Prototype Augmentation and Self-Supervision for Incremental Learning

Fei Zhu[1,2], Xu-Yao Zhang[1,2]*, Chuang Wang[1,2], Fei Yin[1,2], Cheng-Lin Liu[1,2,3]
[1]NLPR, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China
[3]CAS Center for Excellence of Brain Science and Intelligence Technology, Beijing 100190, China
zhufei2018@ia.ac.cn, {xyz, fyin, liucl}@nlpr.ia.ac.cn, wangchuang@ia.ac.cn

Abstract

Despite the impressive performance in many individual tasks, deep neural networks suffer from catastrophic forgetting when learning new tasks incrementally. Recently, various incremental learning methods have been proposed, and some approaches achieved acceptable performance relying on stored data or complex generative models. However, storing data from previous tasks is limited by memory or privacy issues, and generative models are usually unstable and inefficient in training. In this paper, we propose a simple non-exemplar based method named PASS, to address the catastrophic forgetting problem in incremental learning. On the one hand, we propose to memorize one class-representative prototype for each old class and adopt prototype augmentation (protoAug) in the deep feature space to maintain the decision boundary of previous tasks. On the other hand, we employ self-supervised learning (SSL) to learn more generalizable and transferable features for other tasks, which demonstrates the effectiveness of SSL in incremental learning. Experimental results on benchmark datasets show that our approach significantly outperforms non-exemplar based methods, and achieves comparable performance compared to exemplar based approaches.

1. Introduction

Incremental learning (IL) enables humans to acquire novel experience continually while maintaining existing knowledge. In dynamic and open environment, it is critical for modern artificial intelligence to have the ability of IL because training examples in real-world applications usually appear sequentially. For instance, a face recognition system may encounter new faces which need to be added and learned throughout its life without forgetting or re-learning the people already learned. However, deep neural networks (DNNs) tend to adjust the learned parameters to new task and almost fully forget previously acquired knowledge. Motivated by this, a multitude of works [28, 34, 43, 51, 42] have recently emerged that try to alleviate the catastrophic forgetting [16, 37, 13] problem. In this paper, we consider a challenging scenario of class-incremental learning (CIL), in which each task in the sequence contains a set of classes disjoint from the old tasks, and the model need to learn a unified classifier that can classify all classes seen at different stages without the task-identifier at inference time.

Intuitively, catastrophic forgetting is caused by overlapping or confusion between the representations of new and old classes in the feature space. When learning new classes, *the decision boundary for previous classes can be dramatically changed, and the unified classifier is severely biased*. To address this issue and maintain previous knowledge, one can store a fraction of old data to jointly train the model with current data [50, 43, 50, 6, 12]. However, storing data is undesirable due to memory limits or privacy issues, in which the data are not allowed to be stored. An alternative way is to learn deep generative models to generate pseudo-samples of previous classes [46, 49, 51, 25]. Nevertheless, it is inefficient to train big generative models such as GAN [17, 3] and autoencoder [27, 25] for complex datasets (e.g., natural images). Moreover, the generative models also suffer from catastrophic forgetting. Another direction is to identify and penalize future changes to some important parameters of the original model [28, 54]. These regularization strategies are effective in scenarios where multi-head classifiers are used and the task-identifier is available at inference. However, as noticed in some works [23, 48], those methods show poor performance in CIL scenario.

Besides the catastrophic forgetting, another obstacle for IL is the task-level overfitting phenomenon, which has been ignored by previous works. Specifically, DNNs can easily overfit to the training task when learning task continually. Intuitively, the model may focus on capturing features that are useful for current task, while discarding those less discriminative directions which could capture data characteristics for future tasks. This may not be a problem for common

*Corresponding author.

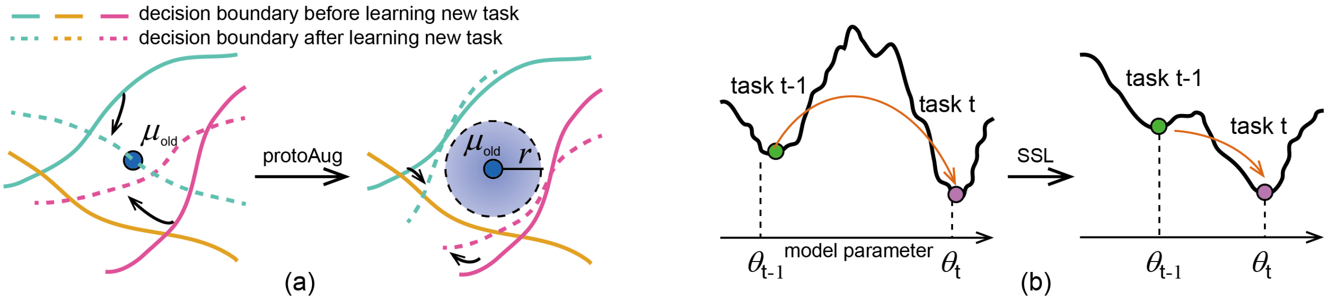


Figure 1: Motivation of **PASS**. (a) When learning new task, the decision boundary of previous tasks could be dramatically changed, resulting in catastrophic forgetting. ProtoAug is proposed to restrain the decision boundary, thus maintaining the discrimination and balance between old and new classes. (b) If the learned features are task-specific in each stage, the model trained on previous task might be a bad initialization for current task. We propose to leverage the benefit of SSL to learn richer and more transferable features. Intuitively, different tasks would be closer in the parameter space, and it would be easier to find a model to perform well on all tasks, thus improving both the stability and plasticity of the model.

single task learning scenario, but leads particles influence for IL since the model for current task is initialized with previous model. A recent study [42] found that a model trained from scratch using samples stored can surprisingly outperforms many recently proposed algorithms. This study indicates that *the previous model, which mainly carries task-specific features, might be a bad initialization for current task*, as shown in Fig. 1(b). Consequently, the model would need more updates to perform well on current task, which increases the forgetting problem on the other hand.

