TCP: Triplet Contrastive-relationship Preserving for Class-Incremental Learning

Shiyao Li[1], Xuefei Ning[1]*, Shanghang Zhang[2], Lidong Guo[1], Tianchen Zhao[1],
Huazhong Yang[1] and Yu Wang[1]*
[1]Tsinghua University, China, [2]Peking University, China
lishiyao20@mails.tsinghua.edu.cn, foxdoraame@gmail.com, yu-wang@tsinghua.edu.cn

Abstract

In class-incremental learning (CIL), when deep neural networks learn new classes, their recognition performance in old classes will drop significantly. This phenomenon is widely known as catastrophic forgetting. To alleviate catastrophic forgetting, existing methods store a small portion of old class data with a memory buffer and replay it while learning new classes. These methods suffer from a severe imbalance problem between old and new classes. In this paper, we discover that the imbalance problem in CIL makes it difficult to preserve the feature relation of old classes and hard to learn the feature relation between old and new classes. To mitigate the above two issues, we design a triplet contrastive preserving (TCP) loss to preserve old knowledge, and propose an asymmetric augmented contrastive learning (A2CL) method to learn new classes. Comprehensive experiments demonstrate the effectiveness of our method, which increases the average accuracies by 1.26% and 0.95% on CIFAR-100 and ImageNet. Especially under smaller memory buffer settings where the imbalance problem is more severe, our method can surpass the baselines by a large margin (up to ***3.2%****). We also show that TCP can be easily plugged into other methods and further improve their performance.*

1. Introduction

Nowadays, deep neural networks (DNNs) show remarkable performance gains in many fields such as image recognition [12, 31], object detection [27], natural language processing [34], and so on. The traditional learning paradigm assumes that the training and testing data are subject to the same distribution. However, in real-world applications, the model will continually encounter new data and may need to handle new classes of data that have never been seen during their previous training. Therefore, the goal of enabling DNNs to incrementally learn from continually accumulated data has attracted lots of research efforts.

A naive approach is to retain the full version of the old dataset and train the DNN on all the old and new data. Nevertheless, this approach is inefficient and even impractical in terms of memory constraints. However, learning on data of new classes without accessing old data causes the DNN to rapidly forget old knowledge, which is widely known as the catastrophic forgetting phenomenon [9, 28]. Lots of strategies have been proposed to alleviate catastrophic forgetting [1, 17, 21–23, 26, 32]. Among them, the replay-based methods [1, 21, 26, 32] combined have exhibited promising results and thus are widely used. These methods only keep a small portion of old training data as the memory buffer and replay the stored old data when training DNN with new data.

Although storing a small portion of old training data can alleviate catastrophic forgetting to some extent, it suffers from a severe data imbalance problem between old and new classes. The imbalance problem has a negative impact on both learning new knowledge and preserving old knowledge. For one thing, with a few old training data, it is hard for us to learn the feature relation between the old and the new classes [24]. For another, with a few old training data, preserving the feature relation of old classes is challenging because of overfitting. Existing methods propose to alleviate this issue with distillation [21, 26, 32], which trade off the ability to learn new classes. Actually, this stability-plasticity trade-off between the learning of new classes and the knowledge preserving of old classes is widely observed in CIL [6]. Achieving effective CIL requires considering these two aspects simultaneously.

In this paper, we focus on two fundamental questions for CIL: *To facilitate an effective CIL learner, (1) what types of knowledge in the old model should be preserved? and (2) how to adapt the feature space to the new data?*

(1) What to Preserve. We identify an important but overlooked issue in the distillation strategies which leads to the suboptimal trade-offs: forcing the new model to output

*Corresponding authors

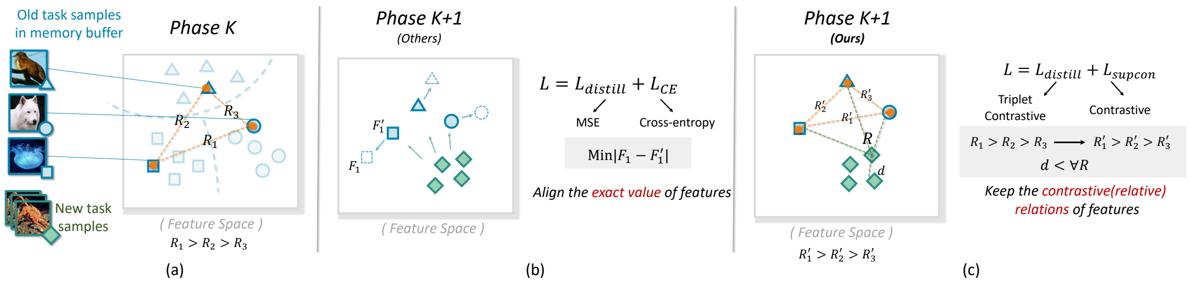

Figure 1. Here, R_1, R_2, R_3 and R_1', R_2', R_3' are the cosine similarities between features from the old model and the new model, respectively. (a) In phase K, the contrastive relationship for old data feature $R_1 > R_2 > R_3$ is critical for distinguishing old data. (b) Traditional point-wise distillation strategies penalize the exact feature positions remain unchanged. (c) Our proposed method keep the contrastive relationship of feature vectors satisfy $R_1^{'} > R_2^{'} > R_3^{'}$.

similar features *in the sense of exact value* as the old model for old data could harm the learning of new data. This is because that learning new classes can change the optimal feature space (illustrated in Figure 1 (b)), and forcing the new model to preserve the exact value of the old model logits hinders the learning of the new optimal feature space.

