Contrastive Continuity on Augmentation Stability Rehearsal for Continual Self-Supervised Learning

Haoyang Cheng, Haitao Wen, Xiaoliang Zhang, Heqian Qiu *,
Lanxiao Wang, Hongliang Li *
University of Electronic Science and Technology of China, Chengdu, China
chenghaoyang@std.uestc.edu.cn, haitaowen@std.uestc.edu.cn, xlzhang@std.uestc.edu.cn,
hqqiu@uestc.edu.cn, lanxiao.wang@std.uestc.edu.cn, hlli@uestc.edu.cn

Abstract

Self-supervised learning has attracted a lot of attention recently, which is able to learn powerful representations without any manual annotations. However, self-supervised learning needs to develop the ability to continuously learn to cope with a variety of real-world challenges, i.e., Continual Self-Supervised Learning (CSSL). Catastrophic forgetting is a notorious problem in CSSL, where the model tends to forget the learned knowledge. In practice, simple rehearsal or regularization will bring extra negative effects while alleviating catastrophic forgetting in CSSL, e.g., overfitting on the rehearsal samples or hindering the model from encoding fresh information. In order to address catastrophic forgetting without overfitting on the rehearsal samples, we propose Augmentation Stability Rehearsal (ASR) in this paper, which selects the most representative and discriminative samples by estimating the augmentation stability for rehearsal. Meanwhile, we design a matching strategy for ASR to dynamically update the rehearsal buffer. In addition, we further propose Contrastive Continuity on Augmentation Stability Rehearsal (C^2ASR) based on ASR. We show that C^2ASR is an upper bound of the Information Bottleneck (IB) principle, which suggests that C^2ASR essentially preserves as much information shared among seen task streams as possible to prevent catastrophic forgetting and dismisses the redundant information between previous task streams and current task stream to free up the ability to encode fresh information. Our method obtains a great achievement compared with state-of-the-art CSSL methods on a variety of CSSL benchmarks.

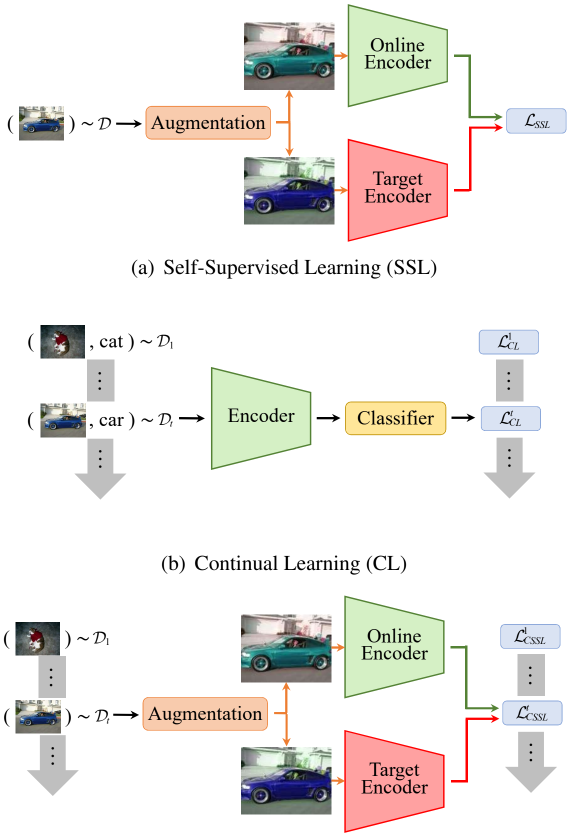
1. Introduction

Recently, self-supervised learning (SSL) has received much attention from the community due to its great potential [11, 21, 18, 5, 12, 49]. Self-supervised learning is able to learn beneficial representations for a variety of downstream tasks without any manual annotations, where contrastive learning based on dual branch framework is the mainstream, as shown in Figure 1(a). However, data is often presented as streams progressively over time in real-world scenarios. It's nearly infeasible for self-supervised learning to collect the whole data streams to train the networks, since the ever-increasing data makes the notoriously costly training of self-supervised learning models even more expensive and sometimes it can't even access previous data due to privacy protection. Self-supervised learning needs to develop continuity to cope with a variety of real-world challenges, which is also called Continual Self-Supervised Learning (CSSL) [16], as shown in Figure 1(c).

Continual learning (CL) aims to learn from non-stationary data distributions, as shown in Figure 1(b). Catastrophic forgetting is a notorious problem in CL, which refers to that the model tends to forget what it has already learned. Many methods [20, 40, 47, 39, 25, 50, 32, 1, 4] are proposed to alleviate it. CSSL also suffers from catastrophic forgetting, and some pioneers start to address this problem. Rehearsal-based method LUMP [33] utilizes rehearsal samples to augment current task samples by mixup [51], and regularization-based method CaSSLe [16] encourages current model to maintain a consistency with previous state via a prediction head. However, LUMP which is based on dark experience sampling strategy for rehearsal tends to overfit on the rehearsal samples due to the long training epoches of self-supervised learning [16], and CaSSLe introduces too much invariance among task streams, which preserves most information of previous task streams and hinders the model from learning fresh knowledge.

In order to address catastrophic forgetting without overfitting on the rehearsal samples, we propose Augmentation Stability Rehearsal (ASR) in this paper, which selects the most representative and discriminative samples by estimat-

[*] Corresponding authors.



(a) Self-Supervised Learning (SSL)

(b) Continual Learning (CL)

(c) Continual Self-Supervised Learning (CSSL)

Figure 1: A simplified illustration of SSL, CL and CSSL. (a) SSL aims to learn beneficial representations from images that are not manually labeled, which is usually based on the popular dual branch framework. (b) The proposition of CL is to learn from non-stationary data distributions, investing the neural networks with the ability to cope with a variety of real-world challenges. (c) CSSL is committed to investing self-supervised learning with the ability to deal with non-stationary data distributions, which aims to learn beneficial representations from non-stationary data distributions without manual annotations.

ing the augmentation stability for rehearsal. Specifically, ASR aims to select the samples which are located at the center and boundary of each category distribution, i.e., the most representative and discriminative samples, since they are able to retain the most information of previous task streams to overcome catastrophic forgetting as well as alleviating the overfitting effect. However, we are not able to obtain the relative position of each sample in corresponding category distribution, since we cannot access to the class label under unsupervised scenarios. Instead, we find the augmentation stability (i.e., the average similarity of the pairs of augmented features) of each sample is positively correlated with its relative position in the feature space. Thus, we design a rehearsal selection strategy based on the augmentation stability, i.e., selecting the samples with especially high score (located at the center of the category distribution) and low score (located at the boundary of the category distribution) from the augmentation stability distribution to fill the rehearsal buffer. Meanwhile, since the traditional queue and stack update strategy cannot satisfy the requirement that retaining the most representative and discriminative samples for the rehearsal buffer, we develop a matching strategy for ASR to dynamically update the rehearsal buffer.