Motivated by the above analysis, we propose to improve CIL performance by maintaining the decision boundary and reducing task-level overfitting phenomenon, as shown in Fig. 1. The proposed **PASS** mainly consists of **P**rototype **A**ugmentation and **S**elf-**S**upervision. On the one hand, prototype augmentation (protoAug) memorizes *one* class-representative prototype (typically the class mean in the deep feature space) for each old class, and augments the memorized prototypes via Gaussian noise when learning new classes. Then, the augmented prototypes and deep features of new data are jointly classified to maintain the discrimination and balance between old and new classes. This is inspired by a recent work [35] in long-tailed recognition which expands the distribution of the tail classes by augmenting the tail classes with certain disturbances. While [35] focuses on class-imbalance learning and learns the embedding augmentation strategy from the head classes, in our work, we focus on CIL and investigate the value of simple Gaussian noise based augmentation.

On the other hand, we take inspiration from self-supervised learning (SSL) to alleviate task-level overfitting phenomenon in IL. In particular, SSL aims to learn transferable representations that would be useful for other tasks. Inspired by the natural connection between IL and SSL, we propose to leverage the benefit of SSL to learn task-agnostic and transferable representations. Intuitively, with SSL, different tasks would be closer in the parameter space, and the model trained on current task would be a better initialization for learning the next task. In conclusion, our main contributions are summarized as follows:

- We propose a simple and effective non-exemplar based method to overcome catastrophic forgetting problem in CIL by memorizing and augmenting prototypes of old classes in the deep feature space.
- We emphasize the task-level overfitting phenomenon in IL, and adopt self-supervised learning to learn more generalizable and transferable features.
- Our method significantly outperforms non-exemplar based methods and obtains comparable results compared to exemplar based methods in CIL scenario.

2. Related Work

Incremental Learning. IL has been a long-standing research topic. Several early approaches used nearest class mean classifier [38] or random forest [47] for IL based on fixed data representations. Recently, a variety of attempts have been made to enable IL for DNNs. Regularization strategies such as elastic weight consolidation (EWC) [28], synaptic intelligence (SI) [54], and memory aware synapses (MAS) [2] use different metrics to identify and penalize the changes of important parameters of the original network when learning a new task. An alternative solution is to perform implicit regularization by using knowledge distillation technique [34, 21]. Nevertheless, it is hard to design a reasonable metric to evaluate the importance of parameters of a model, and the performances of regularization strategies based methods for CIL remain significantly inferior to those obtained by joint training. Recently, Yu et al., [53] found that embedding network suffers less forgetting for CIL. However,

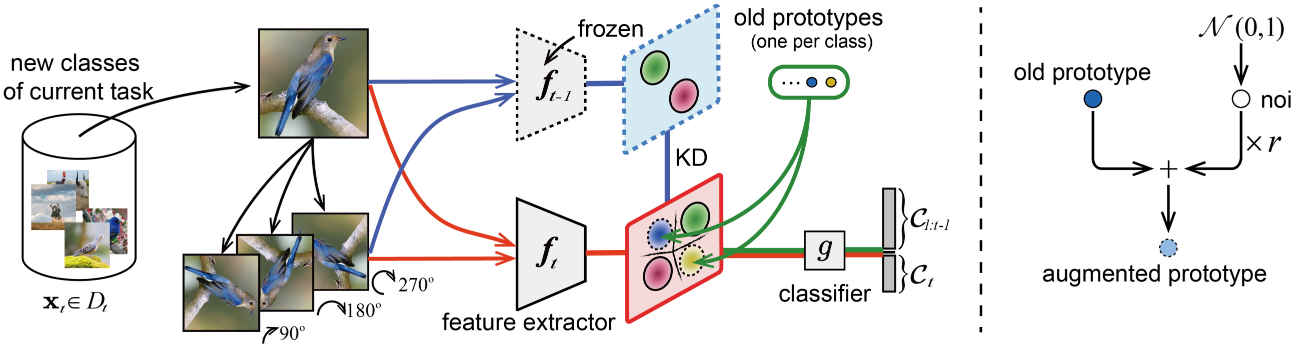


Figure 2: Illustration of PASS for CIL. The classes of current task are augmented by rotation based transformation [32], and the augmented data are fed to the feature extractor. In the deep feature space, we augment the memorized prototypes (one for each classes) via Gaussian noise (right). Our method is non-exemplar based, simple and effective.

training embedding network with metric learning could often be harder than softmax-based networks.

Another direction is rehearsal strategies, which provide a strong baseline for CIL by storing and replaying a fraction of samples from the old classes. With stored samples, some works [50, 43, 12] use a distillation loss to prevent forgetting, while others [44, 8] only include classification loss and construct each mini-batch with an equal amount of new data and the rehearsal data. More recently, the imbalance problem between the previous and current tasks has been found to be constituting a key challenge for CIL, and several works, such as EEIL [6], BiC [50], UCIR [22] and WA [56] were proposed to reduce the bias towards currents tasks. However, those techniques may not be applicable without storing data. Without directly storing raw data, a line of work [46, 49, 51] sequentially constructs a separate generative model to generate old samples. Nevertheless, those approaches rely heavily on the quality of the generative model. In this paper, we aim to reduce catastrophic forgetting in CIL without storing old data or leveraging complex generative models.

Self-Supervised Learning. Recently, learning with self-supervision [24] has been demonstrated effective to learn general representations, by learning some proxy tasks, e.g. prediction rotations [15], patch permutation [40], image colorization [30] and clustering [4, 5]. More recently, contrastive losses based SSL methods [9, 18] show great success. By SSL, the model could learn features that are unnecessary for current task but useful for other tasks, e.g., semi-supervised learning [55], few-shot learning [14], and improving robustness [20]. In particular, it has been found that self-supervised pretraining is a good choice to initialize the model for class-imbalance learning [52]. Lee et al., [32] propose to augment original labels via self-supervision of input transformation, and show that the supervised classification accuracy could be improved by this simple technique. Inspired by the natural connection between IL and SSL, we employ the self-supervised method in [32] to investigate SSL in CIL, revealing surprising yet intriguing findings that SSL can boost the performance of CIL significantly.

