Co^{2}L: Contrastive Continual Learning

Hyuntak Cha Jaeho Lee Jinwoo Shin
KAIST
Daejeon, South Korea
{hyuntak.cha, jaeho-lee, jinwoos}@kaist.ac.kr

Abstract

Recent breakthroughs in self-supervised learning show that such algorithms learn visual representations that can be transferred better to unseen tasks than cross-entropy based methods which rely on task-specific supervision. In this paper, we found that the similar holds in the continual learning context: contrastively learned representations are more robust against the catastrophic forgetting than ones trained with the cross-entropy objective. Based on this novel observation, we propose a rehearsal-based continual learning algorithm that focuses on continually learning and maintaining transferable representations. More specifically, the proposed scheme (1) learns representations using the contrastive learning objective, and (2) preserves learned representations using a self-supervised distillation step. We conduct extensive experimental validations under popular benchmark image classification datasets, where our method sets the new state-of-the-art performance. Source code is available at https://github.com/chaht01/Co2L.

1. Introduction

Modern deep learning algorithms show impressive performances on the task at hand, but it is well known that they often struggle to retain their knowledge on previously learned tasks after being trained on a new one [32]. To mitigate such "catastrophic forgetting," prior works in the continual learning literature focus on *preserving* the previously learned knowledge using various types of information about the past task. Replay-based approaches store a small portion of past samples and rehearse the samples along with present task samples [35, 29, 34, 5]. Regularization-based approaches force the current model to be sufficiently close to the past model—which may be informative about the past task—in the parameter/functional space distance [25, 6, 39]. Expansion-based approaches allocate a unit (*e.g.*, network node, sub-network) for each task and keep the unit untouched during the training for other tasks [38, 31].

In this paper, instead of asking how to isolate previous knowledge from new knowledge, we draw attention to the following fundamental question:

What type of knowledge is likely to be useful for future tasks (and thus not get forgotten), and how can we learn and preserve such knowledge?

To demonstrate its significance, consider the simple scenario that the task at hand is to classify the given image as an apple or a banana. An easy way to solve this problem is to extract and use the color feature of the image; red means apple, and yellow means banana. The color, however, will no longer be useful if our future task is to classify another set of images as apples or strawberries; color may not be used anymore and eventually get forgotten. On the other hand, if the model had learned more complicated features, *e.g.*, shape/polish/texture, the features may be re-used for future tasks and remain unforgotten. This line of thoughts suggests that forgetting does not only come from the limited access to the past experience, but also from the innately restricted access to future events; to suffer less from forgetting, learning more *transferable representations* in the first hand may be as important as carefully preserving the knowledge gained in the past.

To learn more transferable representations for continual learning, we draw inspirations from a recent advance in self-supervised learning, in particular, *contrastive learning* [19, 10]. Contrastive methods learn representations using the inductive bias that the prediction should be invariant to certain input transformations instead of relying on task-specific supervisions. Despite their simplicity, such methods are known to be surprisingly effective; for ImageNet classification [37], contrastively trained representations closely achieve the fully-supervised performance even without labels [10] and outperform counterparts in the supervised case [24]. More importantly, while the methods are originally proposed for better in-domain[1] performance, recent works also show that such methods provide significant performance gains on unseen domains [10, 21]. Under a continual scenario, we make a similar observation: *contrastively learned representations suffer less from forgetting than the ones trained with cross-entropy loss* (see Section 5.2 for details).

[1] The term 'in-domain' is used here for the setup where data distributions for representation learning and linear classifier training are the same.

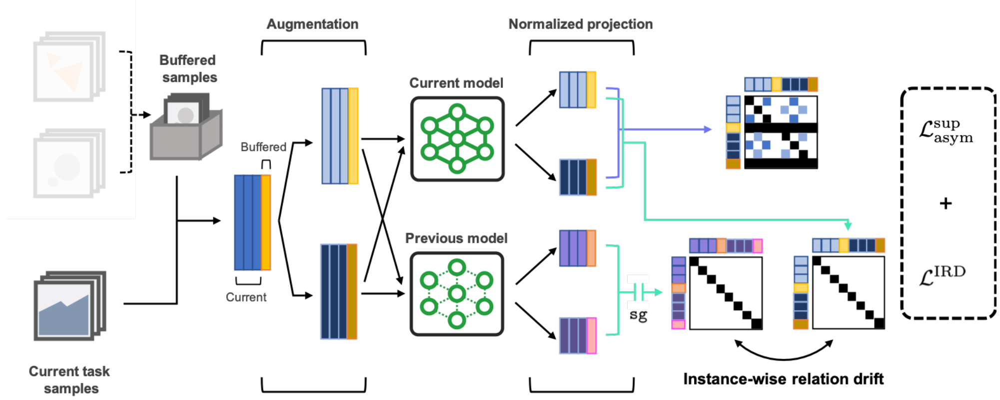


Figure 1. An overview of the Co^{2}L framework. Mini-batch samples from the current task and the memory buffer are augmented and passed through current and past (stored at the end of the previous task) representations. Co^{2}L minimizes the weighted sum of two losses: (1) Asymmetric SupCon loss contrasts anchor samples from the current task against the samples from other classes (Section 4.1). (2) IRD loss measures the drift of the instance-wise similarities given by the current model from the one given by the previous model (Section 4.2).

Unfortunately, applying this idea to continual settings is not straightforward due to at least two reasons: First, having access to informative negative samples is known to be crucial for the success of contrastive learning [36], while the instantaneous demographics of negatives samples are severely restricted under standard continual setups; in class-incremental learning, for instance, it is common to assume that the learner can access samples from only a small number of classes at each time step. Second, the question of how to preserve the contrastively learned *representations* on continual learning setups has not been fully answered. Indeed, recent works on representation learning for continual setups aim to learn representations accelerating future learning under a similar decoupled learning setup but lack an explicit design to preserve representations.