Generally, the continual self-supervised model needs to encode the information of previous task streams to overcome catastrophic forgetting, as well as the information of current task to be of the ability to continuously learn. However, the whole information of previous task streams is not only redundant for preventing catastrophic forgetting [24], but also hinders the model from encoding fresh information. In order to balance the prevention of catastrophic forgetting and the ability to continuously learn, we further propose Contrastive Continuity on Augmentation Stability Rehearsal (C^{2}ASR) based on ASR, which aims to encourage the feature distribution of current model on rehearsal samples to be consistent with previous states to prevent catastrophic forgetting and the feature distribution of current model on current task samples to be inconsistent with previous states to free up the ability to continuously learn. In addition, we show that the proposed C^{2}ASR is an upper bound of the Information Bottleneck (IB) principle [42, 41], which suggests that C^{2}ASR essentially preserves as much information shared among seen task streams as possible to prevent catastrophic forgetting and dismisses the redundant information between previous task streams and current task stream to free up the ability to encode fresh information. Finally, we introduce the augmentation invariance and symmetrization strategy [18, 12] into C^{2}ASR to further increase the diversity and stability of contrastive continuity pairs.

We validate the effectiveness of our method on several popular CSSL benchmarks, e.g., the average accuracy and average forgetting on Split CIFAR-10, Split CIFAR-100 and Split Tiny-ImageNet and the average accuracy on out of distribution (OOD) datasets. Our method achieves the best performance on most evaluation metrics compared with state-of-the-art CSSL methods.

2. Related Work

2.1. Continual learning

Continual learning aims to learn from non-stationary data distributions without forgetting what it has been learned on seen data. Current popular partition manner mainly divides existing continual learning methods into three categories, i.e., regularization-based, architecture-based and rehearsal-based.

Regularization-based methods are to regulate the model parameters during training. EWC [25] alleviates catastrophic forgetting by slowing down the learning on the weights which are important to previous tasks during training. RWalk [9] proposes an online and efficient version of EWC. SI [50] introduces the synapses to track the parameter value of previous tasks, and fix the important synapses to keep the memories of the past. LwF [32] utilizes distillation to make the output of current network approach to that of previous networks. Based on network quantization and pruning, piggyback [34] learns binary masks to selectively mask the weights of the backbone network, and achieves better performance on new tasks. UCL [1] designs two regularization terms to alleviate forgetting by freezing important parameters of previous tasks and support future learning by controlling the active parameters.

Architecture-based methods are to dynamically add extra network architectures and design task-specific parameters to meet future learning requirements. PNN [39] introduces progressive networks to alleviate catastrophic forgetting and designs lateral connections to use learned knowledge to assist future learning. DEN [48] dynamically expands the capacity of the network according to each task, so as to effectively capture the shared knowledge among tasks to prevent forgetting. Utilizing architecture search, [31] finds the optimal structure for each task to best exploit the parameters shared among tasks.

Rehearsal-based methods are to replay a fixed number of previous learned samples during training. [7] stores some representative samples from previous tasks to alleviate intransigence. Based on the constrained optimization view of continual learning, [2] store the samples which best approximate the feasible region defined by the original constraints. DER [4] designs a dark experience sampling strategy to select rehearsal samples and utilizes distillation to match the output logits on the rehearsal data, thus preserving the memory of previous tasks. Rainbow Memory [3] focuses on the diversity of replayed data, and proposes memory management strategy based on classification prediction uncertainty to increase diversity.

Some methods focus on representation continual learning, which aims to prevent the model from forgetting the learned representation and utilize it for future learning. iCaRL [38] utilizes distillation to learn an anti-forgetting representation. Meta-learning-based approaches OML [23] and La-MAML [19] learn representations by designing special meta-objectives that prevent catastrophic forgetting and promote future learning. Inspired by contrastive learning, Co^{2}L [8] designs a supervised contrastive loss to learn a representation which is of nature resistance to catastrophic forgetting. LUMP [33] and CaSSLe [16] focus on continual self-supervised learning, where LUMP uses mixup [51] to merge rehearsal samples from previous tasks with the samples in current task and CaSSLe utilizes distillation mechanisms to associate the current state of the representation with its previous state via a prediction head to alleviate catastrophic forgetting.

2.2. Self-supervised learning

Self-supervised learning aims to learn a representation which is beneficial to various downstream tasks without any manual annotations. Some early works are devoted to designing special pretext tasks, e.g., Colorization [28], Inpainting [37], Jigsaw [36], Rotate prediction [17], etc. Contrastive learning based on instance discrimination [45] has become the mainstream in the community in recent years, whose core idea is to constrain input image to be as close as possible to its augmented view and far away from other images in the feature space. SimCLR [11] and MoCo [21] are the most classical contrastive learning methods, where SimCLR uses a large batchsize to increase the number of negative samples and MoCo introduces a queue to store a large number of negative samples and applies the momentum update strategy to ensure the consistency of negative samples. BYOL [18] argues that comparing with the negative samples is not indispensable in contrastive learning, and learns a brilliant representation by only encouraging augmentation invariance of input image. SimSiam [12] studies the non-negative-samples framework in detail and finds that siamese networks play an important role in the framework, where stop-gradient operation is the key to preventing collapsing. DINO [6] deploys self-supervised learning to ViT [14] and gets better results. Instead of applying stop-gradient operation to avoid collapsed solutions, Barlow Twins [49] constrains the cross-correlation matrix between the features of two augmented views to be the identity matrix, achieving the same results.

In addition, some self-supervised learning methods[30, 5, 44, 26, 15] add clustering to take account of the relationship among samples in the feature space, and get further improvements. PCL [30] combines clustering and augmentation invariance and designs an expectation maximization framework to perform clustering and augmentation invariance respectively, which searches for the semantic prototypes by K-means in step E and forces different augmented views of the same sample to be subject to the same prototype in step M. SwAV [5] introduces online clustering into contrastive learning, which obtains pseudo label assignments via online clustering and constrains different augmented views of the same sample to share the assignments.