3. Methodology

3.1. Problem Statement and Analysis

The goal of CIL is to sequentially learn a unified model to classify the test samples of all classes that have been learned so far. Specifically, the model consists of two parts: the feature extractor F_θ and a unified classifier G_ϕ. Let $D = \{D_t\}_{t=1}^{T}$ be a stream of data, where $D_t = \{\mathbf{X}_t, \mathbf{Y}_t\} = \{x_{t,j}, y_{t,j}\}_{j=1}^{N_t}$ is the dataset that the system receives at step t. Dataset D_t consists of N_t labeled samples for training, and $y_{t,j} \in C_t$, where C_t is the class set of task t and the class sets of different task are disjoint. At step t, the goal is to minimize a predefined loss function L on new dataset D_t without interfering with and possibly improving on those that were learned previously [1]:

$$\begin{array}{r}\{\theta_t, \phi_t\} = \underset{\theta_t, \phi_t, \epsilon}{\operatorname{argmin}} \, L_t(G(F(\mathbf{X}_t; \theta_t); \phi_t), \mathbf{Y}_t) + \sum \epsilon_i \\ \text{s.t. } L_t(\mathbf{X}_i, \mathbf{Y}_i) - L_i(\mathbf{X}_i, \mathbf{Y}_i) \leqslant \epsilon_i, \epsilon_i \geqslant 0;\ \forall i \in [1, t-1]\end{array}, \tag{1}$$

where $L_t(\mathbf{X}_i, \mathbf{Y}_i) = L(G(F(\mathbf{X}_i; \theta_t); \phi_t), \mathbf{Y}_i)$ is the loss of the model at t on old data set D_i and $L_i(\mathbf{X}_i, \mathbf{Y}_i) = L(G(F(\mathbf{X}_i; \theta_i); \phi_i), \mathbf{Y}_i)$ is the loss of the previous model at i on old dataset D_i. The last term $\epsilon = \{\epsilon_i\}$ is a slack variable that tolerates a small increase in old dataset.

There are mainly two obstacles in CIL: classifier bias and task-level overfitting. First, with only new data, the decision boundary learned previously can be dramatically changed, and the unified classifier is severely biased. Second, it is difficult to learn general features which could be generalizable well on other classes with data only for current classes. As a result, the feature extractor is also biased and the parameter space of model at different stages would be far, which makes it difficult to find a model to perform well on all tasks. Therefore, from a multi-task learning perspective, learning task-agnostic representations is important for CIL.

Overview of Framework. The framework of our method is shown in Fig. 2. Specifically, for each old class, we do not store any old samples, but to memorize a class-representative prototype in the deep feature space. Then, when learning new task, each old prototype is augmented with certain disturbances and fed to the unified classifier for classification. Consequentially, it alleviates the distortion of the learned feature space and the classifier bias. In addition, to reduce the task-level overfitting, SSL is adopted to learn more general features for other (previous and future) tasks by using rotation-based label augmentation [32].

3.2. Prototype Augmentation

At stage t, only D_t is available for training, thus we can not directly optimize Eq. (1). To alleviate distortion of the feature space when learning new task, we compute and memorize one prototype (class mean) for each classes:

$$\mu_{t,k} = \frac{1}{N_{t,k}} \sum_{n=1}^{N_{t,k}} F(\mathbf{X}_{t,k}; \theta_t). \tag{2}$$

When learning new task, the prototype of each old class, e.g. class k_{old} at stage t_{old}, is augmented as below (shown in Fig. 2):

$$F_{t_{old},k_{old}} = \mu_{t_{old},k_{old}} + e * r, \tag{3}$$

where $e \sim \mathcal{N}(0,1)$ is the derived Gaussion noise which has the same dimension as prototype. r is a scale to control the uncertainty of the augmented prototypes. In particular, the scale r can be pre-defined, or computed as the average variance of the class representations:

$$r_t^2 = \frac{1}{K_{old} + K_{new}} (K_{old} * r_{t-1}^2 + \sum_{k=1}^{K_{new}} \frac{\text{Tr}(\Sigma_{t,k})}{D}), \tag{4}$$

where K_{old} and K_{new} represent the number of old classes and new classes at stage t, respectively. D is the dimension of the deep feature space. $\Sigma_{t,k}$ is the covariance matrix for the features from class k at stage t, and the Tr operation computes the trace of a matrix. We observed that the r_t changes slightly at different stage in the course of a CIL experiment. Therefore, one can only compute and use the average variance of the features in the first task as follows: $r^2 = r_1^2 = \frac{1}{K_1 * D} \sum_{k=1}^{K_1} \text{Tr}(\Sigma_{1,k})$.

Then, the features of new classes and the augmented prototypes are feed to the unified classifier. As a result, Eq. (1) could be empirically approximated by Eq. (5):

$$\begin{aligned} \{\theta_t, \phi_t\} = \underset{\theta_t, \phi_t, \epsilon}{\text{argmin}} \{ & L_t(G(F(\mathbf{X}_t; \theta_t); \phi_t), \mathbf{Y}_t) \\ & + \sum_{i=1}^{t-1} L(G(F_i; \phi_t), \mathbf{Y}_i)\}, \end{aligned} \tag{5}$$

where F_i represents the features augmented for old class set C_i. Intuitively, in the feature space, the prototypes of old classes are augmented with soft variance, which represents the confidence of reality of the features generated. During training with current data, the augmented features are feed to classifier to maintain discrimination and balance among all classes that have been learned so far.

3.3. SSL based Label Augmentation

Inspried by [32], we simply learn a unified model by augmenting the current class based on SSL. Specifically, for each class, we rotate its training data 90, 180, and 270 degrees to generate 3 novel classes, extending the original K-class problem to a new 4K class problem:

$$\mathbf{X'}_t = rotate(\mathbf{X}_t, \theta), \theta \in \{90, 180, 270\}, \tag{6}$$

and the augmented sample is assigned a new label $\mathbf{Y'}_t$. Comparing the widely used 4-way self-supervised tasks, as demonstrated in [32], the above approach relaxes a certain invariant constraint during learning the original and self-supervised tasks simultaneously, which is beneficial to learning richer features. As shown in our experiments, the performance of CIL can be improved by this simple method.

