CONTRASTIVE CONTINUITY ON AUGMENTATION STABILITY REHEARSAL FOR CONTINUAL SELF-SUPERVISED LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Self-supervised learning has attracted a lot of attention recently, which is able to learn powerful representations without any manual annotations. In order to cope with a variety of real-world scenarios, it also needs to develop the ability to continuously learn, i.e. Continual Self-Supervised Learning (CSSL). However, simple rehearsal or regularization will bring some negative effects while alleviating catastrophic forgetting in CSSL, e.g. overfitting on the rehearsal samples or hindering 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 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^2 ASR) based on ASR, which preserves as much information shared among seen task streams as possible to prevent catastrophic forgetting and dismisses the redundant information to free up the ability to learn fresh knowledge. Our method obtains a great achievement compared with state-of-the-art CSSL methods on a variety of CSSL benchmarks. The source code will be released soon.

1 INTRODUCTION

Recently, self-supervised learning, or unsupervised visual representation learning, has received much attention from the community due to its great potential Chen et al. (2020a); He et al. (2020); Grill et al. (2020); Caron et al. (2020); Chen & He (2021); Zbontar et al. (2021). Self-supervised learning is able to learn representations that are beneficial to a variety of downstream tasks without any manual annotations. However, data is often presented as streams over time in real-world scenarios. It's nearly infeasible for self-supervised learning to collect the whole data streams to train the networks every time, since the ever-increasing data amount makes the notoriously costly training of self-supervised learning models even more expensive and sometimes the previous data streams are not able to access at all. Thus, self-supervised learning must develop the ability to continuously learn to cope with a variety of real-world scenarios, which is also called Continual Self-Supervised Learning (CSSL) in Fini et al. (2022).

Catastrophic forgetting is a notorious problem in continual learning, where many methods Rusu et al. (2016); Kirkpatrick et al. (2017); Zenke et al. (2017); Li & Hoiem (2017); Ahn et al. (2019); Buzzega et al. (2020) are proposed to alleviate it. CSSL also suffers from catastrophic forgetting, and some pioneers start to address this problem. Rehearsal-based method LUMP Madaan et al. (2022) utilizes rehearsal samples to augment current task samples by mixup Zhang et al. (2018), and regularization-based method CaSSLe Fini et al. (2022) encourages current model to maintain a consistency with previous state via a predictive head while training on current task samples. However, LUMP which is based on random sampling strategy for rehearsal tends to overfit on the rehearsal samples due to the long training epochs of self-supervised learning, and CaSSLe introduces too much invariance among task streams, which preserves most information of previous tasks 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 estimating the augmentation stability for rehearsal. Specifically, ASR aims to select the most representative and discriminative samples, i.e. the samples which are located at the center and the boundary of each category distribution, since they are able to retain the most information of previous tasks to overcome catastrophic forgetting as well as alleviating the overfitting effect. However, we are not able to obtain the relative position of the sample in corresponding category distribution under unsupervised scenarios, since we cannot access to the class label. Instead, we find the augmentation stability of the 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. we sample 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 buffer. Meanwhile, since the traditional queue and stack update cannot meet the requirement that retains the most representative and discriminative samples, we develop a matching strategy for ASR to dynamically update the rehearsal buffer.

Generally, current network needs to encode the information of previous tasks to alleviate catastrophic forgetting, as well as the information of current task to learn fresh knowledge. However, the whole information of previous states is not only redundant for preserving the memory of previous tasks Kang et al. (2022), but also hinders the learning on current task. In order 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) based on ASR. Inspired by the Information Bottleneck (IB) principle Tishby & Zaslavsky (2015); Tishby et al. (2000), C^{2}ASR expects current model to preserve as much information shared among seen task streams as possible to prevent catastrophic forgetting, and to dismiss the redundant information to free up the ability to learn fresh knowledge. In practice, C^{2}ASR encourages current model to be consistent with the previous states on the rehearsal samples to encode as much information as possible which is shared among seen task streams, and to be inconsistent with the previous states on current task samples to dismiss the redundant information. In addition, we incorporate the augmentation invariance and symmetrization strategy Grill et al. (2020); Chen & He (2021) 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 datasets. Our method achieves the best performance on most evaluation metrics compared with the state-of-the-art CSSL methods. In general, our contributions can be summarized as follows:

