\hat{y}|\hat{\mathbf{x}})$. The weighted discriminative loss is:

$$\mathcal{L}_{dis}^w = -\frac{1}{N_g}\sum_{j=1}^{N_g} s_e(\hat{\mathbf{x}}_j) w_e(\hat{\mathbf{x}}_j) logP(\hat{y}_j|\hat{\mathbf{f}}_j, \{\kappa_i, \boldsymbol{\mu}_i\}_{i=1}^n). \tag{15}$$

Guided Sampling for the Next Stage Sample selection is also used to build a reliable pseudo-image pool for the

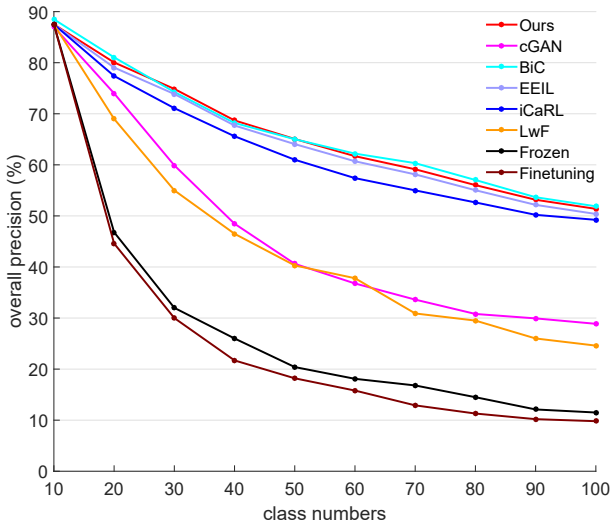


Figure 4: Incremental learning results on CIFAR-100 with 10 classes added at every stage.

next stage, which should take sample diversity and confidence into account. We modify the GOLD estimator (Mo et al. 2019), which is simple yet effective approach measuring the generation quality and class accuracy of generated samples. For a given generated sample $\hat{\mathbf{x}} = G(\mathbf{f})$ with the corresponding class $\hat{y}$, let $\hat{\mathbf{f}}$ denote $E(\hat{\mathbf{x}})$, the acceptance rate for such a sample is designed as follows:

$$r(\hat{\mathbf{x}}) = \frac{1}{H} exp\big(log\frac{D(\hat{\mathbf{x}})}{1-D(\hat{\mathbf{x}})} + log\frac{D^r(A^g(\hat{\mathbf{f}}))}{1-D^r(A^g(\hat{\mathbf{f}}))} + logE_c(\hat{y}|\hat{\mathbf{x}})\big), \quad (16)$$

where H is the normalization parameter. Using this acceptance function we can select accurate samples without losing diversity. The whole training procedure of D-collab is summarized in Algorithm 1.

Experiments

In this section, we first introduce the datasets and describe implementation details. We then show comparisons with the state-of-the-art methods, followed by the ablation study.

Datasets

Experiments are conducted on three datasets: CIFAR-100, MNIST, and CUB-200. CIFAR-100 (Krizhevsky 2009) contains 100 object classes. MNIST (LeCun et al. 1998) contains 10 digits classes. CUB-200 is a fine-grained image classification dataset containing high resolution images of 200 different bird species (Welinder et al. 2010). Following the class incremental benchmark protocol in iCaRL on CIFAR-100, 100 classes are arranged in a random order and come in as s parts. A multi-class classifier is adapted to recognize all seen classes along the incremental learning process. We have performed experiments for $s = 5$ and $s = 10$. For MNIST and CUB-200 we set $s = 5$.

	2	4	6	8	10
EWC	98.79	73.67	66.92	44.43	31.01
iCaRL	98.91	96.55	93.56	90.03	86.89
cGAN	98.81	98.56	98.23	98.16	98.21
Ours	98.82	**98.63**	**98.51**	**98.46**	**98.34**

Table 2: Incremental learning results (accuracy %) on MNIST dataset with 2 new classes at every stage.

	40	80	120	160	200
LwF	83.53	64.43	54.35	44.67	39.42
iCaRL	83.22	76.43	72.54	69.14	64.56
cGAN	83.42	74.06	66.16	58.31	52.60
Ours	83.46	**78.52**	**74.31**	**71.82**	**66.28**

Table 3: Incremental learning results (accuracy %) on CUB-200 dataset with 40 new classes at every stage.