3.4. Integrated Objective of PASS

When learning new classes, the feature extractor would be updated continually. To alleviate the mismatch between the saved old prototypes and the feature extractor, the well-known knowledge distillation (KD) [21, 22] is employed to regularize the feature extractor. Specifically, we restrain the feature extractor by matching the features of new data extracted by current model with that of previous model:

$$L_{t,kd} = \|F_t(\mathbf{X'}_t; \theta_t) - F_{t-1}(\mathbf{X'}_t; \theta_{t-1})\|. \tag{7}$$

Combining the techniques presented above, we reach a total loss of PASS that comprised of three terms, given as:

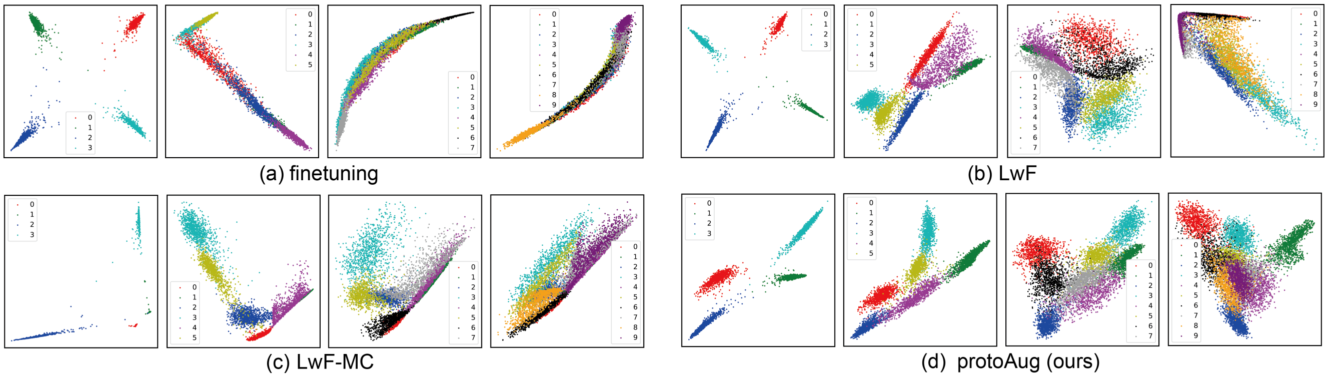
$$L_{t,total} = L_{t,ce} + \lambda * L_{t,protoAug} + \gamma * L_{t,kd}. \tag{8}$$

$L_{t,ce} = L_{t,ce}(G(F(\mathbf{X'}_t; \theta_t); \phi_t), \mathbf{Y'}_t)$, and $L_{t,protoAug} = \sum_{i=1}^{t-1} L_{t,ce}(G(F_i; \phi_t), \mathbf{Y}_i)$. λ and γ are loss weights, and we use $\lambda = \gamma = 10$ in our experiments.

3.5. Preliminary Experiments

3.5.1 2D Visualization of ProtoAug

To provide an illustration of protoAug, we conduct experiment on MNIST [31] with a *2-dimensional* feature space which is suitable for visualization. SSL is not applied here since the effect of protoAug is the focus in this experiment. We start from a Resnet-18 model [19] trained on 4 classes and the remaining 6 classes are continually added in 3 phases. We compare our method with finetuning, LwF [34], and LwF-MC (binary cross entropy based) [43]. As shown in Fig. 3, the distribution of old classes is dramatically changed


(a) finetuning

(b) LwF

(c) LwF-MC

(d) protoAug (ours)

Figure 3: Visualization of class representations in the feature space when learning MNIST [31] incrementally. The outputted features are *2-dimensional* which is suitable for visualization. Best viewed in color.

Table 1: Results of zero-cost class incremental learning. The model is tested using nearest class mean classifier.

#classes			4 (base)	5	6	7	8	9	Final	Average
CIFAR-10	Novel	Baseline	—	27.20	20.55	17.40	17.23	15.68	14.80	18.81
		+ SSL	—	**76.40**	**61.10**	**46.83**	**40.80**	**40.36**	**37.57**	**50.50**$_{+31.69}$
	All	Baseline	94.55	79.26	68.00	59.65	52.88	48.97	46.46	64.25
		+ SSL	**95.35**	**87.26**	**79.22**	**70.04**	**64.05**	**61.18**	**58.36**	**73.63**$_{+9.38}$
#classes			40 (base)	50	60	70	80	90	Final	Average
CIFAR-100	Novel	Baseline	—	43.50	33.10	30.43	27.45	25.20	23.58	30.54
		+ SSL	—	**55.70**	**44.85**	**42.67**	**38.37**	**34.70**	**32.15**	**41.46**$_{+10.92}$
	All	Baseline	71.83	63.60	55.73	50.64	46.38	42.61	39.37	52.93
		+ SSL	**72.03**	**64.52**	**58.37**	**54.46**	**50.50**	**46.48**	**43.46**	**55.68**$_{+2.74}$

in finetuning, and there is an obvious overlap of distribution from different classes, resulting in catastrophic forgetting. Contrarily, our method can maintain the distribution of old classes when learning new classes, thus reduces the forgetting phenomenon in the course of CIL.