(2) How to Adapt. We propose to employ a contrastive-based loss [16] to learn the contrastive relationship between features of old classes and new classes. Nevertheless, directly applying contrastive learning in the CIL setting faces a severe challenge caused by the imbalance of old and new data. As shown in Figure 1 (b) and (c), since there are much fewer data points in old classes than in new classes, directly contrasting old and new data points would learn the suboptimal feature relation. To address this issue, we **introduce an asymmetric data augmentation strategy into the contrastive learning process** to learn better feature relation.

To summarize, the main contributions of this paper are:

- We discover that learning and preserving the contrastive relationship of features (i.e., the order of similarity values between features) is essential for achieving a superior stability-plasticity trade-off in CIL.
- We propose a triplet contrastive-relationship preserving (TCP) loss for distillation to preserve the contrastive relationship of old data features and thereby retain the discriminative knowledge of the old model and allow the feature space to change for better forward transfer.
- We propose an asymmetrical augmented contrastive learning (A2CL) method to alleviate the severe class imbalance problem in CIL and learn the better contrastive relationship between new and old data.
- Extensive experiments show that our method can reduce classification errors by 1.26% and 0.95% on CIFAR-100 and ImageNet. With a very small memory buffer (only ten exemplars per class are saved), our method outperforms the baseline significantly by 3.2%. We also show that TCP can be easily plugged into other methods and boost their performance.

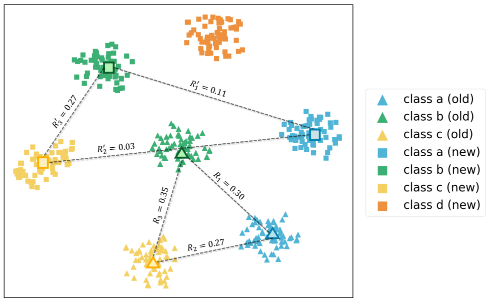

Figure 2. t-SNE visualization of the oracle feature spaces using all training data of 50 and 60 classes in CIFAR-100. We take three representative classes as the example. Blue, green, and yellow **triangles** represent the feature space of a DNN learned with all data of 50 classes; Blue, green, yellow, and orange **squares** represent the feature space of finetuning the DNN with all data of 60 classes.

2. Motivation

To verify our motivation, we train a DNN using 50 classes and show its feature space in Figure 2, where only three classes (blue, green, and yellow dots) are plotted. Here, R_i denotes the cosine similarity of two selected samples. The cosine similarities between the three samples satisfy $R_1 > R_2 > R_3$ (i.e., the contrastive relationship $R_1 - R_2 > 0$, $R_1 - R_3 > 0$, and $R_2 - R_3 > 0$). Then, we further finetune DNN with all the data from the 50 old classes and 10 new classes (i.e., the joint-training setting) and show the adapted feature space in Figure 2. We can see that in the new feature space learned from all the data, the exact positions of features (i.e., feature values) have changed. In other words, the learning of new data could result in the drift of old data features. Nevertheless, the contrastive relationship between sample features remains unchanged and still satisfies the same order $R_1' > R_2' > R_3'$ (i.e., the contrastive relationship $R_1' - R_2' > 0$, $R_1' - R_3' > 0$, and $R_2' - R_3' > 0$). In the oracle experiment, we found that more than 77% of the sample triples will keep the contrast relationship unchanged during incremental learning.

Based on the above oracle experiments, we argue that the key knowledge in the old model to preserve is not the absolute mapping from the input to output logits but rather the contrastive relationship of features. As illustrated in Figure 1 (c), in the learning phase of new classes (Phase K+1), we want the contrastive relationship of the feature similarities R'_1, R'_2 and R'_3 to remain the same as their previous relationship in Phase K: $R_1 > R_2 > R_3$. In this way, we **preserve the essential contrastive relationship between features** to retain the discriminative knowledge of the old model. In the meantime, we **allow the feature space to change**, which is desirable for better forward transfer since the learning of new data requires the adaption of the feature space (see Figure 2).

3. Related Work

3.1. Class-incremental Learning

Most methods use the logits distillation [13] to preserve old knowledge, which is introduced by LwF [19]. Then, iCaRL [26] introduces the replay-based setting and shows the potential to effectively alleviate forgetting by storing and replaying a small number of old samples. After that, a lot of studies [1, 14, 20, 32, 35] follow the distillation-based and replay-based setting to increase the performance of CIL. The aforementioned methods mostly use cross-entropy to learn new classes. Yu et al. [37], and Cha et al. [2] use the triplet loss and contrastive loss to learn more representative feature space for new classes. In this paper, unlike the previous distillation-based work, we aim to preserve the contrastive relationship instead of logits or feature values.

3.2. Knowledge distillation

Existing knowledge distillation (KD) methods mainly train a student DNN to mimic the logits or feature position of the old model [13] and are widely used in class-incremental learning. In order to transfer the structured information to the student, Tian et al. [33] proposes contrastive representation distillation (CRD) to learn a new student representation. Specifically, CRD pulls the representations from the teacher and student closer with the same input sample; otherwise, it pushes the representations apart. Our work focuses on keeping the learned contrastive representation unchanged instead of learning a new representation. Park et al. [25] and Gao et al. [25] propose relational knowledge distillation (RKD) to penalize the change of exact angle values in each sample triplet. Unlike learning on fixed datasets, CIL aims to learn new classes that can change the optimal feature space of old classes. Although distilling the exact values of old data, such as feature position and angle values of features, can alleviate forgetting, it can also hinder learning new data. Therefore, instead of keeping some exact values unchanged, we aim to preserve the contrastive relationship in each triplet (i.e., the order of similarity values between features).