3. Method

3.1. Augmentation Stability Rehearsal (ASR)

Generally, we should choose the most representative and discriminative samples for rehearsal [3], i.e., the samples which are located at the center and boundary of corresponding category distribution. These samples are able to retain the most information of previous tasks, which can largely overcome catastrophic forgetting as well as effectively alleviating the overfitting effect caused by the long training epoches of self-supervised learning [16].

In practice, a naive way to select the target samples is to determine their relative positions in corresponding category distribution by calculating the pairwise similarity. However, we are not able to access to the categories of samples in unsupervised scenarios, so as not to obtain the relative positions. Fortunately, we find the augmentation stability (i.e., the average similarity of the pairs of augmented features) of each sample is positively correlated with its relative position in corresponding category distribution. Therefore, we estimate the relative position distribution by utilizing the augmentation stability, and choose corresponding samples from the distribution to fill the rehearsal buffer.

Specifically, given an input image x, we first generate two augmented views $\mathcal{T}^1(x)$ and $\mathcal{T}^2(x)$ via the standard contrastive learning augmentation strategy $\mathcal{T}(\cdot)$. A view $\mathcal{T}^1(x)$ is fed to online encoder f_θ, and outputs the feature representation $z^1 = f_\theta\left(\mathcal{T}^1(x)\right)$. Another view $\mathcal{T}^2(x)$ is fed to target encoder $f_{\theta'}$ and outputs the corresponding feature $z^2 = f_{\theta'}\left(\mathcal{T}^2(x)\right)$. We design a discriminator to estimate the augmentation stability, which is essentially a binary classifier. The discriminator takes the pairwise features outputted by self-supervised model as input, and outputs the prediction probability of whether the input pairwise features is from the same image. During training, we construct the loss of the discriminator $\mathcal{L}_D$ as follows:

$$\begin{aligned}\mathcal{L}_D = & CE\left(D\left(\text{Concat}\left(z^1, z^2\right)\right), \text{“0”}\right) + \\ & CE\left(D\left(\text{Concat}\left(z^1, \bar{z}^2\right)\right), \text{“1”}\right)\end{aligned} \tag{1}$$

where z^1 and z^2 are the pairwise augmentation features encoded by self-supervised model, $\bar{z}^2$ are the augmentation features from other images, $\text{Concat}(\cdot)$ denotes the cascade operation, $D(\cdot)$ denotes the discriminator, $CE(\cdot)$ denotes the cross entropy loss. The combined augmentation feature from one image $\text{Concat}\left(z^1, z^2\right)$ is classified as class 0, while the combined augmentation feature from different images $\text{Concat}\left(z^1, \bar{z}^2\right)$ is classified as class 1. In summary, the discriminator aims to discriminate whether the input pairwise features are from the same image, so as to learn the ability to capture the augmentation stability.

When storing current data stream, we first utilize the discriminator to infer its augmentation stability score, i.e.,

$$p(y = \text{“0”}|x) = \mathbb{E}_{\mathcal{T}(\cdot)}\left[p_D\left(y = \text{“0”} | \left(z_\theta, z_{\theta'}\right)\right)\right] \tag{2}$$

where z_θ and $z_{\theta'}$ obey the augmentation feature distribution based online encoder and target encoder respectively, i.e., $z_\theta \sim f_\theta(\mathcal{T}(x))$ and $z_{\theta'} \sim f_{\theta'}(\mathcal{T}(x))$, $p_D\left(y = \text{“0”} | \left(z_\theta, z_{\theta'}\right)\right)$ is the prediction probability that the discriminator classifies the pairwise augmentation features $\text{Concat}\left(z_\theta, z_{\theta'}\right)$ as class 0. However, $\mathbb{E}_{\mathcal{T}(\cdot)}$ is almost infeasible to be calculated in practice. We approximate it by randomly sampling from the augmentation distribution:

$$\begin{aligned}p(y = \text{“0”}|x) &= \int_{z_\theta}\int_{z_{\theta'}} p_D\left(y = \text{“0”} | \left(z_\theta, z_{\theta'}\right)\right) d_{z_\theta} d_{z_{\theta'}} \\ &\approx \sum_i \sum_j p_D\left(y = \text{“0”} | \left(z^i, z^j\right)\right)\end{aligned} \tag{3}$$

where z^i and z^j is the sample from corresponding augmentation distribution, i.e., $z^i = f_\theta\left(\mathcal{T}^i(x)\right)$ and $z^j = f_{\theta'}\left(\mathcal{T}^j(x)\right)$. In practice, we set the actual sampling number to 20. Then, we use the augmentation stability score to rank current data stream, and select the appropriate samples according to the sorted list for rehearsal buffer.

ASR update strategy. We develop a matching update strategy for ASR to dynamically update the rehearsal buffer. Specifically, we recalculate the same amount of memory slots for all seen task streams when storing current data stream. Then, we discard the excess samples which are located in the middle of the augmentation stability sort for previous task streams (i.e., the least representative or discriminative ones) and load the selected samples of current data stream. We give the specific update process in Algorithm 1.

```
Algorithm 1 ASR Update Algorithm
Input: Buffer size: $K$, data stream of task $t$: $\mathcal{D}_t$, existing data in the buffer: $B_{t-1}$,
$B_t = \{\ \}$
$k_t = \lfloor K/t \rfloor$
for $i = 1$ to $t-1$ do
    $B_{t-1}^i = \{(x, task\_id) | task\_id = i, (x, task\_id) \in B_{t-1}\}$
    Sort $B_{t-1}^i$ by the augmentation stability computed by (3)
    $B_t$ += $B_{t-1}^i[0 : \lfloor k_t/2 \rfloor] + B_{t-1}^i[|B_{t-1}^i| - (k_t - \lfloor k_t/2 \rfloor) : |B_{t-1}^i|]$
end for
Sort $\mathcal{D}_t$ by the augmentation stability computed by (3)
Calculate memory slots of current data stream $k_t^{\mathcal{D}_t} = K - k_t * (t-1)$
$B_t$ += $\mathcal{D}_t[0 : \lfloor k_t^{\mathcal{D}_t}/2 \rfloor] + \mathcal{D}_t[|\mathcal{D}_t| - (k_t^{\mathcal{D}_t} - \lfloor k_t^{\mathcal{D}_t}/2 \rfloor) : |\mathcal{D}_t|]$
Output: Updated buffer $B_t$ after task stream $t$
```