Contribution. To address these challenges, we propose a new rehearsal-based continual learning algorithm, coined **Co^{2}L** (**Co**ntrastive **Co**ntinual **L**earning). Unlike previous continual (representation) learning methods, we aim to *learn* and *preserve* representations continually in a decoupled representation-classifier scheme. The overview of Co^{2}L is illustrated in Figure 1.

Our contribution under this setup is twofold:

1. *Contrastive learning:* We design an asymmetric version of supervised contrastive loss for learning representations under continual learning setup (Section 4.1) and empirically show its benefits on improving the representation quality.
2. *Preserving representations:* We propose a novel preservation mechanism for contrastively learned representations, which works by self-distillation of instance-wise relations (Section 4.2); to the best of our knowledge, this is a first method explicitly designed to preserve contrastively learned representations on continual learning.

We validate Co^{2}L under various experimental scenarios encompassing task-incremental learning, domain-incremental learning, and class-incremental learning. Co^{2}L consistently outperforms all baselines on various datasets, scenarios, and memory setups. With careful ablation studies, we also show that both components we propose (asymmetric supervised contrastive loss, instance-wise relation distillation) are essential for performance. In the ablation of distillation, we empirically show that distillation preserves learned representations and efficiently uses buffered samples, which might be the main source of consistent gains over all comparisons: distillation provides 22.40% and 10.59% relative improvements with/without buffered samples respectively on the Seq-CIFAR-10 dataset. In the ablation of asymmetric supervised contrastive loss, we quantitatively verify that the asymmetric version consistently provides performance gains over the original one on all setups, *e.g.*, 8.15% relative improvement on the Seq-CIFAR-10 with buffer size 500. We also provide qualitative implications on this performance gain by visualizing learned representations, which shows our asymmetric version prevents severe drifts of learned features.

2. Related Work

Rehearsal-based continual learning. Continual learning methods have been developed in three major streams: using a fixed-sized buffer to replay past samples (rehearsal-based approach), regulating model parameter changes through learning (regularization-based approach), or dynamically expanding model architecture on demand (expansion-based approach). Among them, the rehearsal-based approach has shown great performance in continual learning settings, albeit its simplicity. The idea of Experience Replay (ER [34]) is simply managing a fixed-sized buffer to retain a few samples and replaying those to prevent forgetting past knowledge. Following this simple setup, ER-based methods mainly focus on either regulating model updates not to contradict the learning objectives on past samples [29, 5] or selecting the most representative/forgetting-prone samples to prevent changes in past predictions [2, 7, 33]. In a purely decoupled representation learning setup, however, there are few studies related to ER since representation learning objectives may not be directly aligned to task-specific objectives in typical training schemes. In this work, we focus on utilizing buffered samples to learn representations continually on a decoupled representation-classifier learning scheme.

Representation learning in continual learning. Only a few recent studies on continual learning focus on representations in two aspects: how to maintain learned representations [33] and how to learn representations accelerating future learning [23, 17, 43, 15]. iCaRL [33] prevents representations from being forgotten by leveraging distillation. [23, 17] directly optimize objectives that minimize forgetting by learning representations that accelerate future learning on meta-learning [14] frameworks. Concurrent works [43, 15, 30] exploit self-supervised learning objectives to learn more generalizable representations than ones trained with supervised learning objectives. In this work, we further exploit the benefits of contrastive learning scheme on continual learning setups with additional technical components designed to preserve learned representations.

Contrastive representation learning. Recent progress in contrastive representation learning shows superior downstream task performance, even competitive to supervised training. Noise-contrastive estimation [18] is the seminal work that estimates the latent distribution by contrasting with artificial noises. Info-NCE [42] tries to learn representations from visual inputs by leveraging an auto-regressive model to predict the future in an unsupervised manner. Recent advances in this area stem from the use of multiple views as positive samples [40]. These core concepts have been followed by studies [10, 21, 16, 12] that have resolved practical limitations that have previously made learning difficult such as negative sample pairs, large batch size, and momentum encoders. Meanwhile, it has been shown that supervised learning can also enjoy the benefits of contrastive representation learning by simply using labels to extend the definition of positive samples [24]. In this work, we mainly leverage contrastive representation learning schemes on the continual learning setup based on our novel observation (Section 5.2).

Knowledge distillation. In continual learning, knowledge distillation is widely used to mitigate forgetting by distilling past signatures to the current models [28, 33]. However, it has not been studied to design/utilize knowledge distillation for decoupled representation-classifier training in the continual learning setup. In this work, we develop novel self-distillation loss for contrastive continual learning, which is inspired by the recently proposed distillation loss [13] for contrastive learning framework.

3. Problem Setup and Preliminaries

In this section, we formalize the considered continual learning setup and briefly describe a recently proposed supervised contrastive learning scheme [24] that will be used as the main framework for designing Co^{2}L (Section 4).

3.1. Problem Setup: Continual Learning

We consider three popular scenarios of continual learning as categorized by [41]: task-incremental learning (Task-IL), domain-incremental learning (Domain-IL), and class-incremental learning (Class-IL).