3.3. Supervised Contrastive Learning

Self-supervised contrastive learning (SSCL) [3–5, 11, 18] is widely used to learn the representation from unlabeled images. Some studies [8, 10] show that contrastive learning shows a high potential to increase the plasticity in unsupervised continual learning. Recently, Khosla et.al. [16] extended the SSCL to a fully-supervised setting by leveraging the label information and introducing the supervised contrast learning. They propose supervised contrastive learning (SCL) to pull each sample pair from the same class closer and push each sample pair from the different classes away. Meanwhile, Khosla et.al. [16] also show the connection between supervised contrastive loss and triplet loss, which is widely used in deep metric learning [15, 29]. In short, the triplet loss and the contrastive loss share the same idea of pulling the features from the same class closer and pushing the features from different classes away.

In our work, we introduce the idea of contrastive learning in two aspects: (1) To preserve old knowledge, we follow the idea of triplet loss [29] and design a Triplet Contrastive Preserving loss term to maintain the contrastive relationship of old sample features. Different from SDC [37], we introduce the triplet loss as the distillation term to preserve old knowledge rather than learn new classes. (2) When adapting to new classes, we use the supervised contrastive loss to model the contrastive relationship between new classes and old classes.

4. Background

4.1. Replay-based class-incremental Learning

Here we introduce the general setup of class-incremental learning. Let $\mathcal{D}$, X, and Y denote the training dataset, training images, and training labels. In CIL, we aim to learn a DNN on a sequence of tasks $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_N\}$, where $(X_i, Y_i) \in \mathcal{D}_i$ is the sub-dataset of phase k. In phase k, the neural network can access the full $\mathcal{D}_k$, and the previous sub-datasets $\{\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_{(k-1)}\}$ are no longer fully available. In this paper, we follow the popular replay-based class-incremental learning settings to store a small part of exemplars of the previous sub-dataset in a memory buffer $\mathcal{M}$. Then, in phase k, we need to learn a DNN model by accessing dataset $\mathcal{D}_k^{(*)} = \mathcal{D}_k \cup \mathcal{M}$ and the model learned in phase k-1 (usually referred to as the old model). And the data in $\mathcal{M}$ is usually used for distillation to preserve the knowledge of the old model.

The mainstream class incremental learning framework is shown in Figure 3 (b). For learning new data (i.e., forward transfer), the mainstream framework minimizes cross-entropy loss on new training data. To alleviate catastrophic forgetting (i.e., negative backward transfer), they use the old model and the new model to extract the features of the memory buffer data, denoted as $\{m_i\}_{i=1,\cdots}$ and $\{m_i^{'}\}_{i=1,\cdots}$,

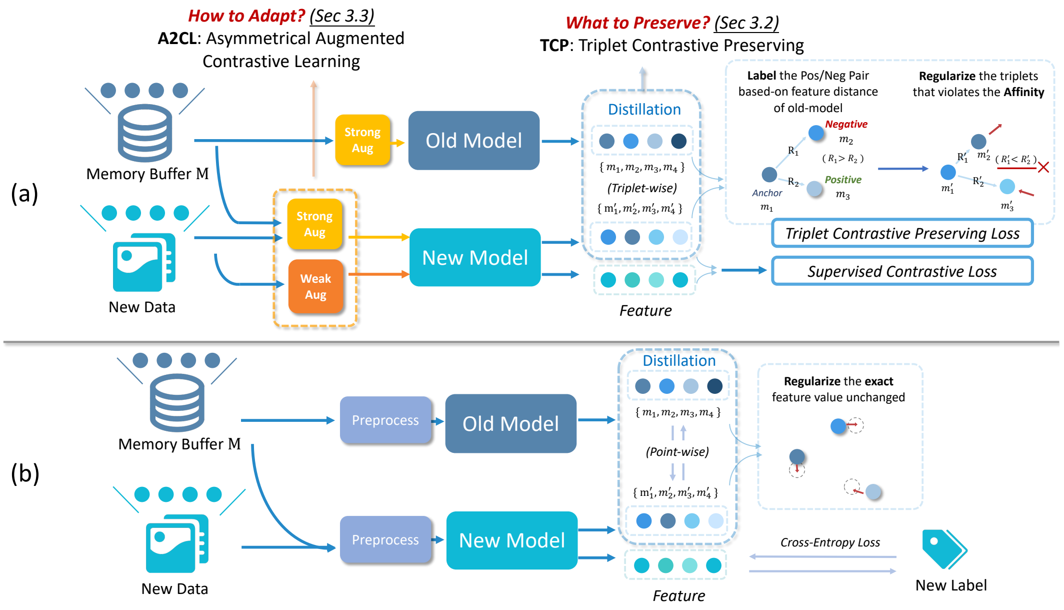

Figure 3. (a) Our framework. We apply the triplet contrastive-relationship preserving (TCP) loss to preserve the contrastive relationship among old exemplars. Then, we employ supervised contrastive loss with a new asymmetric augmentation strategy to optimize the contrastive relationship between new and old classes. (b) Traditional framework. They penalize the feature positions to remain unchanged to alleviate forgetting and use the cross-entropy loss to learn new data.

respectively. Then, they employ a distillation loss that encourages m_i and $m_i^{'}$ to be similar for each data point i in $\mathcal{M}$ in a point-wise manner.