Implementation Details

Our incremental learning strategy is applicable to commonly used deep networks, e.g., GoogLeNet (Szegedy et al. 2015) and ResNet (He et al. 2016). A ResNet with 32 layers is used as the encoder for CIFAR-100 and CUB-200, as it's the base-network of iCaRL. We follow the designs in (Gong et al. 2019) to implement the generator and the discriminator. The encoder for MNIST is also implemented following this work. The adaptors in the system are implemented using fully connected networks with 3 layers. Only during the initial training phase we use mini-batch stochastic gradient descent (SGD) to train the encoder for 90 epoches, with weight decay set to 0.0001. We set the initial learning rate to 0.05 and momentum to 0.9 for the encoder. After each 20 epochs, the initial learning rate is divided by 10. After that all the models are optimized using Adam (Kingma and Ba 2015) with learning rate of 0.0002 and $\beta = (0.0, 0.999)$. The samples within a minibatch are randomly and uniformly picked from the set of images of new classes and generated images of old classes. The hyperparameter κ used by the discriminative model is set to 10 at the initial stage and increased by 10 at every new stage. Other hyperparameters include the weights for different losses. We set λ_{image} = 10, λ_{dis} = 0.5, λ_{KL} = 0.1 in all experiments.

Results

We compare our approach with naive baselines, distillation-based approaches and generative replay methods. One of the naive baselines we use is Finetuning, which directly trains all the layers of a neural network using new data only. The other one, denoted by Frozen, freezes the feature extractor and only finetune the classifier. Distillation-based methods to which we compare include LwF (Li and Hoiem 2016), iCaRL (Rebuffi et al. 2017), EEiL (Castro et al. 2018) and BiC (Wu et al. 2019). LwF utilizes distillation loss with only real images of the current stage. Differently, iCaRL keeps

	20	40	60	80	100
cGAN	81.82	70.06	58.24	49.03	44.79
cGAN(w/ GS)	81.65	72.24	61.52	51.10	46.29
Ours (base)	81.51	72.46	64.23	57.72	50.38
Ours (w/o GS)	81.43	73.21	65.26	59.14	52.34
Ours (w/o DA)	81.57	74.06	66.94	60.76	54.84
Ours (w/o SW)	81.49	74.18	67.12	61.07	55.16
Ours (full)	81.55	**74.45**	**67.82**	**62.01**	**56.92**

Table 4: Ablation study results (accuracy %) on CIFAR-100 dataset with 20 new classes at every stage.

a sample set to store a small portion of old data. EEiL is similar with iCaRL, however it achieves unified feature and classifier learning. Compared with iCaRL, BiC tries to alleviate the data imbalance between a few stored old samples and a large amount of new ones. We implement a conditional GAN (Shin et al. 2017) as a baseline generative replay method, denoted by cGAN. We also implement an autoencoder (AE) following FearNet (Kemker and Kanan 2018), and AE+cGAN (Xiang et al. 2019) which combines AE with cGAN. As incremental representation learning models have to be trained in an end-to-end manner, we use AE and AE+cGAN to replay images rather than features here. All experiment results are averaged by 5 repeats.

Comparison with the State-of-the-Art Methods We divide the 100 classes of CIFAR-100 to 5 or 10 groups and the models need to learn incrementally. Table 1 shows the results of 5-stages learning setting. Performance of the naive baselines, Finetuning and Frozen, drop quickly. With knowledge distillation, LwF improves the accuracy of networks a lot. Combining knowledge distillation with storing samples, iCaRL is one of the state-of-the-art methods. Compared to iCaRL, the accuracy of our method is 2%-3% above this strong baseline. The performance of our approach is on par with the state-of-the-art methods which are developed by improving iCaRL, such as BiC and EEiL. What's more, our method outperforms cGAN by a large margin throughout the continual learning process. While AE and AE+cGAN perform worse than cGAN in this end-to-end incremental learning setting. We also test our model for 10-stages learning. As shown in Figure 4, the classification accuracy on all current learned classes are calculated and plotted. It can be seen from the plots that our method outperforms iCaRL by a consistent margin along the incremental classification accuracy curve. The overall performance on the total 100 classes is improved by more than 4% compared with iCaRL. Compared with cGAN, the best of the previous generative replay method, our method has about 80% improvement on the final classification accuracy.