- We propose Augmentation Stability Rehearsal (ASR), which aims to store the most representative and discriminative samples for rehearsal, to address catastrophic forgetting without overfitting on the rehearsal samples. Meanwhile, we design a matching strategy for ASR to dynamically update the rehearsal buffer.
- We further propose Contrastive Continuity on Augmentation Stability Rehearsal (C^{2}ASR) based on ASR, which aims to preserve the information shared among seen task streams and dismiss the redundant information, to balance catastrophic forgetting and the learning on current task. In addition, we incorporate the augmentation invariance and symmetrization strategy with C^{2}ASR to further increase the diversity and stability of contrastive continuity pairs.
- The proposed method achieves significant improvements compared with the state-of-the-art CSSL methods on several popular CSSL benchmarks, showing strong competitiveness.

2 Related work

2.1 Continual learning

Continual learning aims to learn from a sequence of task streams without forgetting what has been learned in previous tasks. 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 Kirkpatrick et al. (2017) alleviates catastrophic forgetting by slowing down learning on the weights which are important to previous tasks during training. SI Zenke et al. (2017) introduces the synapses to track the parameter value of previous tasks, and fix the important synapses to keep the memories of the past. LwF Li & Hoiem (2017) utilizes distillation to make the output of current network approach to that of previous networks. Based on network quantization and pruning, piggyback Mallya et al. (2018) learns binary masks to selectively mask the weights of the backbone network, and achieves better performance on new tasks. UCL Ahn et al. (2019) 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 to meet future learning requirements. PNN Rusu et al. (2016) introduces progressive networks to alleviate catastrophic forgetting and designs lateral connections to use learned knowledge to assist current learning. DEN Yoon et al. (2018) dynamically expands the capacity of the network according to each task, so as to effectively capture the shared knowledge among tasks and prevent forgetting. Utilizing architecture search, Li et al. (2019) 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. RWalk Chaudhry et al. (2018) constructs the training set with the current task data and the rehearsal data, and uses distillation to make the model further strengthen the learned knowledge. Castro et al. (2018) stores some representative samples from previous tasks to alleviate intransigence. Based on the constrained optimization view of continual learning, Aljundi et al. (2019) store the samples which best approximate the feasible region defined by the original constraints. DER Buzzega et al. (2020) utilizes distillation to match the output logits on the rehearsal data, thus preserving the memory of previous tasks.

In addition, some works focus on representation continual learning, which aims to prevent forgetting the learned representation and utilizes it for future learning. iCaRL Rebuffi et al. (2017) utilizes distillation to learn an anti-forgetting representation. Meta-learning-based approaches OML Javed & White (2019) and La-MAML Gupta et al. (2020) learn representations by designing special meta-objectives that prevent catastrophic forgetting and promote future learning. Inspired by contrastive learning, Co^{2}L Cha et al. (2021) designs a supervised contrastive loss to learn a representation which is of nature resistance to catastrophic forgetting. LUMP Madaan et al. (2022) and CaSSLe Fini et al. (2022) aim to learn continual unsupervised representations, where LUMP uses mixup Zhang et al. (2018) to merge the samples in 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 to alleviate catastrophic forgetting.

2.2 Unsupervised representation learning

Unsupervised representation learning, or 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 Larsson et al. (2016), Inpainting Pathak et al. (2016), Jigsaw Noroozi & Favaro (2016), Rotate prediction Gidaris et al. (2018), etc. Contrastive learning based on instance discrimination Wu et al. (2018) 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 Chen et al. (2020a) and MoCo He et al. (2020) 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 Grill et al. (2020) 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 Chen & He (2021) 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. SwAV Caron et al. (2020) incorporates clustering into contrastive learning, which obtains pseudo label assignments via online clustering and constrains different augmented views of the same image to share the same assignment. NNCLR Dwibedi et al. (2021) finds some nearest neighbors in the feature space to serve as complementary positive samples, providing more semantic variations for standard data augmentations. DINO Caron et al. (2021) deploys self-supervised learning to ViT Dosovitskiy et al. (2021) and gets better results. Instead of applying stop-gradient operation to avoid collapsed solutions, Barlow Twins Zbontar et al. (2021) constrains the cross-correlation matrix between the features of two augmented views to be the identity matrix, achieving the same results.