To summarize, the objective of existing mainstream CIL method in each incremental phase is:

$$\mathcal{L} = \mathcal{L}_{\text{CE}} + \lambda \mathcal{L}_{\text{distill}}, \tag{1}$$

where $\mathcal{L}_{\text{CE}}$ and $\mathcal{L}_{\text{distill}}$ denote the cross-entropy loss and point-wise distillation loss.

4.2. Supervised Contrastive Learning

In supervised contrastive learning (SCL) [16], Given a batch of N training samples $\{(x_i, y_i)\}_{i=0}^{N}$, we generate two copies for one sample by applying two random augmentations to form an augmented batch $\{(\hat{x_i}, \hat{y_i})\}_{i=0}^{2N}$, where $x_{2\hat{k}-1}$ and $x_{2\hat{k}}$ denote two augmented views of a sample x_k and $y_k = y_{2\hat{k}} = y_{2\hat{k}-1}$. Denote h as the feature extractor and g as a nonlinear projection head [3], we use the following formula to map the augmented sample batch into a normalized feature space:

$$f_i = g\left(h\left(\hat{x_i}\right)\right). \tag{2}$$

Then we train the feature extractor by minimizing the following supervised contrastive loss:

$$\mathcal{L}_{\text{supcon}} = \sum_{i=1}^{2N} \frac{-1}{|\mathcal{P}_i|} \sum_{j\in\mathcal{P}_i} \left(\frac{exp\left(f_i \cdot f_j/\tau\right)}{\sum_{k\neq i} exp\left(f_i \cdot f_k/\tau\right)} \right), \tag{3}$$

where $\tau > 0$ is a hyperparameter that stands for the temperature and $\mathcal{P}_i$ is the index set of positive samples with respect to the anchor sample $\hat{x_i}$, defined as

$$\mathcal{P}_i = \{j \in \{1, \dots, 2N\} | j \neq i, y_i = y_j\}. \tag{4}$$

The sample in $\mathcal{P}_i$ is either x_i passing another augmentation or one of the other augmented samples with the same label as x_i.

5. Method

5.1. Framework Overview

Figure 3 (a) demonstrates how our method trains the new model in one incremental phase. To alleviate forgetting, we propose to preserve the knowledge of the old model on the contrastive relationship of features by designing a triplet contrastive relationship preserving (TCP) loss. To learn the contrastive relationship of features from the new data, we apply the supervised contrastive loss (SCL). Directly applying SCL in the CIL setting faces a severe imbalance challenge between old and new data. Thus, we develop an asymmetrical augmented contrastive learning (A2CL) method to address this issue. To summarize, our objective of training the new model can be written as

$$\mathcal{L} = \mathcal{L}_{\text{A2CL}} + \alpha \mathcal{L}_{\text{TCP}}, \tag{5}$$

where $\mathcal{L}_{\text{A2CL}}$ and $\mathcal{L}_{\text{TCP}}$ denote asymmetrical augmented contrastive loss and the TCP loss.

5.2. TCP: Triplet Contrastive-relationship Preserving Loss

Traditional logits or feature distillation strategies aim to align the exact logits or feature value of the old model and

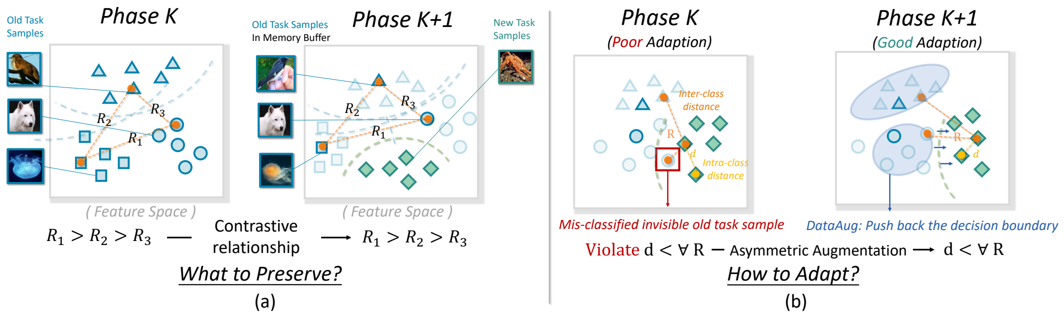

Figure 4. (a) What to preserve: We design a triplet contrastive preserving loss to preserve the contrastive relationship of features of memory buffer exemplars. (b) How to adapt: We apply SCL with asymmetrical data augmentation for learning better feature relation. Specifically, the contrastive loss is computed between a weakly augmented view of a data point and a list of strongly augmented views of other data (including memory buffer exemplars and new data).

the new model. As shown in Figure 1 (b), Strictly penalizing the changes of the old sample's feature positions obtained by the new model as feature distillation [14] does would harm the performance of CIL. It ensures the backward transfer, however, would harm the forward transfer. When incrementally learning new classes, the feature space will naturally drift as the new class's feature vectors join; forcing the old data's feature positions unchanged will hinder the learning of new classes.