3.2. Contrastive Continuity on Augmentation Stability Rehearsal (C^{2}ASR)

In practice, continual self-supervised model requires to encode the information of previous task streams to prevent catastrophic forgetting, as well as encoding the information of current task stream to be of the ability to continuously learn. One of the simplest ways to alleviate catastrophic forgetting is to encode the whole information of previous task streams. However, the whole information of previous task streams is not only redundant for preventing catastrophic forgetting [24], but also hinders the model from encoding fresh information. In order to dismiss the redundant information of previous task streams to balance the prevention of catastrophic forgetting and the learning on current task, we further propose Contrastive Continuity on Augmentation Stability Rehearsal (C^{2}ASR), which aims to encourage the feature distribution consistency on rehearsal samples between current model and previous states to prevent catastrophic forgetting, as well as the feature distribution inconsistency between current model and previous states on current task samples to free up the ability to continuously learn. We show that the proposed C^{2}ASR is an upper bound of the Information Bottleneck (IB) principle [42, 41], which suggests that C^{2}ASR essentially preserves as much information shared among seen task streams as possible to prevent catastrophic forgetting and dismisses the redundant information between previous task streams and current task stream to free up the ability to encode fresh information.

Specifically, given current date stream $\mathcal{D}_t$ and corresponding buffer B_{t-1} where we denote the data of task stream $\tau\,(\tau = 1, ..., t-1)$ in the buffer by B_{t-1}^{τ}, C^{2}ASR encourages the feature distribution $Z_{t|\tau} = f_t\left(B_{t-1}^{\tau}\right)$ to be consistent with $Z_{\tau|\tau} = f_\tau\left(B_{t-1}^{\tau}\right)$ to prevent catastrophic forgetting, and encourages the feature distribution $Z_{t|t} = f_t\left(\mathcal{D}_t\right)$ to be inconsistent with $Z_{\tau|t} = f_\tau\left(\mathcal{D}_t\right)$ to free up the ability to continuously learn, whose loss function is as follows:

$$\mathcal{L}_\tau = -\log \frac{\exp \sum\limits_{B_{t-1}^{\tau}} \log \frac{p\left(z_{t|\tau}|z_{\tau|\tau}\right)}{p\left(z_{t|\tau}\right)}}{\exp \sum\limits_{\mathcal{D}_t} \log \frac{p\left(z_{t|t}|z_{\tau|t}\right)}{p\left(z_{t|t}\right)} + \exp \sum\limits_{B_{t-1}^{\tau}} \log \frac{p\left(z_{t|\tau}|z_{\tau|\tau}\right)}{p\left(z_{t|\tau}\right)}} \tag{4}$$

where $z_{t|\tau} \sim Z_{t|\tau}$, $z_{\tau|\tau} \sim Z_{\tau|\tau}$, $z_{t|t} \sim Z_{t|t}$ and $z_{\tau|t} \sim Z_{\tau|t}$. Formally, C^{2}ASR constrains the density ratio $p\left(z_{t|\tau}|z_{\tau|\tau}\right)/p\left(z_{t|\tau}\right)$ which represents the correlation between $z_{t|\tau}$ and $z_{\tau|\tau}$ to be larger and the density ratio $p\left(z_{t|t}|z_{\tau|t}\right)/p\left(z_{t|t}\right)$ which represents the correlation between $z_{t|t}$ and $z_{\tau|t}$ to be smaller. Theoretically, we show that $\mathcal{L}_\tau$ is an upper bound of the Information Bottleneck (IB) principle[42, 41] at the end of this subsection, which suggests that C^{2}ASR essentially preserves as much information shared among seen task streams as possible to prevent catastrophic forgetting and dismisses the redundant information between previous task streams and current task stream to free up the ability to encode fresh information.

In practice, the density ratio $p\left(z_{t|\tau}|z_{\tau|\tau}\right)/p\left(z_{t|\tau}\right)$ and $p\left(z_{t|t}|z_{\tau|t}\right)/p\left(z_{t|t}\right)$ are almost infeasible to calculate, and we model them as exponential feature similarity via a non-linear head [43]:

$$\exp\left(\mathrm{sim}\left(h\left(z_{t|\tau}\right), z_{\tau|\tau}\right)/\epsilon\right) \rightarrow \frac{p\left(z_{t|\tau}|z_{\tau|\tau}\right)}{p\left(z_{t|\tau}\right)} \tag{5}$$

$$\exp\left(\mathrm{sim}\left(h\left(z_{t|t}\right), z_{\tau|t}\right)/\epsilon\right) \rightarrow \frac{p\left(z_{t|t}|z_{\tau|t}\right)}{p\left(z_{t|t}\right)} \tag{6}$$

where $h(\cdot)$ denotes the non-linear head, ϵ is a hyperparameter. With above approximation, we can rewrite (4) as:

$$\mathcal{L}_\tau = -\log \frac{\exp \sum\limits_{B_{t-1}^{\tau}} s\left(z_{t|\tau}, z_{\tau|\tau}\right)}{\exp \sum\limits_{\mathcal{D}_t} s\left(z_{t|t}, z_{\tau|t}\right) + \exp \sum\limits_{B_{t-1}^{\tau}} s\left(z_{t|\tau}, z_{\tau|\tau}\right)} \tag{7}$$

where $s(\cdot,\cdot)$ denotes $\mathrm{sim}\left(h(\cdot), \cdot\right)/\epsilon$.

Obviously, there is an imbalance between current task samples $\mathcal{D}_t$ and rehearsal samples B_{t-1}^{τ}, i.e., $|\mathcal{D}_t| \gg |B_{t-1}^{\tau}|$. Thus, we sample a subset $\mathcal{D}_t^{\tau}$ from $\mathcal{D}_t$ ($|\mathcal{D}_t^{\tau}| = |B_{t-1}^{\tau}|$) to address the imbalance problem, as well as reducing the computational complexity:

$$\mathcal{L}_\tau = -\log \frac{\exp \sum\limits_{B_{t-1}^{\tau}} s\left(z_{t|\tau}, z_{\tau|\tau}\right)}{\exp \sum\limits_{\mathcal{D}_t^{\tau}} s\left(z_{t|t}, z_{\tau|t}\right) + \exp \sum\limits_{B_{t-1}^{\tau}} s\left(z_{t|\tau}, z_{\tau|\tau}\right)} \tag{8}$$