3 METHOD

3.1 AUGMENTATION STABILITY REHEARSAL (ASR)

Generally, we should choose the most representative and discriminative samples for rehearsal, i.e. the samples which are located at the center and the boundary of each 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.

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 the samples in unsupervised scenarios. Meanwhile, it would take a huge computational cost to compute the similarity between the pairwise samples (with a computational complexity $O(N^2)$), which makes the algorithm infeasible to perform on a large scale data. Fortunately, we find the augmentation stability of each sample is positively correlated with its relative position in corresponding category distribution. Thus, we estimate the relative position distribution by utilizing the augmentation stability, and sample corresponding exemplars from the distribution to fill the rehearsal buffer.

Specifically, 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 $\mathcal{L}_D$ using the classical idea of contrastive learning, i.e.

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

where Z^1 and Z^2 are the pairwise augmentation features encoded by self-supervised model $f(\cdot)$, i.e. $Z^1 = f(\mathcal{T}^1(x))$ and $Z^2 = f(\mathcal{T}^2(x))$, $\mathcal{T}(\cdot)$ is the standard augmentation strategy in self-supervised learning where the augmentation pairs are distinguished by different right superscripts; $\bar{Z}^2$ is the augmentation feature from another image; $Concat(\cdot)$ denotes the cascade operation; $D(\cdot)$ denotes the discriminator; $CE(\cdot)$ denotes the *cross entropy loss*. The combined augmentation feature from one image $Concat(Z^1, Z^2)$ is classified as class 0, while the combined augmentation feature from different images $Concat(Z^1, \bar{Z}^2)$ 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.:

$$p(y = \text{“0”}|x) = \mathbb{E}_{(Z^1,Z^2)}\left[p_D(y = \text{“0”}|(Z^1, Z^2))\right] \tag{2}$$

However, $\mathbb{E}_{(Z^1,Z^2)}$ is almost infeasible to be calculated in practice. We approximate it by randomly sampling the augmentation distribution:

$$\begin{aligned} p(y = \text{“0”}|x) &= \int_{Z^1}\int_{Z^2} p_D(y = \text{“0”}|(Z^1, Z^2)) d_{Z^1} d_{Z^2} \\ &\approx \sum_{i=1}^{m}\sum_{j=1}^{m} p_D(y = \text{“0”}|(Z_i^1, Z_j^2)) \end{aligned} \tag{3}$$

where Z_i^1 and Z_j^2 is the sampling from corresponding augmentation distribution, i.e. $Z_i^1 \sim Z^1$ and $Z_j^2 \sim Z^2$; m is the sampling number which is set to 20 in practice; $p_D(y = \text{“0”}|(Z_i^1, Z_j^2))$ is the prediction probability that the discriminator classifies $Concat(Z_i^1, Z_j^2)$ as class 0. 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. In addition, 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 tasks when storing current data stream. Then, we discard the excess samples which are located in the middle of the augmentation stability sort for previous tasks (i.e. the least representative or discriminative ones) and load the selected samples of current data stream. It is worth noting that we select samples uniformly in the sorted list when storing current data stream, which ensures to contain the representative and discriminative samples as well as increasing the overall diversity. We give the specific update process in Algorithm 1.