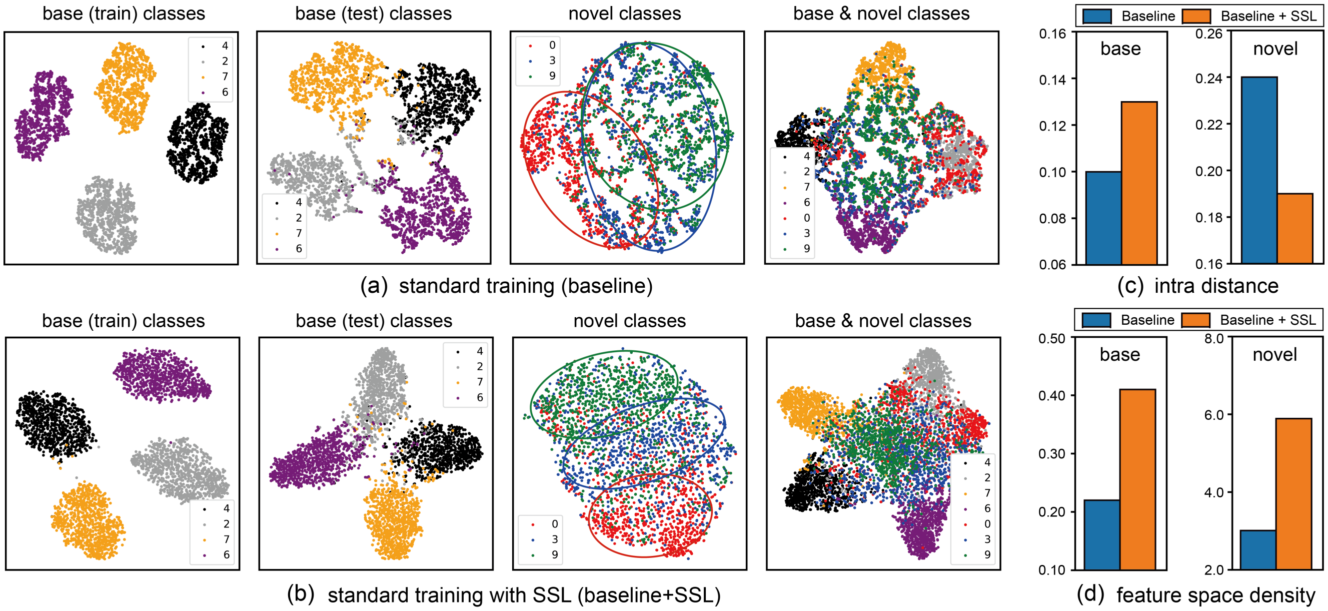
3.5.2 A Closer Look at SSL for CIL

Setup. We train ResNet-18 for classifying CIFAR-10 and CIFAR-100 [29]. Similar to [33, 39], we first train a classification model on some base classes. Then a nearest class mean (NCM) classifier is built on the pre-trained feature extractor to classify both base and new classes incrementally. For SSL based model, the based classes are augmented using the label augmentation method in Section 3.3. We train all the models for 120 epochs with batch size 64 and Adam [26] optimizer with 0.001 initial learning rate, and the learning rate is multiplied by 0.1 after 50 and 100 epochs.

Results. For each learning stage, we report the test accuracy on novel classes that appeared so far. And we also test on both base and novel classes that appeared so far. As shown in Table 1, the accuracy of novel classes can be significantly improved with SSL. For instance, SSL based model achieves 50.50% average incremental task accuracy on novel classes of CIFAR-10, and surpasses the baseline model by a large margin of 31.69%. Similarly, on CIFAR-100, SSL based model outperforms the baseline model by a margin of 10.92%. Those results strongly demonstrate the suitability and effectiveness of SSL for CIL.

Deep feature space anaysis. An intuitively explanation for the effectiveness of SSL on the above experiments is that SSL improves the separation of the distribution of novel classes. As shown in Fig. 4, the class representations of novel classes are much more separated with SSL, and the overlap between base and novel classes is less, comparing with baseline model. We further anaysis the deep feature space quantitatively. Specifically, we use average inter-class distances $\pi_{inter}(F) = \frac{1}{Z_{inter}} \sum_{y_l, y_k, l \neq k} d(\mu(F_{y_l}), \mu(F_{y_k}))$, and average intra-class distances $\pi_{intra}(F) = \frac{1}{Z_{intra}} \sum_{y_l \in y} \sum_{f_i, f_j \in F_{y_l}, i \neq j} d(f_i, f_j)$ to measure the distribution of class representations. $d(\cdot;\cdot)$ is the cosine distance in our experiment. $F_{y_l} = \{f_i := f_\theta(x_i) | x_i \in X, y_i = y_l\}$ denotes the set of embedded samples of a class y_l. $\mu(F_{y_l})$ is their mean embedding. Z_{intra} and Z_{inter} are two normalization constants.

As shown in Fig. 4, for unseen classes, SSL results in smaller intra distance on novel classes, which implies that



(a) standard training (baseline)

(c) intra distance

(b) standard training with SSL (baseline+SSL)

(d) feature space density

Figure 4: (a-b) SSL improves the separation of the distribution of novel classes, and reducing the the overlap between base and novel classes. (c-b) SSL results in smaller intra distance on novel classes, and high feature space density.

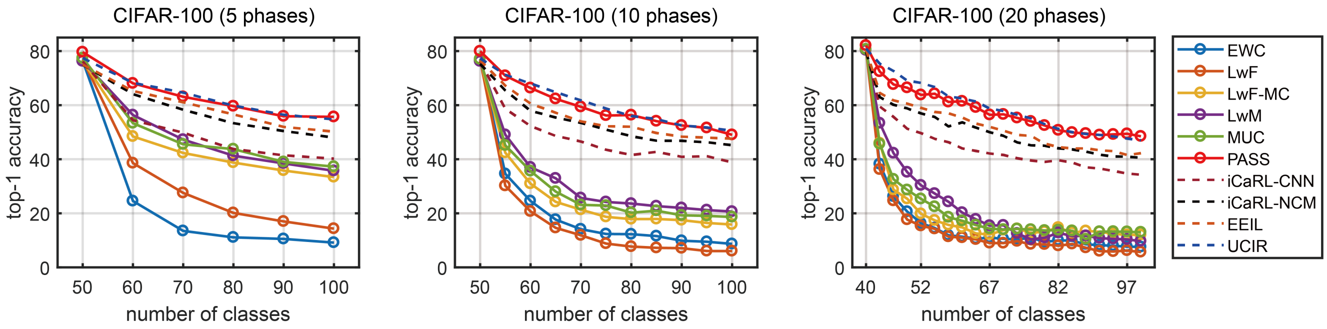


Figure 5: Results of classification accuracy on CIFAR-100, which contains 5, 10 and 20 sequential tasks.

the model learned with SSL generalizes better than baseline on novel classes. While for training classes, baseline has more compact feature distributions. This indicates that representation learning for new class generalization may be hurt by excessive feature compression. In particular, Roth et al., [45] proposed a concept of feature space density: $\pi_{ratio}(F) = \pi_{intra}(F)/\pi_{inter}(F)$, and found that an increased feature space density π_{ratio} is linked to stronger generalization under considerable shifts between training and testing distribution. Fig. 4(d) shows that SSL leads to a higher feature space density π_{ratio}, and the improvement on generalization is consistent with the observation in [45].

4. Experiments

Datasets. We perform our experiments on CIFAR-100 [29], TinyImageNet [41] and ImageNet-Subset [10]. The classes are arranged in a fixed random order. Except for one setting on CIFAR-100, we mainly train the model on half of classes for the first task, and equal classes in the rest phases.