In contrast, we adopt the "triplet-wise" TCP loss to preserve the sample's relative relationship learned by the old model. As shown in Figure 3, other distillation strategies focus on minimizing the similarities between feature m_i and $m_i^{'}$. In contrast, our TCP loss is conducted within sampled triplets. It enforces the consistency of the triplet's contrastive relationship between the old model $(R_1 > R_2)$ and the new model $(R_1^{'} > R_2^{'})$. It enables the feature space to flexibly adapt for new classes, which is important for better forward transfer.

To measure the contrastive relationship, we consider a triplet $\{m_i, m_j, m_k\}$ in the memory buffer $\mathcal{M}$. We pass it through the old and the new model to get two normalized feature triplets $\{f_i, f_j, f_k\}$ and $\{f_i^{'}, f_j^{'}, f_k^{'}\}$. We set m_i as the anchor sample and calculate the cosine similarity between the anchor and two other samples.

$$R_{i,j} = f_i \cdot f_j, R_{i,k} = f_i \cdot f_k, \tag{6}$$

$$R_{i,j}^{'} = f_i^{'} \cdot f_j^{'}, R_{i,k}^{'} = f_i^{'} \cdot f_k^{'}. \tag{7}$$

Without the loss of generality, we choose m_j as the positive sample and m_k as the negative one. The contrastive relationship of the triplet is reflected through the difference in the feature similarity between the anchor and positive or negative sample $D_{i,jk}$ and $D_{i,jk}^{'}$. We aim to ensure their consistency.

$$D_{i,jk} = R_{i,j} - R_{i,k}, \tag{8}$$

$$D_{i,jk}^{'} = R_{i,j}^{'} - R_{i,k}^{'}. \tag{9}$$

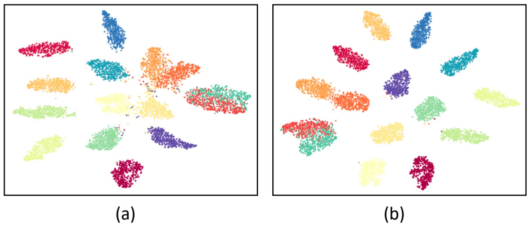

Figure 5. Feature embedding of ResNet-32 learned using contrastive learning with two different augmentation strategies. (a) Apply an uniformly strong augmentation on both two views in SCL; (b) Apply the asymmetrical augmentation (our method), i.e., one weak and one strong augmentation on the two views.

Following the idea of triplet loss, treating the old model's relative feature similarity $D_{i,jk}$ as a reference, our TCP loss enforces the new model's feature similarities of anchor-positive pair to be smaller than that of anchor-negative pair by a margin of σ. Specifically, for $D_{i,jk} > 0$, the $D_{i,jk}^{'}$ is encouraged to meet $D_{i,jk}^{'} > 0$ and $D_{i,jk}^{'} > \sigma$. In light of the above, for each triplet satisfies $D_{i,jk} > 0$, we formulate the triplet contrastive-relationship preserving (TCP) loss as follows:

$$\mathcal{L}_{\text{triplet}} = \max\{\sigma - D_{i,jk}^{'}, 0\}, \tag{10}$$

where σ denotes the margin in the triplet loss. Then for each triplet, we minimize the mean value of the triplet contrastive-relationship preserving loss function. The final loss function is:

$$\mathcal{L}_{\text{TCP}} = \frac{1}{N_{\text{tri}}} \sum_{D_{i,jk}>0} \max\{\sigma - D_{i,jk}^{'}, 0\}, \tag{11}$$

where N_{tri} denotes the total number of triplets in the memory buffer $\mathcal{M}$ which satisfy the $D_{i,jk} > 0$ constrain. We choose the relative feature similarities of the old model $D_{i,jk}$ as the margin σ. Note that we select all the triplets that satisfy $D_{i,jk} > 0$ without any hard sample mining strategy; the training speed is only 5% slower than point-wise distillation loss on RTX 3090. With the same batch size, the additional computation overhead of $\mathcal{L}_{\text{TCP}}$ is negligible.

5.3. A2CL: Asymmetrical Augmented Contrastive Learning

In order to adapt the feature space to learn new classes, instead of using the cross-entropy loss, we employ the supervised contrastive loss to learn the representations from the perspective of contrastive relationship optimization.

However, directly applying the supervised contrastive loss in the CIL setting faces a severe challenge caused by the imbalance of old and new data as we can only access a small part of old data in the memory buffer $\mathcal{M}$. As illustrated in Figure 4 (b)-left, if we directly use the old samples in the memory buffer, we would learn a suboptimal feature representation with the biased feature relation.

In order to alleviate the imbalance issue, the straight idea is to apply different data augmentation strategies on old and new data in SCL instead of using a simple augmentation for each exemplar.

For the old data with only a few exemplars, we use a uniformly strong data augmentation strategy for the two views of each old exemplar. In this way, the model can see more complex variations of the old data and avoid overfitting the old data. As shown in Figure 3 (a), we feed old exemplars in the memory buffer $\mathcal{M}$ into the strong augmentation module and then push its output feature away from the feature of the new data.

For the new data with a large number of exemplars, we empirically find that adopting a uniformly strong data augmentation causes the DNN to fail to learn well-separated features for new classes, as shown in Figure 5. In addition, we cannot apply a uniformly weak augmentation strategy to both views of each new exemplar, either. We argue that this is necessary since the "new data" in phase k would become the "old exemplars" in phase k+1. If the model has not seen strongly augmented views of the new data in phase k, it is difficult for the model to correctly handle their strongly augmented views in phase k+1. Therefore, we apply an **asymmetric augmentation strategy** to generate a weakly augmented view and a strongly augmented view for each new exemplar.