Formally, the learner is trained on a sequence of tasks indexed by $t \in \{1, 2, \dots, T\}$. For each task, we suppose that there is a task-specific class set C_t. For Task-IL and Class-IL, $\{C_t\}_{t=1}^T$ are assumed to be disjoint, *i.e.*,

$$t \neq t' \Rightarrow C_t \cap C_{t'} = \emptyset, \qquad \text{(Task/Class-IL)}. \tag{1}$$

For Domain-IL, C_t remains the same throughout the tasks:

$$C_1 = C_2 = \cdots = C_T, \qquad \text{(Domain-IL)}. \tag{2}$$

During each task, n_t copies of training input-label pairs are independently drawn from some task-specific distribution, *i.e.*, $\{(\mathbf{x}_i, y_i)\}_{i=1}^{n_t} \sim D_t$. Here, $\mathbf{x}_i$ denotes the input image, and $y_i \in C_t$ denotes the class label belonging to the task-specific class set. For Task-IL, the learned models are assumed to have access to the task label t during the test phase; the goal is to find a predictor $\varphi_\theta(\mathbf{x}, t)$ parameterized by θ such that

$$\mathcal{L}(\theta) := \sum_{t=1}^{T} \mathbb{E}_{D_t}[\ell(y, \varphi_\theta(x, t))], \qquad \text{(Task-IL)} \tag{3}$$

is minimized for some loss function $\ell(\cdot, \cdot)$. For Domain-IL and Class-IL, the model cannot access the task label during the test phase; the goal is to find a predictor $\varphi_\theta(\mathbf{x})$ minimizing

$$\mathcal{L}(\theta) = \sum_{t=1}^{T} \mathbb{E}_{D_t}[\ell(y, \varphi_\theta(x))], \qquad \text{(Domain/Class-IL)}. \tag{4}$$

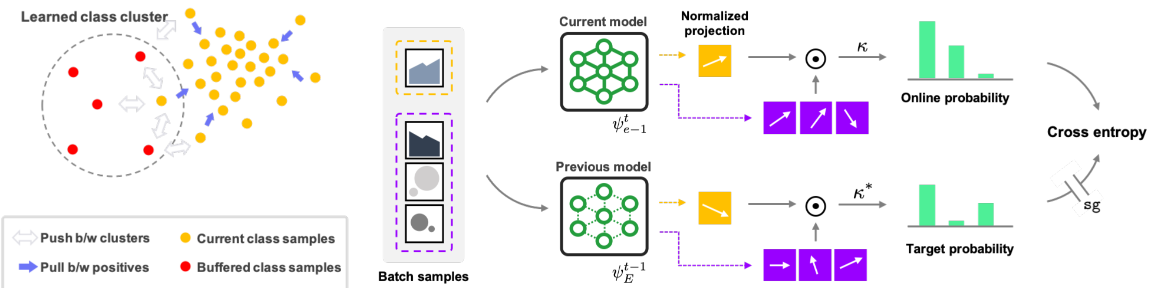
3.2. Preliminaries: Contrastive Learning

We now describe the SupCon (Supervised Contrastive learning) algorithm, proposed by [24]. Suppose that the classification model can be decomposed into two components

$$\varphi_\theta = \mathbf{w} \circ f_\vartheta \tag{5}$$

with parameter pairs $\theta = (\vartheta, \mathbf{w})$, where $\mathbf{w}(\cdot)$ is the linear classifier and $f_\vartheta(\cdot)$ is the representation. Without training $\mathbf{w}$, SupCon directly trains f_ϑ as follows: Given a batch of N training samples $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, SupCon first generates an augmented batch $\{(\tilde{\mathbf{x}}_i, \tilde{y}_i)\}_{i=1}^{2N}$ by making two randomly augmented versions of $\mathbf{x}_k$ as $\tilde{\mathbf{x}}_{2k-1}, \tilde{\mathbf{x}}_{2k}$, with $\tilde{y}_{2k-1} = \tilde{y}_{2k} = y_k$. The samples in the augmented batch are mapped to a unit d-dimensional Euclidean sphere as

$$\mathbf{z}_i = (g \circ f)_\psi(\tilde{\mathbf{x}}_i), \tag{6}$$



(a) Asymmetric SupCon Loss (b) Instance-wise Relation Distillation Loss

Figure 2. Illustration of Asymmetric Supervised Constrastive Loss and Instance-wise Relation Distillation (IRD). (a) Given augmented mini-batch samples, asymmetric SupCon considers samples from the same class of the current task as positives. In other words, the pulling effects between anchors only exist between current task samples. (b) Given augmented mini-batch samples, the instance-wise relation is defined on the normalized projected feature vectors. The relation vectors, *i.e.*, dot products ($\odot$) of feature vectors, are computed from the learnable (ψ^t_{e-1}) and reference model (ψ^{t-1}_E), respectively. For E epoch training, such temperature scaled relation is distilled from the reference model to the learnable model. Note that the reference model is snapped at the end of $(t-1)$-*th* task training, and we only update the learnable model's weights using stop-gradient (denoted by $\mathtt{sg}$).

where $g = g_\phi$ denotes the projection map parametrized by ϕ, and ψ denotes the concatenation of ϑ and ϕ. Now, the feature map $(g \circ f)_\psi$ is trained to minimize the supervised contrastive loss

$$\mathcal{L}^{\text{sup}} = \sum_{i=1}^{2N} \frac{-1}{|\mathfrak{p}_i|} \sum_{j \in \mathfrak{p}_i} \log\left(\frac{\exp(\mathbf{z}_i \cdot \mathbf{z}_j/\tau)}{\sum_{k \neq i} \exp(\mathbf{z}_i \cdot \mathbf{z}_k/\tau)}\right), \quad (7)$$

where $\tau > 0$ is some temperature hyperparameter and $\mathfrak{p}_i$ is the index set of positive samples with respect to the anchor $\tilde{\mathbf{x}}_i$, defined as

$$\mathfrak{p}_i = \big\{ j \in \{1, \ldots, 2N\} \;\big|\; j \neq i,\, y_j = y_i \big\}. \quad (8)$$

In other words, the sample in $\mathfrak{p}_i$ is either the other augmentation of the unaugmented version of $\tilde{\mathbf{x}}_i$, or one of the other augmented samples having the same label.