Comparison Approaches. We compare our method (PASS) with non-exemplar based methods such as EWC [28], LwF [34], LwF-MC [43], LwM [11] and MUC [36]. We also compare with several state-of-the-art exemplar-based approaches: iCaRL [43], EEIL [6], UCIR [22]. Note that our method is non-exemplar based since we do not save any old samples, but to memorize one prototype in the deep feature space for each class, which is very memory efficient and has no privacy issues.

Evaluation metrics. We report the standard metrics to measure the quality of CIL: ***Accuracy*** [43] is computed as the average accuracy of all the *classes* that have already been learned. ***Average forgetting*** [7] is defined to estimate the forgetting of previous tasks. The forgetting measure

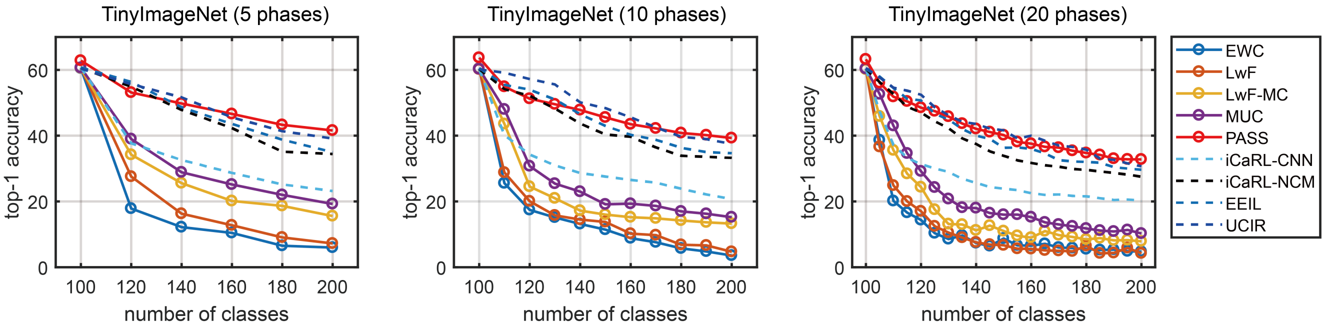


Figure 6: Results of classification accuracy on TinyImageNet, which contains 5, 10 and 20 sequential tasks.

f_k^i of the i-th task after training k-th task is defined as $f_k^i = \max\limits_{t \in 1,\dots,k-1} (a_{t,i} - a_{k,i}), \forall i < k$, in which $a_{m,n}$ is the accuracy of task n after training task m. The average forgetting measure F_k is then defined as $F_k = \frac{1}{k-1} \sum_{i=1}^{k-1} f_k^i$.

Implementation details.[1] ResNet-18 [19] is used and trained from scratch in our experiments. We train all the models with batch size 64 and Adam [26] optimizer with 0.001 initial learning rate. We train all the models for 100 epochs, and the learning rate is multiplied by 0.1 after 45 and 90 epochs. All the experiments are repeated three times and the average results are reported. We conduct different incremental settings (5, 10 and 20 phases) for both CIFAR-100 and TinyImageNet. For ImageNet-Subset, we use the 10 incremental phases evaluation protocol. After each phase, the model is evaluated on all the learned classes so far. For the exemplar-based approaches: iCaRL [43], EEIL [6], UCIR [22], we use *herd selection* [43] to select and store 20 samples per old class, which is a common setting [43, 22].

4.1. Comparative Results

Results are shown in Fig. 5, Fig. 6 and Fig. 7. We observe that our method outperforms significantly better than non-exemplar based methods, which confirms that PASS can effectively address the catastrophic forgetting in CIL without storing old training samples. Take the results of 10 phases as an example, our method outperforms the best non-exemplar methods MUC [36] with a gap of 29.3% on CIFAR-100 and with a gap of 25.2% on TinyImageNet. In addition, our method outperforms the strong baseline method, iCaRL-NCM [43], by 3.7% on CIFAR-100 (10 phases), and achieves comparable accuracy with state-of-the-art exemplar-based approaches which are based on many saved samples overall. The observations on ImageNet-Subset are consistent with those on CIFAR-100 and TinyImageNet.

To compare the effectiveness of alleviating forgetting, we show the average forgetting results in Table 2. Our method suffers from less forgetting than iCaRL-NCM on CIFAR-100. The results on TinyImageNet are also conclusive. In conclusion, PASS outperforms all the non-exemplar based methods and some exemplar based methods in terms of both accuracy and average forgetting.

Table 2: Results of average forgetting on CIFAR-100 and TinyImageNet.

	CIFAR-100			TinyImageNet		
Method	5 phases	10 phases	20 phases	5 phases	10 phases	20 phases
LwF_MC	44.23	50.47	55.46	54.26	54.37	63.54
MUC	40.28	47.56	52.65	51.46	50.21	58.00
PASS	25.20	30.25	30.61	**18.04**	23.11	**30.55**
iCaRL-CNN	42.13	45.69	43.54	36.89	36.70	45.12
iCaRL-NCM	24.90	28.32	35.53	27.15	28.89	37.40
EEIL	23.36	26.65	32.40	25.56	25.91	35.04
UCIR	**21.00**	**25.12**	**28.65**	20.61	**22.25**	33.74

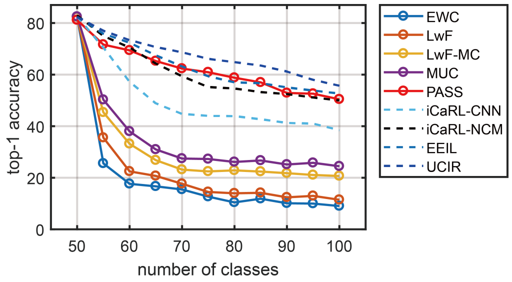


Figure 7: Results of classification accuracy on ImageNet-Subset, which contains 10 sequential tasks.