Table 2 shows the results of 5-stages learning on MNIST. The regularization-based method EWC (Kirkpatrick et al. 2017) is also serving as a baseline. Our method outperforms iCaRL by a large margin. As images in MNIST dataset have compact background, even simple cGAN method can achieve good performance. However, our method results in clearly defined clusters for each class and appropriately aligned distributions in feature space, which is good for both classification and conditional generation. We show this in Figure 3 by plotting the latent space learned by our method and the cGAN.

The experiment results on CUB-200, as shown in Table 3, are consistent with those on CIFAR-100. Our method outperforms iCaRL and cGAN, with a large margin to the latter. In a word, we carry out experiments on simple digits dataset, complex natural image dataset and more challenging fine-grained image dataset. The results show that our method can achieve state-of-the-art performance, while significantly outperforms existing generative replay methods.

The inference time of our method is almost the same with other baselines. However the training time of our method is 1.1 and 2.5 times as long as that of cGAN and iCaRL.

Ablation Study We carry out experiments on CIFAR-100 with incremental learning of 5 stages, to verify the effectiveness of the components of our collaborative framework. The results are shown in Table 4. We implement a base model, which only contains $\{E, G, D\}$. The classification accuracy of this base model at the end of incremental learning is 50.38%, which is much better than what cGAN achieves (44.79%), showing the superiority of the base model. Guided sampling (GS) enables the model to learn from informative generated samples only. If we remove guided sampling from our full model, which means pseudo-images generated from previous stage are randomly sampled, the accuracy drop is about 4.6%. It's worth noticing that our guided sampling method can also boost the performance of cGAN. Similarly, when we remove sampling and weighting (SW), the performance degrades by 1.8%. This is because with SW, the system can start learning from the easy samples and gradually select the harder ones, thereby achieving better convergence. Domain adaptors (DA) designed for feature alignment are good supplements for the base model to alleviate domain shift. Without DA the performance of the system degrades by 2.1%. The experimental results show that each of these components plays important role in the whole system.

Conclusion

This work develops a novel framework to train neural networks end-to-end for class incremental learning. We use a well-designed deep generator to generate samples from learned data distribution. Meanwhile, a discriminative learning model is developed to incrementally update the latent feature space. More importantly, the generator and the discriminative model are connected through invertible mapping between latent feature space and real image space. To ease the training process of the collaborative models and provide high-quality samples for the training of following stage, an effective sample selection method is proposed. With collaboration of these components, our system can effectively preserve the previous learned knowledge and reduce the ambiguities between old and new classes. Extensive experiments on visual classification datasets demonstrate that our approach outperforms or is on par with strong distillation based approaches, and brings significant improvements over existing generative replay methods.

Acknowledgments

This work was supported by the Strategic Priority Research Program of the Chinese Academy of Sciences (XDB32040200) and Beijing Academy of Artificial Intelligence.

References

Banerjee, A.; Dhillon, I. S.; Ghosh, J.; and Sra, S. 2005. Clustering on the unit hypersphere using von Mises-Fisher distributions. *Journal of Machine Learning Research* 6(Sep): 1345–1382.

Castro, F. M.; Marín-Jiménez, M. J.; Guil, N.; Schmid, C.; and Alahari, K. 2018. End-to-end incremental learning. In *ECCV*, 233–248.

Gong, M.; Xu, Y.; Li, C.; Zhang, K.; and Batmanghelich, K. 2019. Twin auxilary classifiers GAN. In *Advances in Neural Information Processing Systems*, 1330–1339.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2672–2680.

Hashimoto, T. B.; Guu, K.; Oren, Y.; and Liang, P. 2018. A retrieve-and-edit framework for predicting structured outputs. In *Advances in Neural Information Processing Systems*, 10073–10083.

Hayes, T. L.; Kafle, K.; Shrestha, R.; Acharya, M.; and Kanan, C. 2020. Remind your neural network to prevent catastrophic forgetting. In *ECCV*, 466–483.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*, 770–778.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* .