4. Co^{2}L: Contrastive Continual Learning

Here, we propose a rehearsal-based contrastive continual learning scheme, coined **Co^{2}L** (**Co**ntrastive **Co**ntinual **L**earning). At a high level, Co^{2}L (1) *learns* the representations with an asymmetric form of supervised contrastive loss (Section 4.1) and (2) *preserves* learned representations using self-supervised distillation (Section 4.2) in a decoupled representation-classifier training scheme. This is done by a mini-batch gradient descent based on the compound loss

$$\mathcal{L} = \underbrace{\mathcal{L}^{\text{sup}}_{\text{asym}}}_{\text{(1) learning}} + \underbrace{\lambda \cdot \mathcal{L}^{\text{IRD}}}_{\text{(2) preserving}}. \quad (9)$$

Here, each batch is composed of two independently augmented views of N samples (thus $2N$ in total), where each sample is drawn from the union of current task samples and buffered samples.

4.1. Representation Learning with Asymmetric Supervised Contrastive Loss

For continually learning representations, we use an asymmetrically modified version of the SupCon objective $\mathcal{L}^{\text{sup}}$. In the modified version, we only use current task samples as anchors; past task samples from the memory buffer will only be used as negative samples (see Figure 2(a)). Formally, if we let $S \subset \{1, \ldots, 2N\}$ be the set of indices of current task samples in the batch, the modified supervised contrastive loss is defined as

$$\mathcal{L}^{\text{sup}}_{\text{asym}} = \sum_{i \in S} \frac{-1}{|\mathfrak{p}_i|} \sum_{j \in \mathfrak{p}_i} \log\left(\frac{\exp\left(\mathbf{z}_i \cdot \mathbf{z}_j/\tau\right)}{\sum_{k \neq i} \exp\left(\mathbf{z}_i \cdot \mathbf{z}_k/\tau\right)}\right). \quad (10)$$

The motivation behind this asymmetric design is to prevent a model from overfitting to a small number of past task samples. It turns out that such a design indeed helps to boost the performance. In Section 5.3, we empirically observe that the asymmetric version $\mathcal{L}^{\text{sup}}_{\text{asym}}$ outperforms the original $\mathcal{L}^{\text{sup}}$ and generates better-spread features of buffered samples.

4.2. Instance-wise Relation Distillation (IRD) for Contrastive Continual Learning

While using the contrastive learning objective (eq. 10) readily provides a more transferable representation, one may still benefit from having an explicit mechanism to preserve the learned knowledge. Taking the inspiration from [13], we propose an instance-wise relation distillation (IRD); IRD regulates the changes in feature relation between batch samples via self-distillation (see Figure 2(b)). Formally, we define the IRD loss $\mathcal{L}^{\text{IRD}}$ as follows: For each sample $\tilde{\mathbf{x}}_i$ in a batch $\mathcal{B}$, we define the *instance-wise similarity vector*

$$\mathbf{p}\left(\tilde{\mathbf{x}}_i; \psi, \kappa\right) = \left[p_{i,1}, \ldots, p_{i,i-1}, p_{i,i+1}, \ldots, p_{i,2N}\right], \quad (11)$$

where $p_{i,j}$ denotes the normalized instance-wise similarity

$$p_{i,j} = \frac{\exp\left(\mathbf{z}_i \cdot \mathbf{z}_j/\kappa\right)}{\sum_{k\neq i}^{2N}\exp\left(\mathbf{z}_i \cdot \mathbf{z}_k/\kappa\right)} \tag{12}$$

given the representation parameterized by ψ and the temperature hyperparameter κ. In other words, the instance-wise similarity vector $\mathbf{p}(\cdot)$ is the normalized similarity of a sample to other samples in the batch.

Roughly, the IRD loss quantifies the discrepancy between the instance-wise similarities of the current representation and the past representation; the past representation is a snapshot of the model at the end of the previous task. Denoting the parameters of the past/current model as ψ^{past} and ψ, the IRD loss is defined as

$$\mathcal{L}^{\text{IRD}} = \sum_{i=1}^{2N} -\mathbf{p}\left(\tilde{\mathbf{x}}_i; \psi^{\text{past}}, \kappa^*\right) \cdot \log \mathbf{p}\left(\tilde{\mathbf{x}}_i; \psi, \kappa\right), \tag{13}$$

where the logarithms and multiplications on the vectors denote the entrywise logarithms and multiplications. We note that we are using different temperature hyperparameters for the past and current similarity vectors; on the other hand, both κ, κ^* will remain fixed throughout the tasks.

By using fixed model weights snapped at the end of previous task training as the reference model ψ^{past}, IRD distills learned representations to the current training model ψ, thereby leading to preserving learned representations. Since contrastive representation learning stems from deep metric learning, IRD achieves knowledge preservation by regulating overall *structure* changes of learned representations. Note that IRD does not regulate exact changes in feature space and does not define relation from encoder outputs like [13]. More detailed comparisons between [13] and ours are provided in the supplementary material.

4.3. Algorithm Details

Here, we give a complete picture of the overall training procedure and give additional details. The full algorithm is provided in Algorithm 1.

Data preparation. As the initial or new task arrives, the dataset is built as a union of current task samples and buffered samples, without any oversampling [9, 20]. The mini-batch is drawn from this dataset, where each sample is independently drawn with equal probability. To enjoy the benefits of contrastive representation learning, each sample is augmented into two views following [11]. The detailed augmentation scheme for contrastive learning is provided in the supplementary material.

Learning new representation. The augmented samples are forwarded to the encoder f_ϑ and projection map g_ϕ sequentially. The projection map outputs are used to compute asymmetric supervised contrastive loss (eq. 10).