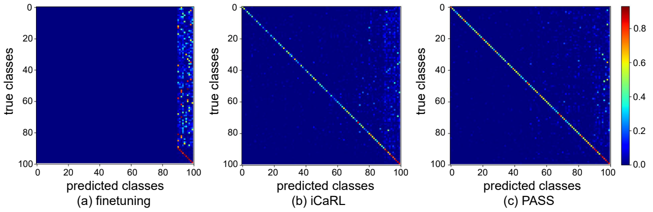
The comparison of the confusion matrix. Fig. 8 shows the comparison of confusion matrix by finetuning, iCaRL, and our approach. The diagonal entries represent the correction predictions and off-diagonal entries represent the misclassification. Because of the severe imbalance between old and new classes, finetuning tends to classify the samples into new classes (strong confusions on the last task), as shown in Fig. 8(a). PASS is capable to remove most of the bias and achieves better overall performance without relying on stored data of old classes.

The comparison of weight in the FC Layer. For the ex-

[1] Code available at https://github.com/Impression2805/CVPR21_PASS.

Table 3: The effectiveness of each component in our method.

#dataset & classes				CIFAR-100			TinyImageNet		
	Method	protoAug	SSL	5 phases	10 phases	20 phases	5 phases	10 phases	20 phases
Accuracy	KD	✗	✗	14.33	6.04	5.67	7.23	4.70	4.23
	KD+SSL	✗	✓	17.15	8.46	8.57	9.71	6.53	6.60
	KD+protoAug	✓	✗	50.19	39.80	38.61	33.11	26.52	20.97
	KD+protoAug+SSL	✓	✓	**55.67**	**49.03**	**48.48**	**41.58**	**39.28**	**32.78**
Forgetting	KD+protoAug	✓	✗	28.72	35.70	40.59	25.62	35.33	43.91
	KD+protoAug+SSL	✓	✓	**25.20**	**30.25**	**30.61**	**18.04**	**23.12**	**30.55**



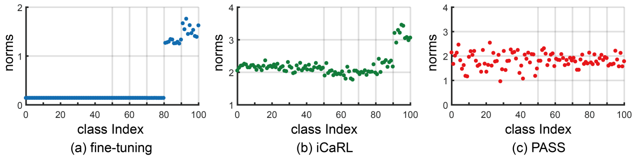
(a) finetuning (b) iCaRL (c) PASS

Figure 8: The comparison of confusion matrix of finetuning, iCaRL and PASS.

periment on CIFAR-100 (5 phases), after the last step, we calculate the norms of the weight vectors and plot them in Fig. 9. As shown in Fig. 9(a), by finetuning, the norms of the weight vectors of new classes are much larger than those of old classes. As a result, an input image can be easily predicted to a new class. Moreover, the weight learned by iCaRL suffers less imbalance problem comparing with finetuning, but the bias still exists in Fig. 9(b). It can be seen from Fig. 9(c) that our method is capable to remove the bias of the weight vectors in the FC Layer.

4.2. Ablation Study

The proposed PASS is comprised of three components: protoAug, SSL, and KD, as shown in Fig. 2. Here we analyze the effect of isolate individual aspects of the methods. From the results in Table 3, we can observe that: (1) Only using KD (as that in LwF) is completely failed in CIL without protoAug and SSL. (2) SSL has a relatively small effect combining with KD since the imbalance problem of the classifier is severe. (3) ProtoAug successfully mitigates the imbalance problem and achieves much better results than KD, e.g., protoAug improves the performance of KD with a margin of 32.94% on CIFAR-100 (20 phases). (4) The performance of protoAug could be significantly improved by combining with SSL, e.g., SSL improves the performance of KD+protoAug with a margin of 9.87% on CIFAR-100 (20 phases). Moreover, it can be seen that the effectiveness of SSL is more obvious with the help of protoAug, which indicates that SSL and protoAug could benefit from each other. Particularly, we have experimentally observed that the performance will drop significantly without KD. As demonstrated in Section 3.4, KD is critical for the success of PASS.



(a) fine-tuning (b) iCaRL (c) PASS

Figure 9: Norms of the weight vectors in the fully connected (FC) layer after learning all classes incrementally. Our method can remove the bias and learn a balance weight.

By employing SSL in CIL, the model could learn more general and transferable features for other tasks (as demonstrated in Section 3.5.2), which can reduce the feature extractor bais. Thus, it would be easier to find a model to perform well on all tasks, which improves both the stability and plasticity of the model. Therefore, we emphasize that the feature extractor bias should be considered and more future effort should be put into task-agnostic representation learning for IL, especially for non-exemplar based CIL.

5. Conclusion

This paper proposes a simple and effective method of *PASS* for CIL. PASS is capable to alleviate the catastrophic forgetting problem in CIL, and achieves significantly better classification results on several datasets without storing exemplar samples for old class or using complex generative models. In particular, we propose to introduce self-supervised learning to incremental learning for better task generalizable features. Extensive experiments demonstrate that our approach outperforms non-exemplar based methods by large margins, and achieves comparable performance compared to several state-of-the-art exemplar-based approaches under different settings.

Acknowledgements

This work has been supported by the Major Project for New Generation of AI under Grant No. 2018AAA0100400, the National Natural Science Foundation of China (NSFC) grants U20A20223, 61633021, 62076236, 61721004, the Key Research Program of Frontier Sciences of CAS under Grant ZDBS-LY-7004, and the Youth Innovation Promotion Association of CAS under Grant 2019141.

References

[1] Rahaf Aljundi. Continual learning in neural networks. *arXiv preprint arXiv:1910.02718*, 2019.

[2] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *ECCV*, pages 139–154, 2018.

[3] Martín Arjovsky, Soumith Chintala, and L. Bottou. Wasserstein generative adversarial networks. In *ICML*, pages 214–223, 2017.

[4] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *ECCV*, pages 139–156, 2018.

[5] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *NeurIPS*, 2020.

[6] Francisco M Castro, Manuel J Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In *ECCV*, pages 233–248, 2018.

[7] Arslan Chaudhry, P. Dokania, Thalaiyasingam Ajanthan, and P. Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *ECCV*, pages 532–547, 2018.

[8] Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. Continual learning with tiny episodic memories. *ICML Workshop: Multi-Task and Lifelong Reinforcement Learning*, 2019.

[9] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020.

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009.

[11] Prithviraj Dhar, Rajat Vikram Singh, Kuan-Chuan Peng, Ziyan Wu, and Rama Chellappa. Learning without memorizing. In *CVPR*, pages 5138–5146, 2019.