Algorithm 1 ASR Update Algorithm

```
Input: Buffer size: $K$, data stream of task $t$: $D_t$, existing data in the buffer: $B_{t-1}$,
$B_t$ = { }
$k_t$ = $\lfloor K/t \rfloor$
for $i$ = 1 to $t-1$ do
    $B^i_{t-1}$ = $\{(x, task\_id) | task\_id = i, (x, task\_id) \in B_{t-1}\}$
    $B_t$ += $B^i_{t-1}[0 : \lfloor k_t/2 \rfloor]$ + $B^i_{t-1}[\left|B^i_{t-1}\right| - (k_t - \lfloor k_t/2 \rfloor) : \left|B^i_{t-1}\right|]$
end for
Sort $D_t$ by the augmentation stability computed by (3)
for $j$ = 1 to $K - k_t * (t-1)$ do
    $B_t$ += $D_t[j * \lfloor |D_t| / (K - k_t * (t-1)) \rfloor]$
end for
Output: Updated buffer $B_t$ after task $t$
```

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

In practice, continual self-supervised model requires to encode the information of previous tasks to prevent catastrophic forgetting, as well as encoding the information of current task 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 tasks by maintaining a consistency with previous learned models via knowledge distillation Hinton et al. (2015). However, the whole information of previous tasks is not only redundant for preventing catastrophic forgetting Kang et al. (2022), but also hinders the model from learning on current task. In order to dismiss the redundant information of previous tasks to balance the prevention of catastrophic forgetting and the learning on current task, we further propose Contrastive Continuity on Augmentation Stability Rehearsal inspired by the Information Bottleneck (IB) principle Tishby & Zaslavsky (2015); Tishby et al. (2000) in this subsection, which aims to preserve as much information shared among seen task streams as possible to prevent catastrophic forgetting and dismiss the redundant information to free up the ability to learn fresh knowledge.

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:

$$IB = I\left(Z; X\right) - \beta I\left(Z; Y\right) \tag{4}$$

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

Inspired by IB principle, C^{2}ASR expects current model to encode as much information shared among seen task streams as possible to prevent catastrophic forgetting, and to dismiss the redundant information to free up the ability to continuously learn. Specifically, given current date stream D_t and corresponding buffer B_{t-1} where we denote the data of task $\tau(\tau = 1, ..., t-1)$ in the buffer by B^{τ}_{t-1}, C^{2}ASR encourages current model $f_t(\cdot)$ to be consistent with previous states $f_{\tau}(\cdot)$ on corresponding rehearsal samples B^{τ}_{t-1} to capture the shared information among seen task streams, and to be inconsistent with previous states $f_{\tau}(\cdot)$ on current task samples D_t to dismiss the redundant information:

$$\mathcal{L}_{C^2ASR} = \frac{1}{t-1}\sum_{\tau=1}^{t-1}\left(sim(h(f_t(D_t)), f_{\tau}(D_t)) - \beta\left(sim(h(f_t(B^{\tau}_{t-1})), f_{\tau}(B^{\tau}_{t-1}))\right)\right) \tag{5}$$

where $sim\,(\,,)$ denotes the similarity measurement between two features (we apply the normalized cosine similarity in practice, i.e. $sim(a, b) = a \cdot b/(\|a\| \cdot \|b\|)$), $h(\cdot)$ is a projector which projects the representations of current model to previous feature space Fini et al. (2022), and β is a hyperparameter to trade off the amount of the preserved information shared among seen task streams and the eliminated information in previous states, where we set a cosine warmup mechanism for β since the proportion of the encoded shared information is increasing during training.

Obviously, $\mathcal{L}_{C^2ASR}$ starts working when $t > 1$ and there is an imbalance between current task samples D_t and rehearsal samples B^{τ}_{t-1}, i.e. $|D_t| \gg |B^{\tau}_{t-1}|$. Thus, we sample a subset D^{τ}_t from D_t ($|D^{\tau}_t| = |B^{\tau}_{t-1}|$) to address the imbalance problem, as well as reducing the computational complexity.

$$\mathcal{L}_{C^2ASR} = \frac{1}{t-1}\sum_{\tau=1}^{t-1}\left(\Gamma(D^{\tau}_t, D^{\tau}_t) - \beta\Gamma(B^{\tau}_{t-1}, B^{\tau}_{t-1})\right), \Gamma(x, y) = sim(h(f_t(x)), f_\tau(y)) \tag{6}$$