Algorithm 1 Co^{2}L: Contrastive Continual Learning

Input: Buffer memory $\mathcal{M}$, Encoder parameters ϑ, projector parameters ϕ, number of tasks T, family of augmentations $\mathcal{H}$, a set of training sets $\{\{(x_i^t, y_i^t)\}\}_{t=1}^T$, a set of disjoint class sets $\{\mathcal{C}_t\}_{t=1}^T$, learning rate η, number of epochs of t-th task E_t, distillation temperatures κ, κ^*, distillation power λ.
Initialize network $(g \circ f)_\psi(\cdot)$ where $\psi = (\vartheta, \phi)$.
for $t = 1, \cdots, T$ **do**
    Construct dataset $\mathcal{D}_t$ by $\mathcal{D}_t \leftarrow \{(x_i^t, y_i^t)\} \cup \mathcal{M}$
    for $e = 1, \cdots, E_t$ **do**
        Draw a mini-batch $\{(x_i, y_i)\}_{i=1}^N$ from $\mathcal{D}_t$
        for all $k \in \{1, \cdots, N\}$ **do**
            Draw two augmentations $h \sim \mathcal{H}, h' \sim \mathcal{H}$
            Initialize anchor indices sets $S \leftarrow \emptyset, I \leftarrow \emptyset$
            $\tilde{x}_{2k-1} = h(x_k)$
            $\tilde{x}_{2k} = h'(x_k)$
            $I \leftarrow I \cup \{2k-1, 2k\}$
            if $y_k \in C_t$ **then**
                $S \leftarrow S \cup \{2k-1, 2k\}$
            end if
        end for
        Compute $\mathcal{L}$ by $\mathcal{L} \leftarrow \mathcal{L}_{\text{asym}}^{\text{sup}}(I, S; \psi_{e-1}^t)$ (eq. 10)
        if $t > 1$ **then**
            Update $\mathcal{L}$ by
            $\mathcal{L} \leftarrow \mathcal{L} + \lambda \cdot \mathcal{L}^{\text{IRD}}(\psi_{E_{t-1}}^{t-1}, \psi_{e-1}^t, \kappa^*, \kappa)$ (eq. 13)
        end if
        Update ψ_{e-1}^t by $\psi_e^t \leftarrow \psi_{e-1}^t - \eta \nabla_{\psi_{e-1}^t} \mathcal{L}$
    end for
    Manage buffer $\mathcal{M}$ for the number of each class samples to be same by uniform sampling.
end for

Preserving learned representation. When a new task arrives (*i.e.*, $t > 1$), we compute instance-wise relation drifts between reference model and the training model with IRD loss (eq. 13). To this end, we settle the reference model as the trained model at the end of the training of $(t-1)$-th task. Note that while optimizing total loss (eq. 9), the reference model is not updated.

Buffer management. At the end of training each task, a small portion of training samples is pushed into a replay buffer. Due to its buffer size constraint, a small subset of samples from each class is pulled out of the replay buffer at the same ratio. The sample to be pushed or pulled is uniformly randomly selected for all procedures.

5. Experiment

5.1. Experimental Setup

Learning scenarios and datasets. Following [41], we conduct continual learning experiments on Task Incremental

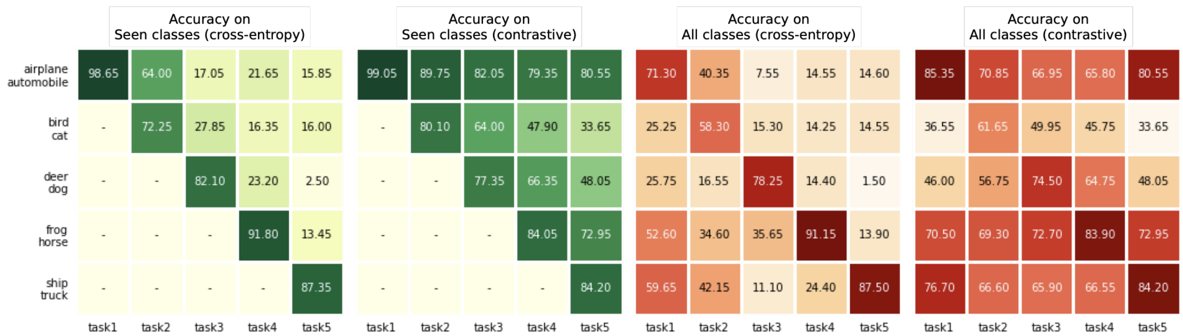


Figure 3. Observation on two learning schemes, cross-entropy loss training and contrastive representation learning on Seq-CIFAR-10 without any design used for the continual learning settings. As new task arrives, each model is trained only with current task samples with model weights without re-initialization. After each task training ends, a new linear classifier is trained on the fixed current representation with samples observed so far (denoted by "seen objects") or all samples including ones from future tasks (denoted by "all objects"). The pair of left figures shows contrastively trained representations suffer less from forgetting than the ones trained with cross entropy loss. The right pair shows contrastively learned representation is much more useful to perform unseen objects classification tasks.

Learning (Task-IL), Class Incremental Learning (Class-IL) and Domain Incremental Learning (Domain-IL) scenarios. We conduct experiments on Seq-CIFAR-10 and Seq-Tiny-ImageNet for Task-IL and Class-IL scenarios and R-MNIST for Domain-IL scenario. **Seq-CIFAR-10** is the set of splits (tasks) of the CIFAR-10 [26] dataset. We split the CIFAR-10 dataset into five separate sample sets, and each sample set consists of two classes. Similarly, **Seq-Tiny-ImageNet** is built from Tiny-ImageNet [1] by splitting 200 class samples into 10 disjoint sets of samples, each consisting of 20 classes. Seq-CIFAR-10 and Seq-Tiny-ImageNet split are given in the same order across different runs, as in [5]. We conduct experiments on **R-MNIST** [29] for Domain-IL experiments. For Domain-IL scenario, R-MNIST is constructed by rotating the original MNIST [27] images by a random degree in the range of $[0, \pi)$. R-MNIST consists of 20 tasks, corresponding to 20 uniformly randomly chosen degrees. We note that we treat samples from different domains with the same digit class as different classes while applying asymmetric supervised contrastive loss.