[12] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *ECCV*, pages 86–102, 2020.

[13] Robert M French. Interactive tandem networks and the sequential learning problem. Citeseer.

[14] Spyros Gidaris, Andrei Bursuc, Nikos Komodakis, Patrick Pérez, and Matthieu Cord. Boosting few-shot visual learning with self-supervision. In *ICCV*, pages 8059–8068, 2019.

[15] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *ICLR*, 2018.

[16] Ian J. Goodfellow, M. Mirza, Xia Da, Aaron C. Courville, and Yoshua Bengio. An empirical investigation of catastrophic forgeting in gradient-based neural networks. *CoRR*, 2014.

[17] Ian J. Goodfellow, Jean Pouget-Abadie, M. Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014.

[18] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, pages 9729–9738, 2020.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[20] Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Song. Using self-supervised learning can improve model robustness and uncertainty. *arXiv preprint arXiv:1906.12340*, 2019.

[21] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[22] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and D. Lin. Learning a unified classifier incrementally via rebalancing. In *CVPR*, pages 831–839, 2019.

[23] Yen-Chang Hsu, Yen-Cheng Liu, Anita Ramasamy, and Zsolt Kira. Re-evaluating continual learning scenarios: A categorization and case for strong baselines. *arXiv preprint arXiv:1810.12488*, 2018.

[24] Longlong Jing and Yingli Tian. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2020.

[25] Ronald Kemker and Christopher Kanan. Fearnet: Brain-inspired model for incremental learning. In *ICLR*, 2018.

[26] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[27] Diederik P. Kingma and Max Welling. An introduction to variational autoencoders. *Found. Trends Mach. Learn.*, pages 307–392, 2019.

[28] J. Kirkpatrick, Razvan Pascanu, Neil C. Rabinowitz, J. Veness, G. Desjardins, Andrei A. Rusu, K. Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, C. Clopath, D. Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, pages 3521 – 3526, 2017.

[29] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, 2009.

[30] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Learning representations for automatic colorization. In *ECCV*, pages 577–593, 2016.

[31] Yann LeCun and Corinna Cortes. The mnist database of handwritten digits. 2005.

[32] Hankook Lee, Sung Ju Hwang, and Jinwoo Shin. Self-supervised label augmentation via input transformations. In *ICML*, 2020.

[33] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *NeurIPS*, pages 7167–7177, 2018.

[34] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Trans. Pattern Anal. Mach. Intell.*, pages 2935–2947, 2018.

[35] Jialun Liu, Yifan Sun, Chuchu Han, Zhaopeng Dou, and Wenhui Li. Deep representation learning on long-tailed data: A learnable embedding augmentation perspective. In *CVPR*, pages 2967–2976, 2020.

[36] Yu Liu, Sarah Parisot, Gregory G. Slabaugh, Xu Jia, Ales Leonardis, and Tinne Tuytelaars. More classifiers, less forgetting: A generic multi-classifier paradigm for incremental learning. In *ECCV*, pages 699–716, 2020.

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

[38] Thomas Mensink, J. Verbeek, F. Perronnin, and G. Csurka. Distance-based image classification: Generalizing to new classes at near-zero cost. *IEEE Trans. Pattern Anal. Mach. Intell.*, pages 2624–2637, 2013.

[39] Thomas Mensink, Jakob J. Verbeek, Florent Perronnin, and Gabriela Csurka. Distance-based image classification: Generalizing to new classes at near-zero cost. *IEEE Trans. Pattern Anal. Mach. Intell.*, pages 2624–2637, 2013.

[40] M. Noroozi and P. Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, pages 69–84, 2016.

[41] Hadi Pouransari and Saman Ghili. Tiny imagenet visual recognition challenge. *CS231N course, Stanford Univ., Stanford, CA, USA*, 2015.

[42] Ameya Prabhu, Philip HS Torr, and Puneet K Dokania. Gdumb: A simple approach that questions our progress in continual learning. In *ECCV*, pages 524–540, 2020.

[43] Sylvestre-Alvise Rebuffi, A. Kolesnikov, Georg Sperl, and Christoph H. Lampert. icarl: Incremental classifier and representation learning. In *CVPR*, pages 5533–5542, 2017.

[44] Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and Gerald Tesauro. Learning to learn without forgetting by maximizing transfer and minimizing interference. In *ICLR*, 2018.

[45] Karsten Roth, Timo Milbich, Samarth Sinha, Prateek Gupta, Björn Ommer, and Joseph Paul Cohen. Revisiting training strategies and generalization performance in deep metric learning. In *ICML*, 2020.

[46] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. In *NeurIPS*, pages 2994–3003, 2017.

[47] P Shrestha et al. Incremental learning strategies with random forest classifiers. In *WIC Symposium on Information Theory in the Benelux*, pages 1–6, 2011.

[48] Gido M Van de Ven and Andreas S Tolias. Three scenarios for continual learning. *arXiv preprint arXiv:1904.07734*, 2019.

[49] Chenshen Wu, L. Herranz, X. Liu, Y. Wang, Joost van de Weijer, and B. Raducanu. Memory replay gans: Learning to generate new categories without forgetting. In *NeurIPS*, pages 5962–5972, 2018.

[50] Y. Wu, Yan-Jia Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *CVPR*, pages 374–382, 2019.

[51] Ye Xiang, Ying Fu, Pan Ji, and Hua Huang. Incremental learning using conditional adversarial networks. In *ICCV*, pages 6618–6627, 2019.

[52] Yuzhe Yang and Zhi Xu. Rethinking the value of labels for improving class-imbalanced learning. In *NeurIPS*, 2020.

[53] Lu Yu, Bartlomiej Twardowski, X. Liu, L. Herranz, Kai Wang, Yong mei Cheng, Shangling Jui, and Joost van de Weijer. Semantic drift compensation for class-incremental learning. In *CVPR*, pages 6980–6989, 2020.

[54] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *ICML*, pages 3987–3995, 2017.

[55] Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer. S4l: Self-supervised semi-supervised learning. In *ICCV*, pages 1476–1485, 2019.

[56] Bowen Zhao, Xi Xiao, Guojun Gan, Bin Zhang, and Shu-Tao Xia. Maintaining discrimination and fairness in class incremental learning. In *CVPR*, pages 13205–13214, 2020.