Iscen, A.; Zhang, J.; Lazebnik, S.; and Schmid, C. 2020. Memory-efficient incremental learning through feature adaptation. In *ECCV*, 699–715.

Kemker, R.; and Kanan, C. 2018. FearNet: Brain-inspired model for incremental learning. In *ICLR*.

Kingma, D. P.; and Ba, J. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Kingma, D. P.; and Welling, M. 2014. Auto-encoding variational bayes. In *ICLR*.

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* 114(13): 3521–3526.

Krizhevsky, A. 2009. Learning multiple layers of features from tiny images. Technical report, University of Toronto.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 1097–1105.

Larsen, A. B. L.; Sønderby, S. K.; Larochelle, H.; and Winther, O. 2016. Autoencoding beyond pixels using a learned similarity metric. In *ICML*, 1558–1566.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86: 2278–2324.

Li, J.; Chen, E.; Ding, Z.; Zhu, L.; Lu, K.; and Huang, Z. 2019. Cycle-consistent conditional adversarial transfer networks. In *ACM Multimedia*, 747–755.

Li, Z.; and Hoiem, D. 2016. Learning without forgetting. In *ECCV*, 614–629.

Liu, X.; Wu, C.; Menta, M.; Herranz, L.; Raducanu, B.; Bagdanov, A. D.; Jui, S.; and van de Weijer, J. 2020. Generative feature replay for class-incremental learning. In *CVPR Workshops*.

McCloskey, M.; and Cohen, N. J. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of Learning and Motivation* 24: 109–165.

Mirza, M.; and Osindero, S. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784* .

Mo, S.; Kim, C.; Kim, S.; Cho, M.; and Shin, J. 2019. Mining GOLD samples for conditional GANs. In *Advances in Neural Information Processing Systems*, 6138–6149.

Rebuffi, S.-A.; Kolesnikov, A.; Sperl, G.; and Lampert, C. H. 2017. iCaRL: Incremental classifier and representation learning. In *CVPR*, 2001–2010.

Sener, O.; and Savarese, S. 2018. Active learning for convolutional neural networks: A core-set approach. In *ICLR*.

Shin, H.; Lee, J. K.; Kim, J.; and Kim, J. 2017. Continual learning with deep generative replay. In *Advances in Neural Information Processing Systems*, 2994–3003.

Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A. 2015. Going deeper with convolutions. In *CVPR*, 1–9.

Van Der Maaten, L. 2014. Accelerating t-SNE using tree-based algorithms. *Journal of Machine Learning Research* 15(1): 3221–3245.

Wang, T.; and Isola, P. 2020. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *ICML*, 9929–9939.

Welinder, P.; Branson, S.; Mita, T.; Wah, C.; Schroff, F.; Belongie, S.; and Perona, P. 2010. Caltech-UCSD birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology.

Wu, Y.; Chen, Y.; Wang, L.; Ye, Y.; Liu, Z.; Guo, Y.; and Fu, Y. 2019. Large scale incremental learning. In *CVPR*, 374–382.

Xiang, Y.; Fu, Y.; Ji, P.; and Huang, H. 2019. Incremental learning using conditional adversarial networks. In *ICCV*, 6619–6628.

Yoon, J.; Yang, E.; Lee, J.; and Hwang, S. J. 2018. Lifelong learning with dynamically expandable networks. In *ICLR*.

Zeng, G.; Chen, Y.; Cui, B.; and Yu, S. 2019. Continual learning of context-dependent processing in neural networks. *Nature Machine Intelligence* 1(8): 364–372.

Zenke, F.; Poole, B.; and Ganguli, S. 2017. Continual learning through synaptic intelligence. In *ICML*, 3987–3995.

Zhang, W.; Ouyang, W.; Li, W.; and Xu, D. 2018. Collaborative and adversarial network for unsupervised domain adaptation. In *CVPR*, 3801–3809.

Zhu, J.-Y.; Park, T.; Isola, P.; and Efros, A. A. 2017a. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2223–2232.

Zhu, J.-Y.; Zhang, R.; Pathak, D.; Darrell, T.; Efros, A. A.; Wang, O.; and Shechtman, E. 2017b. Toward multimodal image-to-image translation. In *Advances in Neural Information Processing Systems*, 465–476.