Training. We compare our contrastive continual learning algorithm with rehearsal-based continual learning baselines: ER [34], iCaRL [33], GEM [29], A-GEM [8], FDR [4], GSS [2], HAL [7], DER [5], and DER++ [5]. We train ResNet-18 [22] on Seq-CIFAR-10 and Tiny-ImageNet, and a simple network with convolution layers on R-MNIST. For all baselines, we report performance given in [5] of buffer size 200 and 500 except for R-MNIST since we choose a different architecture. More training details are provided in the supplementary material.

Evaluation protocol for Co^{2}L. As Co^{2}L is not a cross-entropy based coupled representation-classifier training, we need to train a classifier additionally. For a fair comparison, we train a classifier using only the last task samples and buffered samples on top of the frozen representations learned by Co^{2}L. To avoid the class-imbalance problems, we train a linear classifier with a class balanced sampling strategy, where first a class is selected uniformly from the set of classes, and then an instance from that class is subsequently uniformly sampled. We train a linear classifier for 100 epochs for all experiments, and we report classification test accuracy on this classifier.

5.2. Main Results

Validation of our key hypothesis. Before we provide results of Co^{2}L in comparison with other methods, we first validate our running premise for method design: *Contrastive learning learns more useful representation for the future task than the cross-entropy based coupled representation-classifier supervised learning.* This premise, however, is not easy to verify under the standard continual learning setup. Indeed, the quality of a representation is typically defined as the predictive performance with the best possible (linear) downstream classifier (see, *e.g.*, [3], and references therein), but optimal classifiers are only rarely learned under continual setups.

To circumvent this obstacle, we consider the following synthetic, yet insightful scenario: After training representations under the standard continual setup, we freeze the representations and freshly train the downstream classifier, using training data from *all tasks*. Here, the classifier trained on all observed samples so far will perform learned tasks well unless frozen representations suffer from forgetting.

As shown in the left pair of heatmaps in Figure 3, the average test accuracy on the previous tasks is surprisingly higher in *contrastive* than in *cross-entropy* (for off-diagonal parts, 21.79% vs. 66.46%). In other words, without any specific method to account for continual setup, contrastive method learns representations that suffer less from forgetting than ones trained with cross-entropy loss.

In the right pair of heatmaps in Figure 3, we report test

Buffer	Dataset	Seq-CIFAR-10		Seq-Tiny-ImageNet		R-MNIST
	Scenario	Class-IL	Task-IL	Class-IL	Task-IL	Domain-IL
200	ER [34]	44.79±1.86	91.19±0.94	8.49±0.16	38.17±2.00	93.53±1.15
	GEM [29]	25.54±0.76	90.44±0.94	-	-	89.86±1.23
	A-GEM [8]	20.04±0.34	83.88±1.49	8.07±0.08	22.77±0.03	89.03±2.76
	iCaRL [33]	49.02±3.20	88.99±2.13	7.53±0.79	28.19±1.47	-
	FDR [4]	30.91±2.74	91.01±0.68	8.70±0.19	40.36±0.68	93.71±1.51
	GSS [2]	39.07±5.59	88.80±2.89	-	-	87.10±7.23
	HAL [7]	32.36±2.70	82.51±3.20	-	-	89.40±2.50
	DER [5]	61.93±1.79	91.40±0.92	11.87±0.78	40.22±0.67	96.43±0.59
	DER++ [5]	64.88±1.17	91.92±0.60	10.96±1.17	40.87±1.16	95.98±1.06
	Co^{2}L (ours)	**65.57±1.37**	**93.43±0.78**	**13.88±0.40**	**42.37±0.74**	**97.90±1.92**
500	ER [34]	57.74±0.27	93.61±0.27	9.99±0.29	48.64±0.46	94.89±0.95
	GEM [29]	26.20±1.26	92.16±0.64	-	-	92.55±0.85
	A-GEM [8]	22.67±0.57	89.48±1.45	8.06±0.04	25.33±0.49	89.04±7.01
	iCaRL [33]	47.55±3.95	88.22±2.62	9.38±1.53	31.55±3.27	-
	FDR [4]	28.71±3.23	93.29±0.59	10.54±0.21	49.88±0.71	95.48±0.68
	GSS [2]	49.73±4.78	91.02±1.57	-	-	89.38±3.12
	HAL [7]	41.79±4.46	84.54±2.36	-	-	92.35±0.81
	DER [5]	70.51±1.67	93.40±0.39	17.75±1.14	51.78±0.88	97.57±1.47
	DER++ [5]	72.70±1.36	93.88±0.50	19.38±1.41	51.91±0.68	97.54±0.43
	Co^{2}L (ours)	**74.26±0.77**	**95.90±0.26**	**20.12±0.42**	**53.04±0.69**	**98.65 ±0.31**

Table 1. Classification accuracies for Seq-CIFAR-10, Seq-Tiny-ImageNet and R-MNIST on rehearsal-based baselines and our algorithm. We report performance of baslines of Seq-CIFAR-10 and Seq-Tiny-ImageNet from [5]. '-' indicates experiments unable to run due to compatibility issues (*e.g.*, iCaRL in Domain-IL) or intractable training time (*e.g.*, GEM, HAL or GSS on Tiny ImageNet). All results are averaged over ten independent trials. The best performance marked as bold.

accuracies of the classifiers that are trained with *all samples*, including the samples from unseen tasks. Interestingly, we observe that the average task accuracy on the unseen task is also notably higher in contrastively trained representations (rightmost heatmap) than ones trained with cross-entropy loss (second to right); for lower triangle parts, 32.77% vs. 62.76%. This implies that contrastive learning methods learn more highly transferable representations to future tasks, which might be the source of its robustness against forgetting.