Formally, $\mathcal{L}_\tau$ is similar to the InfoNCE loss in [43, 11, 21]. Inspired by the evolution of infoNCE loss to cosine similarity loss in contrastive learning, we also give the trivial form of $\mathcal{L}_\tau$ on similarity level, which has the same optimization direction and competitive performance:

$$\mathcal{L}_\tau = -\sum_{B_{t-1}^{\tau}} \frac{s\left(z_{t|\tau}, z_{\tau|\tau}\right)}{\left|B_{t-1}^{\tau}\right|} + \sum_{\mathcal{D}_t^{\tau}} \frac{s\left(z_{t|t}, z_{\tau|t}\right)}{\left|\mathcal{D}_t^{\tau}\right|} \tag{9}$$

Meanwhile, we combine C^{2}ASR with augmentation invariance to increase the diversity of contrastive continuity pairs. Specifically, C^{2}ASR encourages the augmentation feature distribution $Z_{t|\tau} = f_t\left(\mathcal{T}^1\left(B_{t-1}^{\tau}\right)\right)$ to be consistent with $Z_{\tau|\tau} = f_\tau\left(\mathcal{T}^2\left(B_{t-1}^{\tau}\right)\right)$, and encourage the feature distribution $Z_{t|t} = f_t\left(\mathcal{T}^1\left(\mathcal{D}_t\right)\right)$ to be inconsistent with $Z_{\tau|t} = f_\tau\left(\mathcal{T}^2\left(\mathcal{D}_t\right)\right)$. In addition, the symmetrization strategy [18, 12] is applied to further increase the diversity, as well as reinforcing the stability, i.e., additionally encourage the augmentation feature distribution $Z_{t|\tau} = f_t\left(\mathcal{T}^2\left(B_{t-1}^{\tau}\right)\right)$ to be consistent with $Z_{\tau|\tau} = f_\tau\left(\mathcal{T}^1\left(B_{t-1}^{\tau}\right)\right)$ and the feature distribution $Z_{t|t} = f_t\left(\mathcal{T}^2\left(\mathcal{D}_t\right)\right)$ to be inconsistent with $Z_{\tau|t} = f_\tau\left(\mathcal{T}^1\left(\mathcal{D}_t\right)\right)$.

Finally, our C^{2}ASR loss is implemented by performing $\mathcal{L}_\tau$ on appropriate previous task stream interval:

$$\mathcal{L}_{C^2ASR}^{t} = \frac{1}{t-m} \sum_{\tau=m}^{t-1} \mathcal{L}_\tau, 1 \leqslant m \leqslant t-1 \tag{10}$$

In practice, we choose previous 2 task streams to calculate C^{2}ASR loss to trade off the computational cost brought by large task stream interval and the information leakage brought by small task stream interval:

$$\mathcal{L}_{C^2ASR}^t = \frac{1}{2} \sum_{\tau=t-2}^{t-1} \mathcal{L}_\tau \tag{11}$$

Our training loss is constructed by weighted sum of the trivial continual self-supervised learning loss and C^{2}ASR loss:

$$\mathcal{L}^t = \mathcal{L}_{CSSL}^t + \lambda \mathcal{L}_{C^2ASR}^t \tag{12}$$

where λ is the weighted parameter which is set to 0.1.

Relation to the Information Bottleneck (IB) principle[42, 41]. The IB principle argues that a desirable representation Z should provide as much important information related to Y as possible while compressing the original information from X by dismissing the redundant part, i.e., increase the mutual information between Z and Y and decrease the the mutual information between Z and X:

$$IB = I\left(Z;X\right) - \beta I\left(Z;Y\right) \tag{13}$$

where $I\left(\,;\,\right)$ denotes mutual information and β is a hyper-parameter to trade off the amount of preserved important information and dismissed redundant part.

We show that $\mathcal{L}_\tau$ in (4) is an upper bound of the IB principle, as follows:

$$\mathcal{L}_\tau = -\log \frac{\exp \sum\limits_{B_{t-1}^\tau} \log \frac{p\left(z_{t|\tau} | z_{\tau|\tau}\right)}{p\left(z_{t|\tau}\right)}}{\exp \sum\limits_{\mathcal{D}_t} \log \frac{p\left(z_{t|t} | z_{\tau|t}\right)}{p\left(z_{t|t}\right)} + \exp \sum\limits_{B_{t-1}^\tau} \log \frac{p\left(z_{t|\tau} | z_{\tau|\tau}\right)}{p\left(z_{t|\tau}\right)}} \tag{14}$$

$$\geqslant I\left(Z_{t|t}; Z_{\tau|t}\right) - \frac{\left|B_{t-1}^\tau\right|}{\left|\mathcal{D}_t\right|} I\left(Z_{t|\tau}; Z_{\tau|\tau}\right) \tag{15}$$

where the hyper-parameter β is equal to $\left|B_{t-1}^\tau\right| / \left|\mathcal{D}_t\right|$. Please refer to Appendix A for the details about the proof.

Based on IB principle, C^{2}ASR essentially encourages the model to encode as much information shared among seen task streams as possible by increasing the mutual information $I\left(Z_{t|\tau}; Z_{\tau|\tau}\right)$ and dismiss the redundant information between previous task streams and current task stream by decreasing the mutual information $I\left(Z_{t|t}; Z_{\tau|t}\right)$. It's worth noting in IB principle that it doesn't mean the representation Z doesn't contain the information of X (corresponding to decreasing the mutual information $I\left(Z;X\right)$), but it needs to dismiss the redundant part of X that is irrelevant to $I\left(Z;Y\right)$. The same is true for C^{2}ASR that it doesn't mean current $f_t\left(\cdot\right)$ doesn't encode the information of previous state $f_\tau\left(\cdot\right)$ on current task stream (corresponding to decreasing the mutual information $I\left(Z_{t|t}; Z_{\tau|t}\right)$), but it needs to dismiss the redundant part of $f_\tau\left(\cdot\right)$ that is irrelevant to $I\left(Z_{t|\tau}; Z_{\tau|\tau}\right)$.

4. Experiments

In this section, we give the experimental results on a variety of CSSL benchmarks. We first describe the experimental setup in 4.1, and provide the main results in 4.2. Then, we report the evaluation results on out of distribution datasets (OOD datasets) in 4.3, and finally we evaluate the effectiveness and expansibility of the proposed components in 4.4. In addition, we give more ablation studies and visualization in Appendix B.

4.1. Experimental setup

Datasets. We deploy our experiments on Split CIFAR-10 [27] (a 10-class dataset with 32×32 images), Split CIFAR-100 [27] (a 100-class dataset with 32×32 images) and Split Tiny-ImageNet [13] (a 100-class dataset with 64×64 images). We follow the division in [33] for the datasets, i.e., two random classes per task stream for CIFAR-10, five random classes per task stream for CIFAR-100 and Tiny-ImageNet.