Specifically, in each training iteration, we sample N new data and M old exemplars, and conduct the augmentation as follows:

$$\begin{aligned}&\hat{x}_{2k-1} = \mathcal{A}_{\text{strong}}(x_k); \hat{x}_{2k} = \mathcal{A}_{\text{weak}}(x_k),\\&\hat{m}_{2k-1} = \mathcal{A}_{\text{strong}}(m_k); \hat{m}_{2k} = \mathcal{A}_{\text{strong}}(m_k),\end{aligned} \qquad (12)$$

where $\{x_k\}_{i=1,\cdots,N}$ and $\{m_k\}_{i=1,\cdots,M}$ denote the new and old exemplars, $\{\hat{x}_i\}_{i=1,\cdots,2N}$ and $\{\hat{m}_i\}_{i=1,\cdots,2M}$ denote the augmented new and old exemplars, $\mathcal{A}_{\text{strong}}$ and $\mathcal{A}_{\text{weak}}$ denote the augmentation function. Then, we concatenate $\{\hat{x}_i\}_{i=1,\cdots,2N}$ and $\{\hat{m}_i\}_{i=1,\cdots,2M}$ to get the overall batch of augmented data. Finally, we feed the asymmetrical augmented data batch into the new model and use Equation 3 to compute the $\mathcal{L}_{\text{A2CL}}$.

5.4. Comparison of the TCP Loss and Other Distillation Losses

Previous distillation losses in CIL methods can be classified into two types: point-wise ones and pair-wise ones.

Point-wise Distillation The distillation losses used in most CIL methods [1,14,19,26,33,35] are point-wise distillation losses. They penalize the change of feature position (i.e., feature values) for each data point. Denoting the feature for data i extracted from the new model and the old model as m_i and $m_i^{'}$, a general form of the point-wise distillation loss is

$$\mathcal{L}_{\text{point-wise}} = \sum_i \mathcal{L}(m_i, m_i^{'}). \qquad (13)$$

Pair-wise Distillation Instead of penalizing changes in the feature positions, the pair-wise distillation losses in Co^{2}L [2] and TPCIL [32] penalize the changes of feature similarities between each pair of data. A general form of the pair-wise distillation loss is

$$\mathcal{L}_{\text{pair-wise}} = \sum_{i \neq j} \mathcal{L}(R(m_i, m_j), R(m_i^{'}, m_j^{'})), \qquad (14)$$

where R denotes a similarity measure (e.g., the cosine similarity measure) between two feature vectors.

Pair-wise distillation losses are more suitable for CIL than point-wise losses as they allow some flexibility in the changes of feature positions. Nevertheless, as shown in Figure 2, the learning of new classes changes the similarities ($R_1 \neq R_1'$, $R_2 \neq R_2'$, $R_3 \neq R_3'$) between data pairs. That is to say, penalizing any changes in pair-wise similarities might still be too constrained for the CIL setting.

Our Triplet-wise Distillation In contrast to the point-wise and pair-wise distillation losses, our proposed TCP loss (Equation 11) is a triplet-wise distillation loss. It only penalizes the change of *relative* feature similarity between data triplets instead of any exact values. Compared with previous distillation losses, TCP allows more flexibility in the changes of the feature space, thus enabling our method to achieve a superior trade-off between preserving old knowledge and learning new knowledge.

6. Experiments

In this section, we first report the average accuracy of the proposed method on CIFAR-100 and ImageNet under different settings and compare it with representative CIL methods. Then, we provide an ablation study to show the effect of the proposed Triplet Contrastive Preserving (TCP), and Asymmetrical Augmented Contrastive Learning (A2CL).

Method	CIFAR-100			ImageNet-100		ImageNet-1000	
	T=1	5	10	5	10	5	10
LwF [19]	64.42	49.78	47.51	53.61	47.98	45.81	41.47
EEiL [1]	47.78	54.10	52.83	57.06	52.60	48.67	44.20
iCaRL [26]	68.08	57.03	52.96	64.79	59.42	53.50	48.73
LUCIR [14]	68.27	63.46	59.93	70.47	68.09	64.18	61.34
AAN [20]	-	66.37	64.86	72.55	69.22	64.69	62.39
Mnemonics [22]	-	63.34	62.28	72.58	71.37	64.54	63.01
PODNet [7]	-	64.07	62.18	72.01	70.57	64.95	62.24
TPCIL [32]	68.72	65.28	62.62	72.53	70.02	64.89	62.88
Ours	**69.23**	**66.54**	**64.11**	**73.48**	**71.14**	**65.76**	**63.78**
GeoDL [30]	-	65.14	65.03	73.87	73.55	-	-
DER [36]	-	67.60	66.36	76.26	74.81	-	-
RMM [21]	-	68.36	66.67	79.50	78.11	-	-
Ours+RMM	**-**	**69.23**	**67.98**	**79.87**	**79.19**	**-**	**-**

Table 1. Average accuracy compared with other methods on CIFAR-100 and imageNet

Methods	Encountered Classes						Average Acc
	50	60	70	80	90	100	
Weak Aug + $\mathrm{TCP}_{\mathrm{both}}$	79.5	70.26	65.08	58.77	55.97	51.27	63.48
Strong Aug + $\mathrm{TCP}_{\mathrm{both}}$	79.5	72.41	66.41	61.51	57.84	53.55	65.20
A2CL + $\mathrm{TCP}_{\mathrm{both}}$ (ours)	79.5	73.45	67.75	62.96	59.57	56.02	66.54

Table 2. Average Accuracy of different augmentation strategies

6.1. Main Results

In this section, we report extensive results to show that the proposed method can outperform other distillation-based baselines on CIFAR100 and ImageNet datasets under different settings. Then, we plug the proposed methods into other methods to show the performance improvements.