Superiority of Co^{2}L over baselines. As shown in Table 1, our contrastive continual learning algorithm consistently outperforms all baselines in various scenarios, datasets, and memory sizes. Such results indicate that our algorithm successfully learns and preserves representations useful for future learning, and thus it significantly mitigates catastrophic forgetting. Moreover, such consistent gains over all comparisons show that our scheme is not limited to certain incremental learning scenarios. In what follows, we provide a more detailed analysis of our algorithm.

5.3. Ablation Studies

Effectiveness of IRD. To verify the effectiveness of IRD, we perform an ablation experiment with the class-IL setup on the Seq-CIFAR-10 dataset (identical to the setup in Section 5.2), with three additional variants of Co^{2}L. *(a) without buffer and IRD:* We optimize using only the SupCon loss (eq. 7); the symmetric version is identical to the asymmetric one since we do not use a replay buffer. *(b) with IRD only:* We use both (symmetric) SupCon loss and IRD loss. *(c) with replay buffer only:* We optimize the asymmetric SupCon loss (eq. 10) without an IRD loss. Note that while we do not use buffered samples to learn representations for (a,b), we still need buffered samples to train the downstream linear classifier; for (a,b), we use 200 auxiliary buffered samples to train the classifier (as in (c) and Co^{2}L).

As shown in Table 2, IRD brings a significant performance gain, with or without the replay buffer. With the replay buffer (rows (c,d)), we observe a 22.40% relative improvement; without the replay buffer (rows (a,b)), there is a 10.59% relative improvement. The former is noticeably larger than the latter; we suspect that maintaining the similarity structure of buffered samples (along with current task samples) is essential in preserving learned representations.

We also note that IRD seems to complement the asymmetric SupCon in terms of using buffered samples, leading to a performance boost. To verify this, we consider a synthetic *infinite-buffer* class-IL scenario: all past samples are available throught the training. Under this setup, we train a model with $\mathcal{L}^{\text{sup}}$ and another with $\mathcal{L}^{\text{sup}}_{\text{asym}}$ on Seq-CIFAR-10. As shown in Figure 4, asymmetric SupCon performs relatively poor without using IRD; under this class-balanced setup, not using past task samples as positive pairs only restricts learning. With increasing IRD power, however, the performance gap closes, indicating that IRD complements asymmetric SupCon by helping fully utilize the buffered samples. Such trend is also aligned with the results in Table 2; the performance boost from buffered samples–and

	Buffer Size	IRD	Accuracy(%)
(a) w/o buffer and IRD	0	✗	53.25±1.70
(b) w/ IRD only	0	✓	58.89±2.61
(c) w/ buffer only	200	✗	53.57±1.03
(d) Co^{2}L(ours)	200	✓	**65.57**±**1.37**

Table 2. Ablation study of Instance-wise Relation Distillation (IRD). We train our model on Seq-CIFAR-10 dataset under class-IL scenario (identical to the setup in Section 5.2) with ablated Co^{2}L. IRD brings significant gain with or without replay buffer. All results are averaged over ten independent trials.

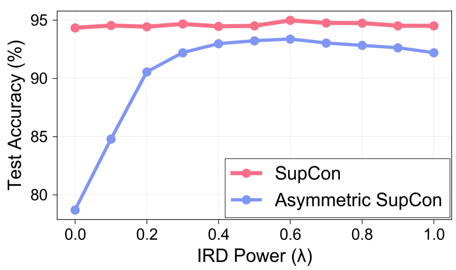


Figure 4. Performance comparison of original and asymmetric SupCon losses on Seq-CIFAR-10 under the ideal Class-IL scenario. Both settings use all past task samples. Instance-wise Relation Distillation (IRD) effectively closes the performance gap, which indicates IRD successfully retains learned representations without using past samples as positive pairs.

thus asymmetric SupCon loss–is relatively small without using IRD. This, however, does not necessarily imply that asymmetricity does not bring any benefit, as we will observe in the following ablation study on asymmetric SupCon.

Effectiveness of asymmetric supervised contrastive loss. To verify the effectiveness of asymmetric supervised contrastive loss, we compare two contrastive learning losses, the original SupCon and the asymmetric SupCon, as variants of Co^{2}L with the identical settings of Section 5.2. As shown in Table 3, asymmetric SupCon consistently provides gains over all counterparts with the original SupCon.

We also compare the visualizations of encoders' outputs of buffered and entire training samples of the Seq-CIFAR-10 dataset where the encoders are trained in the ablation experiments of Table 3. As illustrated in Figure 5, buffered samples' features trained with original SupCon are close to the same class samples while ones with asymmetric SupCon are well-spread. Since the buffered samples with asymmetric SupCon better represents the entire class sample population, representations trained on asymmetric SupCon show better task performance with linear classifiers. Such qualitative results are also well aligned with the motivation of asymmetric SupCon mentioned in Section 4.1 and provide the benefits of asymmetricity.

	Seq-CIFAR-10		Seq-Tiny-ImageNet	
Buffer	200	500	200	500
$\mathcal{L}^{\text{sup}}$	60.49±0.72	68.66±0.68	13.51±0.48	19.68±0.62
$\mathcal{L}^{\text{sup}}_{\text{asym}}$	**65.57**±**1.37**	**74.26**±**0.77**	**13.88**±**0.40**	**20.12**±**0.42**

Table 3. The effectiveness of asymmetric SupCon loss ($\mathcal{L}^{\text{sup}}_{\text{asym}}$) versus the original SupCon loss ($\mathcal{L}^{\text{sup}}$), combining with the IRD loss. All results are averaged over ten independent trials.