Implementation details. We use ResNet-18 [22] as the backbone and SimSiam [12] as the base self-supervised learning algorithm to make a fair comparison with existing methods. LUMP's technique is an effective rehearsal strategy, and we also apply it for rehearsal samples in practice. We train our method with SGD optimizer for 200 epoches, whose batchsize is 128, learning rate is 0.015, weight decay is 5e-4, and momentum is 0.9. The buffer size in our method is set to 200 for CIFAR-10 and CIFAR-100, 256 for Tiny-ImageNet.

Evaluation metrics. We follow LUMP to utilize the KNN classifier [45] to verify the quality of the learned representation, where ”Average Accuracy” and ”Average Forgetting” are served as the two key indicators. Specifically, we use a KNN classifier on the frozen pre-trained representation after learning each task stream to evaluate the test accuracy. To allow for simplification, we denote ”the test accuracy by training on task stream i and testing on task stream j” by ”Tr_iTe_j”, ”the forgetting of task stream j” by ”F_j”, where ”$F_j = \underset{1 \leqslant k \leqslant T}{max} Tr_kTe_j - Tr_TTe_j$”, i.e., the accuracy decrease of task stream j between its maximum accuracy and the accuracy after learning the final task stream T.

The average accuracy after training on task stream t is defined by the average of the test accuracy on all seen task streams:

$$A_t = \frac{1}{t} \sum_{j=1}^{t} Tr_tTe_j \tag{16}$$

The average forgetting is defined by the average forget-

Table 1: The main results (Average Accuracy and Average Forgetting) on Split CIFAR-10, Split CIFAR-100 and Split Tiny-ImageNet. All methods are pre-trained with Resnet-18 as backbone for 200 epoches and evaluated with KNN classifier [45]. CaSSLe† is the improved reproduced version by incorporating with the replay strategy in LUMP [33] to make a fair comparison. All the performances are measured by calculating the mean and standard deviation across three trials. The Top-2 results are highlighted in bold and underlined respectively.

Method	CIFAR-10		CIFAR-100		Tiny-ImageNet	
	Accuracy	Forgetting	Accuracy	Forgetting	Accuracy	Forgetting
			Supervised Continual Learning			
FINETUNE	82.87(±0.47)	14.26(±0.52)	61.08(±0.04)	31.23(±0.41)	53.10(±1.37)	33.15(±1.22)
PNN [39]	82.74(±2.12)	-	66.05(±0.86)	-	64.38(±0.92)	-
SI [50]	85.18(±0.65)	11.39(±0.77)	63.58(±0.37)	27.98(±0.34)	44.96(±2.41)	26.29(±1.40)
A-GEM [10]	82.41(±1.24)	13.82(±1.27)	59.81(±1.07)	30.08(±0.91)	60.45(±0.24)	24.94(±1.24)
GSS [2]	89.49(±1.75)	7.50(±1.52)	70.78(±1.67)	21.28(±1.52)	70.96(±0.72)	14.76(±1.22)
DER [4]	91.35(±0.46)	5.65(±0.35)	79.52(±1.88)	12.80(±1.47)	68.03(±0.85)	17.74(±0.65)
MULTITASK	97.77(±0.15)	-	93.89(±0.78)	-	91.79(±0.46)	-
			Continual Self-Supervised Learning			
FINETUNE	90.11(±0.12)	5.42(±0.08)	75.42(±0.78)	10.19(±0.37)	71.07(±0.20)	9.48(±0.56)
PNN [39]	90.93(±0.22)	-	66.58(±1.00)	-	62.15(±1.35)	-
DER [4]	91.22(±0.30)	4.63(±0.26)	77.27(±0.30)	9.31(±0.09)	71.90(±1.44)	8.36(±2.06)
LUMP [33]	91.00(±0.40)	2.92(±0.53)	82.30(±1.35)	4.71(±1.52)	76.66(±2.39)	3.54(±1.04)
CaSSLe† [16]	$\underline{91.51}$(±0.38)	$\underline{2.77}$(±0.54)	$\underline{82.65}$(±1.24)	$\underline{3.26}$(±1.39)	$\underline{77.26}$(±2.03)	$\underline{3.27}$(±0.88)
C^{2}ASR(Ours)	**92.47**(±0.41)	**2.59**(±0.58)	**83.12**(±0.92)	**2.22**(±1.48)	**77.85**(±1.87)	**3.08**(±0.79)
MULTITASK	95.76(±0.08)	-	86.31(±0.38)	-	82.89(±0.49)	-

ting of the first $T-1$ task streams:

$$F = \frac{1}{T-1}\sum_{j=1}^{T-1} F_j \tag{17}$$

$$= \frac{1}{T-1}\sum_{j=1}^{T-1}\left(\underset{1\leqslant k\leqslant T}{max} Tr_k Te_j - Tr_T Te_j\right) \tag{18}$$

4.2. Main results

In this subsection, we report the main results (Average Accuracy and Average Forgetting) of our method C^{2}ASR on Split CIFAR-10, Split CIFAR-100 and Split Tiny-ImageNet, as shown in Table 1. Compared with the existing continual self-supervised learning methods, our C^{2}ASR achieves the best performance on all evaluation metrics. The performance gains are mainly reflected in two aspects. On the one hand, C^{2}ASR has a better resistance to forgetting. For example, C^{2}ASR obtains 0.33%, 2.49%, 0.46% and 0.18%, 1.04%, 0.19% average forgetting drops on Split CIFAR-10, Split CIFAR-100 and Split Tiny-ImageNet compared with LUMP and CaSSLe† respectively. On the other hand, C^{2}ASR frees up the ability to continuously learn on new tasks, e.g. it obtains 1.47%, 0.82%, 1.19% and 0.96%, 0.47%, 0.59% average accuracy improvements on Split CIFAR-10, Split CIFAR-100 and Split Tiny-ImageNet compared with LUMP and CaSSLe† respectively.

4.3. Evaluation on OOD datasets

In this subsection, we report the average accuracy of the proposed C^{2}ASR on out of distribution (OOD) datasets, where we recognise MNIST [29], Fashion-MNIST (FMNIST) [46], SVHN [35], CIFAR-100 and MNIST [29], Fashion-MNIST (FMNIST) [46], SVHN [35], CIFAR-10 as the out of distribution datasets for Split CIFAR-10 and Split CIFAR-100 respectively, as shown in Table 2. The proposed C^{2}ASR obtains significant improvements and achieves the best performance on all evaluation metrics compared with the existing continual self-supervised learning methods, showing the learned representation by C^{2}ASR can be easily and effectively applied to unseen data distributions.