6.1.1 CIFAR-100

We run our experiments under 1-, 5-, and 10-phase settings with 50, 10, and 5 classes per incremental phase. As shown in Table 1, for average accuracy, our method achieves the average accuracy of 69.23%, 66.54%, and 64.11% under the 1-, 5- and 10-phase settings and outperforms the recent pair-wise distillation strategy TPCIL by up to 0.51%. 1.26% and 1.49%, respectively. Compared with the recent point-wise distillation strategy LUCIR, we achieved a 0.96%, 3.08%, and 4.18% improvement in average accuracy under 1-, 5- and 10-phase settings. In addition, **as our method can be easily plugged into other methods, we show the performance of the proposed method plugged into RMM [21].** As shown in Table 1, Ours+RMM can outperform the original RMM by 0.87% and 1.31% under the 5- and 10-phase settings. We can also surpass the model expansion method DER [36], which use much larger models, by 1.63% and 1.62% under the 5- and 10-phase settings, respectively.

6.1.2 ImageNet-100

We run the experiments under 5, and 10 phases with respectively 10, and 5 classes per incremental phase on ImageNet-100 dataset. Our method achieves the average accuracy of 73.48% and 71.14% under the 5- and 10-phase settings. Compared with the pair-wise distillation strategy TPCIL, our method can improve the performance by up to 0.95% and 1.12% under 5- and 10-phase settings. For point-wise distillation strategies, our results outperform the LUCIR by up to 3.01% and 3.05% under 5- and 10-phase settings, respectively. In addition, combined with RMM [21], we can also achieve more than 0.37% and 1.08% improvement under 5- and 10-phase settings.

6.1.3 ImageNet-1000

We provide the comparisons under 5 and 10 phases with respectively 10, and 5 classes per incremental phase on ImageNet-1000 dataset. We achieve the average accuracy of 65.37% and 63.11% under the 5- and 10-phase settings. Compared with pair-wise TPCIL, our method can surpass them by up to 0.87% and 0.90% under 5- and 10-phase settings, respectively. For point-wise distillation strategies, our results outperform LUCIR by up to 1.19% and 1.92% under 5- and 10-phase settings, respectively.

Method	Old Acc.	New Acc.	Overall Acc.
A2CL+FDL [14]	72.56	53.78	69.43
A2CL+IRD [2]	73.06	58.66	70.66
A2CL+TCP$_{\mathrm{rep}}$	73.01	67.61	**72.11**
A2CL+TCP$_{\mathrm{con}}$	73.43	65.09	**72.04**
A2CL+TCP$_{\mathrm{both}}$	73.72	72.10	**73.45**

Table 3. Average Accuracy of old and new classes after the first phase of CIFAR100 under 5-phase setting.

6.2. Ablation Study

In this section, we will present the ablation studies of our proposed techniques to show their effectiveness. We evaluate different techniques under the 5-phase setting on CIFAR-100 dataset. More detailed analysis about our method could be found in the supplementary.

6.2.1 Comparison of distillation losses

While keeping a similar accuracy on old classes, our TCP loss can significantly improve new data performance. As shown in Table 3, we report the average accuracy of old and new data after the first incremental phase on CIFAR-100. Specifically, the old data contain 50 classes, and the new data contain ten classes. With a similarly high accuracy on old classes, we compare the accuracy on new classes. For baselines, we show the performance of a point-wise Feature Distillation Loss (FDL) [14] and a pair-wise Instance-wise Relation Distillation (IRD) [2]. We apply our TCP loss in two feature spaces: the representation space before the nonlinear projection head and the contrastive space after the nonlinear projection head. In Table 3, TCP$_{\mathrm{rep}}$, TCP$_{\mathrm{con}}$ and TCP$_{\mathrm{both}}$ denote that we apply the TCP loss in representation space, contrastive learning space, and both. The experiment results show that the new data accuracy of pair-wise IRD can surpass the point-wise FDL by up to 4.88%. Using TCP$_{\mathrm{rep}}$, TCP$_{\mathrm{con}}$ could already significantly improve new class accuracy by up to 6.43% compared with IRD. Then, applying the TCP loss on both spaces, new data accuracy reaches 72.10%, which outperforms the pair-wise IRD by 13.44%. TCP loss achieves a better trade-off between stability and plasticity in CIL.

In addition, **the additional execution time overhead of TCP is negligible**. Measuring the execution time of one batch (256) on RTX 2080Ti, FDL, IRD, and TCP take 5.335, 5.474s, and 5.624s. TCP is 5.08% and 2.67% slower than FDP and IRD, which is acceptable.

6.2.2 Comparison of augmentation strategies

We show that applying an asymmetric data augmentation strategy is useful to alleviate the imbalance between old classes and new classes. In Table 2, we apply a complex data augmentation on the memory buffer $\mathcal{M}$ and report the effect of different data augmentation strategies on new data. Here, the "Weak Aug" and the "Strong Aug" denote using a uniformly weak or a uniformly strong augmentation for both copies of the old data sample. "A2CL" denotes our method that uses a weak and a strong augmentation for the two copies of each new exemplar. As shown in the table, our A2CL scheme outperforms both the strong and weak data augmentation strategies by 3.1% and 1.3%.