Meanwhile, we combine C^{2}ASR with augmentation invariance to increase the diversity of contrastive continuity pairs:

$$\mathcal{L}_{C^2ASR} = \frac{1}{t-1}\sum_{\tau=1}^{t-1}\left(\Gamma(\mathcal{T}^1(D^{\tau}_t), \mathcal{T}^2(D^{\tau}_t)) - \beta\Gamma(\mathcal{T}^1(B^{\tau}_{t-1}), \mathcal{T}^2(B^{\tau}_{t-1}))\right) \tag{7}$$

In addition, the symmetrization strategy Grill et al. (2020); Chen & He (2021) is applied to further increase the diversity, as well as reinforcing the stability:

$$\begin{aligned}\mathcal{L}_{C^2ASR} = \frac{1}{2(t-1)}\sum_{\tau=1}^{t-1}\Big(&\left(\Gamma(\mathcal{T}^1(D^{\tau}_t), \mathcal{T}^2(D^{\tau}_t)) + \Gamma(\mathcal{T}^2(D^{\tau}_t), \mathcal{T}^1(D^{\tau}_t))\right) - \\ &\beta\left(\Gamma(\mathcal{T}^1(B^{\tau}_{t-1}), \mathcal{T}^2(B^{\tau}_{t-1})) + \Gamma(\mathcal{T}^2(B^{\tau}_{t-1}), \mathcal{T}^1(B^{\tau}_{t-1}))\right)\Big)\end{aligned} \tag{8}$$

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Datasets. We deploy our experiments on Split CIFAR-10 Krizhevsky et al. (2009) (a 10-class dataset with 32×32 images), Split CIFAR-100 Krizhevsky et al. (2009) (a 100-class dataset with 32×32 images) and Split Tiny-ImageNet Deng et al. (2009) (a 100-class dataset with 64×64 images). We follow the division in Madaan et al. (2022) for the datasets, i.e. two random classes per task for CIFAR-10, five random classes per task for CIFAR-100 and Tiny-ImageNet.

Implementation details. We use ResNet-18 He et al. (2016) as the backbone and SimSiam Chen & He (2021) as the base self-supervised learning algorithm to make a fair comparison with existing methods. 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 Madaan et al. (2022) to utilize the KNN classifier Wu et al. (2018) to verify the quality of the learned representation, where ”Average Accuracy” and ”Average Forgetting” are served as the two key indicators.

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. In the long sequence datasets, e.g. Split CIFAR-100 and Split Tiny-ImageNet, we incorporates the mixup technique in LUMP into C^{2}ASR to further combat catastrophic forgetting. Compared with the existing continual self-supervised learning methods, our C^{2}ASR achieves the best performance on most 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 0.15%, 0.85%, 0.19% average forgetting drops on Split CIFAR-10, Split CIFAR-100 and

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 Wu et al. (2018). CaSSLe* is our reproduced version in this experimental settings according to original paper. 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 Rusu et al. (2016)	82.74(±2.12)	-	66.05(±0.86)	-	64.38(±0.92)	-
SI Zenke et al. (2017)	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 Chaudhry et al. (2019)	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 Aljundi et al. (2019)	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 Buzzega et al. (2020)	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 Rusu et al. (2016)	90.93(±0.22)	-	66.58(±1.00)	-	62.15(±1.35)	-
SI Zenke et al. (2017)	**92.75**(±0.06)	**1.81**(±0.21)	80.08(±1.30)	5.54(±1.30)	72.34(±0.42)	8.26(±0.64)
DER Buzzega et al. (2020)	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 Madaan et al. (2022)	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* Fini et al. (2022)	91.23(±0.34)	2.74(±0.39)	82.04(±1.17)	3.07(±1.54)	77.01(±2.11)	3.27(±0.62)
C^2ASR(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)	-

Split Tiny-ImageNet compared with LUMP and CaSSLe respectively. On the other hand, C^2ASR frees up the ability to continuously learn on new tasks, e.g. it obtains 1.47%, 0.82%, 1.19% and 1.24%, 1.08%, 0.84% average accuracy improvements on Split CIFAR-10, Split CIFAR-100 and Split Tiny-ImageNet compared with LUMP and CaSSLe respectively.