4.4. The effectiveness and expansibility of proposed Augmentation Stability Rehearsal (ASR) and Contrastive Continuity (CC)

We add extra experiments on Split CIFAR-10 to evaluate the effectiveness and expansibility of proposed Augmentation Stability Rehearsal (ASR) and Contrastive Con-

Table 2: The average accuracy on out of distribution (OOD) datasets. All methods are pre-trained with Resnet-18 as backbone for 200 epoches on Split CIFAR-10 or Split CIFAR-100 and evaluated with KNN classifier [45] on out of distribution datasets i.e., MNIST [29], Fashion-MNIST (FMNIST) [46], SVHN [35], CIFAR-100 or CIFAR-10. CaSSLe† is the improved reproduced version by incorporating with the replay strategy in LUMP [33] to make a fair comparison. All the performances are measured by calculating the mean and standard deviation across three trials. The Top-2 results are highlighted in bold and underlined respectively.

In-class	CIFAR-10				CIFAR-100			
Out of class	MNIST	FMNIST	SVHN	CIFAR-100	MNIST	FMNIST	SVHN	CIFAR-10
Supervised Continual Learning								
FINETUNE	86.42(±1.11)	74.47(±0.84)	41.00(±0.85)	17.42(±0.96)	75.02(±3.97)	62.37(±3.20)	38.05(±0.73)	39.18(±0.83)
SI [50]	87.08(±0.79)	76.41(±0.81)	42.62(±1.31)	19.14(±0.91)	79.96(±2.63)	63.71(±1.36)	40.92(±1.64)	40.41(±1.71)
A-GEM [10]	86.07(±1.94)	74.74(±3.21)	37.77(±3.49)	16.11(±0.38)	77.56(±3.21)	64.16(±2.29)	37.48(±1.73)	37.91(±1.33)
GSS [2]	70.36(±3.54)	69.20(±2.51)	33.11(±2.26)	18.21(±0.39)	76.54(±0.46)	65.31(±1.72)	35.72(±2.37)	49.41(±1.81)
DER [4]	80.32(±1.91)	70.49(±1.54)	41.48(±2.76)	17.72(±0.25)	87.71(±2.23)	75.97(±1.29)	50.26(±0.95)	59.07(±1.06)
MULTITASK	88.79(±1.13)	79.50(±0.52)	41.26(±1.95)	27.68(±0.66)	92.29(±3.37)	86.12(±1.87)	54.94(±1.77)	54.04(±3.68)
Continual Self-Supervised Learning								
FINETUNE	89.23(±0.99)	80.05(±0.34)	49.66(±0.81)	34.52(±0.12)	85.99(±0.86)	76.90(±0.11)	50.09(±1.41)	57.15(±0.96)
DER [4]	88.35(±0.82)	79.33(±0.62)	48.83(±0.55)	30.68(±0.36)	87.96(±2.04)	76.21(±0.63)	47.70(±0.94)	56.26(±0.16)
LUMP [33]	91.03(±0.22)	80.78(±0.88)	45.18(±1.57)	31.17(±1.83)	91.76(±1.17)	81.61(±0.45)	50.13(±0.71)	63.00(±0.53)
CaSSLe† [16]	91.43(±0.33)	80.97(±0.86)	53.31(±1.09)	37.49(±1.46)	91.92(±1.21)	81.87(±0.50)	53.24(±1.19)	66.85(±1.07)
C^{2}ASR(Ours)	**92.14**(±0.38)	**81.48**(±0.79)	**54.51**(±0.84)	**39.48**(±1.12)	**93.09**(±1.38)	**82.04**(±0.54)	**56.31**(±1.85)	**67.74**(±0.97)
MULTITASK	90.69(±0.13)	80.65(±0.42)	47.67(±0.45)	39.55(±0.18)	90.35(±0.24)	81.11(±1.86)	52.20(±0.61)	70.19(±0.15)

tinuity (CC). We report the average accuracy, average forgetting of ASR and the combination between CC and other data replay methods in Table 3. The proposed rehearsal strategy ASR obtains superior performance than DER and LUMP, showing its effectiveness of selecting replayed data. And C^{2}ASR obtains further improvement based on ASR, which shows the effectiveness of the proposed regularization strategy CC to extract important information. In addition, CC can also be combined with other data replay methods, e.g., DER [4] and LUMP [33]. As shown in Table 3, DER + CC and LUMP + CC obtains considerable improvements compared with corresponding baseline, which indicates that the proposed regularization strategy CC can be effectively extended to existing data replay methods.

5. Conclusion

In this paper, we study how to address catastrophic forgetting in Continual Self-Supervised Learning without bringing extra negative effects. We first propose ASR to store the most representative and discriminative samples for rehearsal, which helps to prevent catastrophic forgetting as well as overcoming the overfitting on the rehearsal samples. Secondly, we further propose C^{2}ASR based on ASR. We show that the proposed method is an upper bound of the IB principle. It suggests that C^{2}ASR essentially preserves as much information shared among seen task streams as possible to prevent catastrophic forgetting and dismisses the redundant information between previous task streams and current task stream to free up the ability to encode fresh information. The massive experimental results on several popular CSSL benchmarks show the superiority and competitiveness of the proposed method.

Table 3: **The effectiveness and expansibility of proposed Augmentation Stability Rehearsal (ASR) and Contrastive Continuity (CC)**. We report the average accuracy, average forgetting of ASR and the combination between CC and other data replay methods on Split CIFAR-10. The performances are measured by calculating the mean and standard deviation across three trials. The best results are highlighted in bold.

	Accuracy	Forgetting
FINETUNE	90.11(±0.12)	5.42(±0.08)
ASR (Ours)	91.62(±0.33)	2.74(±0.51)
C^{2}ASR (Ours)	**92.47**(±0.41)	**2.59**(±0.58)
DER [4]	91.22(±0.30)	4.63(±0.26)
DER + CC (Ours)	**91.85**(±0.21)	**3.35**(±0.34)
LUMP [33]	91.00(±0.40)	2.92(±0.53)
LUMP + CC (Ours)	**92.03**(±0.36)	**2.78**(±0.45)

6. Acknowledgements

This work was supported in part by National Key R&D Program of China (2021ZD0112001) and National Natural Science Foundation of China (No. 61831005).