Method	the number of exemplars per class				
	10	20	30	40	50
iCaRL	52.5	56.5	60.0	61.0	62.0
EEIL	41.8	50.3	55.2	57.1	59.7
LUCIR	61.0	64.0	64.5	65.5	66.0
TPCIL	61.5	65.3	66.2	66.5	67.0
Ours	**64.7**	**66.5**	**67.0**	**67.2**	**67.7**

Table 4. Average Accuracy when using different number of exemplars per old class

6.2.3 The effect of exemplar number

memory buffer $\mathcal{M}$ is often used in CIL methods to store the old class exemplars. Although storing more representative exemplars is helpful for performance, it brings a larger memory overhead. Table 4 shows the average accuracy of different methods when using different numbers of exemplars per class. **We find that our method significantly outperforms other methods when using a small memory buffer (e.g., less than ten exemplars per old class).** Specifically, we achieve an average accuracy of 64.7%, surpassing TPCIL by 3.2% (i.e., 64.7 v.s. 61.5). Our method is superior for extremely storage-constrained scenarios.

7. Conclusion

In this paper, we discover that learning and preserving the contrastive relationship is essential in class-incremental learning (CIL). Based on this core idea, we propose two techniques, TCP and A2CL, to preserve and adapt the contrastive relationship in the feature space in CIL scenarios. Extensive experiments demonstrate that these techniques successfully strike a better balance between stability and plasticity and can be easily plugged into other methods to boost their performance.

Acknowledgement

This work was supported by the National Natural Science Foundation of China (No. U19B2019, 61832007, 62104128, U21B2031), Tsinghua EE Xilinx AI Research Fund, Tsinghua-Meituan Joint Institute for Digital Life, and Beijing National Research Center for Information Science and Technology (BNRist).

References

[1] Francisco M Castro, Manuel J Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In *Proceedings of the European conference on computer vision (ECCV)*, pages 233–248, 2018.

[2] Hyuntak Cha, Jaeho Lee, and Jinwoo Shin. Co2l: Contrastive continual learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9516–9525, 2021.

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[4] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. *Advances in neural information processing systems*, 33:22243–22255, 2020.

[5] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[6] Matthias Delange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Greg Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[7] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16*, pages 86–102. Springer, 2020.

[8] Enrico Fini, Victor G Turrisi Da Costa, Xavier Alameda-Pineda, Elisa Ricci, Karteek Alahari, and Julien Mairal. Self-supervised models are continual learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9621–9630, 2022.

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

[10] Alex Gomez-Villa, Bartlomiej Twardowski, Lu Yu, Andrew D Bagdanov, and Joost van de Weijer. Continually learning self-supervised representations with projected functional regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3867–3877, 2022.

[11] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[13] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015.

[14] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 831–839, 2019.

[15] Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. Learning deep representation for imbalanced classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5375–5384, 2016.

[16] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33:18661–18673, 2020.

[17] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

[18] Junnan Li, Pan Zhou, Caiming Xiong, and Steven CH Hoi. Prototypical contrastive learning of unsupervised representations. *arXiv preprint arXiv:2005.04966*, 2020.

[19] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

[20] Yaoyao Liu, Bernt Schiele, and Qianru Sun. Adaptive aggregation networks for class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2544–2553, 2021.

[21] Yaoyao Liu, Bernt Schiele, and Qianru Sun. Rmm: Reinforced memory management for class-incremental learning. *Advances in Neural Information Processing Systems*, 34:3478–3490, 2021.

[22] Yaoyao Liu, Yuting Su, An-An Liu, Bernt Schiele, and Qianru Sun. Mnemonics training: Multi-class incremental learning without forgetting. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 12245–12254, 2020.

[23] Arun Mallya, Dillon Davis, and Svetlana Lazebnik. Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 67–82, 2018.

[24] Zixuan Ni, Haizhou Shi, Siliang Tang, Longhui Wei, Qi Tian, and Yueting Zhuang. Revisiting catastrophic forgetting in class incremental learning. *arXiv preprint arXiv:2107.12308*, 2021.

[25] Wonpyo Park and et al. Relational knowledge distillation. In *CVPR*, pages 3967–3976, 2019.

[26] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017.

[27] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[28] Anthony Robins. Catastrophic forgetting, rehearsal and pseudorehearsal. *Connection Science*, 7(2):123–146, 1995.

[29] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[30] Christian Simon, Piotr Koniusz, and Mehrtash Harandi. On learning the geodesic path for incremental learning. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 1591–1600, 2021.

[31] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[32] Xiaoyu Tao, Xinyuan Chang, Xiaopeng Hong, Xing Wei, and Yihong Gong. Topology-preserving class-incremental learning. In *European Conference on Computer Vision*, pages 254–270. Springer, 2020.

[33] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. *arXiv preprint arXiv:1910.10699*, 2019.

[34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[35] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 374–382, 2019.

[36] Shipeng Yan, Jiangwei Xie, and Xuming He. Der: Dynamically expandable representation for class incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3014–3023, 2021.

[37] Lu Yu, Bartlomiej Twardowski, Xialei Liu, Luis Herranz, Kai Wang, Yongmei Cheng, Shangling Jui, and Joost van de Weijer. Semantic drift compensation for class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6982–6991, 2020.