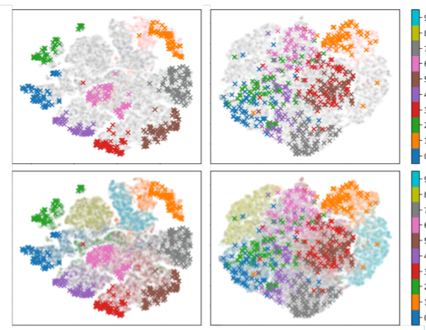


Figure 5. Top: t-SNE visualization of features from buffered (colored) and entire (gray) training samples of Seq-CIFAR-10. Bottom: Same as Top, but non-buffered samples are in opaque color instead of gray for a clear illustration of clusters. Left: Buffered samples' features trained with original SupCon are close to the same class samples but distant from different classes. Right: Buffered samples' features trained on asymmetric SupCon are well-spread; buffered samples better represent the entire class sample population.

6. Conclusion

We propose a contrastive continual learning scheme for learning representations under continual learning scenarios. The proposed asymmetric form of contrastive learning loss and the instance-wise relation distillation help model learn and preserve new and past representations and show a better performance over baselines on various learning setups. We hope that our work will serve as a good reference to how representation learning for continual learning should be designed.

Acknowledgements

This research was partly supported by the Engineering Research Center Program through the National Research Foundation of Korea (NRF) funded by the Korean Government MSIT (NRF-2018R1A5A1059921) and Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2019-0-00075, Artificial Intelligence Graduate School Program (KAIST)). We would like to thank Jaehyung Kim, Jihoon Tack, Sihyun Yu, Jongjin Park, and Yoonseo Choi for providing helpful feedback and suggestions.

References

[1] Stanford 231n. Tiny ImageNet visual recognition challenge. https://tiny-imagenet.herokuapp.com, 2015. 6

[2] Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. In *Advances in Neural Information Processing Systems*, 2019. 2, 6, 7

[3] Sanjeev Arora, Hrishikesh Khandeparkar, Mikhail Khodak, Orestis Plevrakis, and Nikunj Saunshi. A theoretical analysis of contrastive unsupervised representation learning. In *Proceedings of the International Conference on Machine Learning*, 2019. 6

[4] Ari S. Benjamin, David Rolnick, and Konrad P. Körding. Measuring and regularizing networks in function space. In *International Conference on Learning Representations*, 2019. 6, 7

[5] Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. In *Advances in Neural Information Processing Systems*, 2020. 1, 2, 6, 7

[6] Arslan Chaudhry, Puneet K. Dokania, Thalaiyasingam Ajanthan, and Phillip H. S. Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *European Conference on Computer Vision*, 2018. 1

[7] Arslan Chaudhry, Albert Gordo, Puneet K. Dokania, Philip H. S. Torr, and David Lopez-Paz. Using hindsight to anchor past knowledge in continual learning. In *Association for the Advancement of Artificial Intelligence*, 2020. 2, 6, 7

[8] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with A-GEM. In *International Conference on Learning Representations*, 2019. 6, 7

[9] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. SMOTE: Synthetic minority oversampling technique. *Journal of artificial intelligence research*, 2002. 5

[10] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the International Conference on Machine Learning*, 2020. 1, 3

[11] Ting Chen, Xiaohua Zhai, Marvin Ritter, Mario Lucic, and Neil Houlsby. Self-supervised gans via auxiliary rotation loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. 5

[12] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning, 2020. 3

[13] Zhiyuan Fang, Jianfeng Wang, Lijuan Wang, Lei Zhang, Yezhou Yang, and Zicheng Liu. SEED: Self-supervised distillation for visual representation. In *International Conference on Learning Representations*, 2021. 3, 4, 5

[14] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the International Conference on Machine Learning*, 2017. 3

[15] Jhair Gallardo et al. Self-supervised training enhances online continual learning. *arXiv preprint arXiv:2103.14010*, 2021. 3

[16] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning, 2020. 3

[17] Gunshi Gupta, Karmesh Yadav, and Liam Paull. Look-ahead meta learning for continual learning. In *Advances in Neural Information Processing Systems*, 2020. 3

[18] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the International Conference on Machine Learning*, 2010. 3

[19] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2006. 1

[20] Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. In *International Conference on Intelligent Computing*, 2005. 5

[21] 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*, 2020. 1, 3

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. 6

[23] Khurram Javed and Martha White. Meta-learning representations for continual learning. In *Advances in Neural Information Processing Systems*, 2019. 3

[24] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *Advances in Neural Information Processing Systems*, 2020. 1, 3

[25] James Kirkpatrick, Razvan Pascanu, Neil C. Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences of the United States of America*, 2017. 1

[26] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. 6

[27] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, 1998. 6

[28] Zhizhong Li and Derek Hoiem. Learning without forgetting. In *European Conference on Computer Vision*, 2016. 3

[29] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *International Conference on Learning Representations*, 2017. 1, 2, 6, 7

[30] Zheda Mai et al. Supervised contrastive replay: Revisiting the nearest class mean classifier in online class-incremental continual learning. *arXiv preprint arXiv:2103.13885*, 2021. 3

[31] Arun Mallya and Svetlana Lazebnik. PackNet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018. 1

[32] Michael McCloskey and Neal J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of Learning and Motivation*, 1989. 1

[33] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017. 2, 3, 6, 7

[34] 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 *International Conference on Learning Representations*, 2019. 1, 2, 6, 7

[35] Anthony Robins. Catastrophic forgetting, rehearsal, and pseudorehearsal. *Connection Science*, 1995. 1

[36] Joshua David Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples. In *International Conference on Learning Representations*, 2021. 2

[37] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 2015. 1

[38] Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint 1606.04671*, 2016. 1

[39] Joan Serra, Didac Suris, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. In *Proceedings of the International Conference on Machine Learning*, 2018. 1

[40] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019. 3

[41] Gido M. van de Ven and Andreas S Tolias. Three scenarios for continual learning. *arXiv preprint arXiv:1904.07734*, 2019. 3, 5

[42] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 3

[43] Song Zhang et al. Self-supervised learning aided class-incremental lifelong learning. *arXiv preprint arXiv:2006.05882*, 2020. 3