References

[1] Hongjoon Ahn, Sungmin Cha, Donggyu Lee, and Taesup Moon. Uncertainty-based continual learning with adaptive regularization. In *Advances in Neural Information Processing Systems*, pages 4394–4404, Vancouver, BC, Canada, December 2019. 1, 3

[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*, pages 11816–11825, Vancouver, BC, Canada, December 2019. 3, 7, 8

[3] Jihwan Bang, Heesu Kim, YoungJoon Yoo, Jung-Woo Ha, and Jonghyun Choi. Rainbow memory: Continual learning with a memory of diverse samples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8218–8227, Virtual, June 2021. 3

[4] 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*, pages 15920–15930, virtual, December 2020. 1, 3, 7, 8

[5] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *Advances in Neural Information Processing Systems*, pages 9912–9924, Virtual, December 2020. 1, 3

[6] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9650–9660, Virtual, October 2021. 3

[7] 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*, pages 233–248, Munich, Germany, September 2018. 3

[8] 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, Virtual, October 2021. 3

[9] Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European Conference on Computer Vision*, pages 532–547, Munich, Germany, September 2018. 3

[10] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. In *Proceedings the International Conference on Learning Representations*, New Orleans, LA, USA, May 2019. 7, 8

[11] 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*, pages 1597–1607, Virtual, July 2020. 1, 3, 5

[12] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, Virtual, June 2021. 1, 2, 3, 5, 6

[13] Jia Deng, Wei Dong, Richard Socher, Li Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, Miami, Florida, USA, June 2009. 6

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings the International Conference on Learning Representations*, Virtual, May 2021. 3

[15] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9588–9597, Virtual, October 2021. 3

[16] 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, Virtual, June 2022. 1, 3, 4, 7, 8

[17] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *Proceedings of the International Conference on Learning Representations*, Vancouver, BC, Canada, April 2018. 3

[18] Jean Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. In *Advances in Neural Information Processing Systems*, pages 21271–21284, Virtual, December 2020. 1, 2, 3, 5

[19] Gunshi Gupta, Karmesh Yadav, and Liam Paull. Look-ahead meta learning for continual learning. In *Advances in Neural Information Processing Systems*, pages 11588–11598, Virtual, December 2020. 3

[20] Mahmudul Hasan and Amit K Roy-Chowdhury. A continuous learning framework for activity recognition using deep hybrid feature models. *IEEE Transactions on Multimedia*, 17(11):1909–1922, 2015. 1

[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*, pages 9729–9738, Virtual, June 2020. 1, 3, 5

[22] 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, Las Vegas, NV, USA, June 2016. 6

[23] Khurram Javed and Martha White. Meta-learning representations for continual learning. In *Advances in Neural Infor-*

mation Processing Systems, pages 1818–1828, Vancouver, BC, Canada, December 2019. 3

[24] Minsoo Kang, Jaeyoo Park, and Bohyung Han. Class-incremental learning by knowledge distillation with adaptive feature consolidation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16071–16080, Virtual, June 2022. 2, 5

[25] 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. 1, 3

[26] Soroush Abbasi Koohpayegani, Ajinkya Tejankar, and Hamed Pirsiavash. Mean shift for self-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10326–10335, Virtual, October 2021. 3

[27] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images, 2009. 6

[28] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Learning representations for automatic colorization. In *Proceedings of the European Conference on Computer Vision*, pages 577–593, Amsterdam, Netherlands, September 2016. 3

[29] Yann LeCun. The mnist database of handwritten digits. http://yann.lecun.com/exdb/mnist/, 1998. 7, 8

[30] Junnan Li, Pan Zhou, Caiming Xiong, and Steven CH Hoi. Prototypical contrastive learning of unsupervised representations. In *Proceedings of the International Conference on Learning Representations*, Virtual, May 2021. 3

[31] Xilai Li, Yingbo Zhou, Tianfu Wu, Richard Socher, and Caiming Xiong. Learn to grow: A continual structure learning framework for overcoming catastrophic forgetting. In *Proceedings of the International Conference on Machine Learning*, pages 3925–3934, Long Beach, California, USA, June 2019. 3

[32] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, 2017. 1, 3

[33] Divyam Madaan, Jaehong Yoon, Yuanchun Li, Yunxin Liu, and Sung Ju Hwang. Representational continuity for unsupervised continual learning. In *Proceedings the International Conference on Learning Representations*, Virtual, April 2022. 1, 3, 6, 7, 8

[34] 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*, pages 72–88, Munich, Germany, September 2018. 3

[35] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011. 7, 8

[36] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *Proceedings of the European Conference on Computer Vision*, pages 69–84, Amsterdam, Netherlands, September 2016. 3

[37] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2536–2544, Las Vegas, NV, USA, June 2016. 3

[38] 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, Honolulu, HI, USA, July 2017. 3

[39] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv:1606.04671*, 2016. 1, 3, 7

[40] Selvarajah Thuseethan, Sutharshan Rajasegarar, and John Yearwood. Deep continual learning for emerging emotion recognition. *IEEE Transactions on Multimedia*, 2021. 1

[41] Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000. 2, 5, 6

[42] Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *Proceedings of IEEE Information Theory Workshop (ITW)*, pages 1–5, Jerusalem, Israel, April 2015. 2, 5, 6

[43] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv:1807.03748*, 2018. 5

[44] Xudong Wang, Ziwei Liu, and Stella X Yu. Unsupervised feature learning by cross-level instance-group discrimination. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12586–12595, Virtual, June 2021. 3

[45] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3733–3742, Salt Lake City, UT, USA, June 2018. 3, 6, 7, 8

[46] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv:1708.07747*, 2017. 7, 8

[47] Guanglei Yang, Enrico Fini, Dan Xu, Paolo Rota, Mingli Ding, Tang Hao, Xavier Alameda-Pineda, and Elisa Ricci. Continual attentive fusion for incremental learning in semantic segmentation. *IEEE Transactions on Multimedia*, 2022. 1

[48] Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. In *Proceedings the International Conference on Learning Representations*, Vancouver, BC, Canada, May 2018. 3

[49] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *Proceedings of the International Conference on Machine Learning*, pages 12310–12320, Virtual, July 2021. 1, 3

[50] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *Proceedings*

of the International Conference on Machine Learning, pages 3987–3995, Sydney, Australia, August 2017. 1, 3, 7, 8

[51] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *Proceedings of the International Conference on Learning Representations*, Vancouver, BC, Canada, May 2018. 1, 3