Table 2: The average accuracy on out of distribution 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 Wu et al. (2018) on out of distribution datasets i.e. MNIST LeCun (1998), Fashion-MNIST (FMNIST) Xiao et al. (2017), SVHN Netzer et al. (2011), CIFAR-100 or CIFAR-10. CaSSLe* is our reproduced version in this experimental settings according to original paper. 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 Zenke et al. (2017)	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 Chaudhry et al. (2019)	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 Aljundi et al. (2019)	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 Buzzega et al. (2020)	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)
SI Zenke et al. (2017)	**93.72**(±0.58)	**82.50**(±0.51)	**57.88**(±0.16)	36.21(±0.69)	91.50(±1.26)	80.57(±0.93)	54.07(±2.73)	60.55(±2.54)
DER Buzzega et al. (2020)	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 Madaan et al. (2022)	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* Fini et al. (2022)	90.88(±0.36)	80.85(±0.74)	53.17(±0.96)	37.44(±1.33)	91.29(±1.18)	81.32(±0.79)	52.08(±1.47)	66.35(±1.32)
C^2ASR(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)

4.3 Evaluation on OOD datasets

In this subsection, we report the average accuracy of the proposed C^{2}ASR on out of distribution datasets, where we recognise MNIST LeCun (1998), Fashion-MNIST (FMNIST) Xiao et al. (2017), SVHN Netzer et al. (2011), CIFAR-100 and MNIST LeCun (1998), Fashion-MNIST (FMNIST) Xiao et al. (2017), SVHN Netzer et al. (2011), 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 multiple evaluation metrics compared with the existing continual self-supervised learning methods, e.g. Split CIFAR-10 → CIFAR-100 and Split CIFAR-100 → (MNIST, FMNIST, SVHN, CIFAR-10), showing the learned representation by C^{2}ASR can be easily and effectively applied to unseen data distributions. SI obtains surprising results on Split CIFAR-10 → (MNIST, FMNIST, SVHN), whose performance gains largely come from the especially low forgetting on Split CIFAR-10, and C^{2}ASR becomes second only to SI on these evaluation metrics.

4.4 The results of collaboration with existing popular self-supervised learning methods

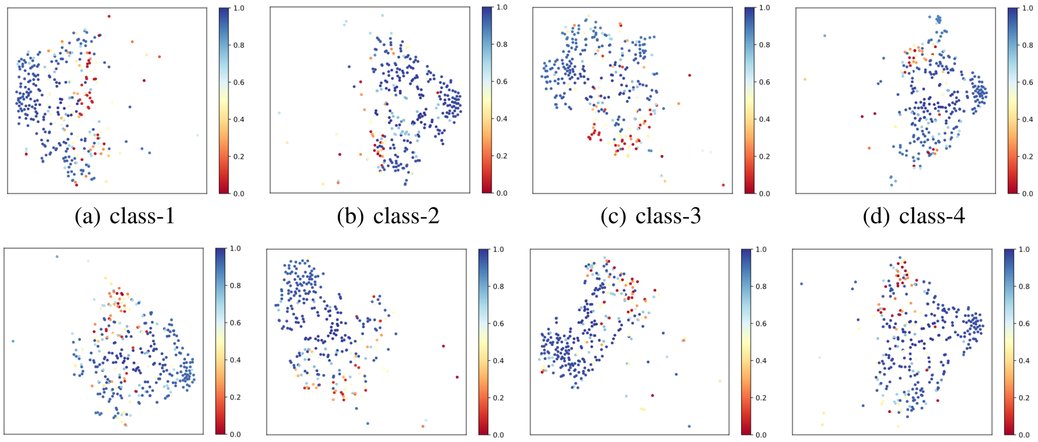
We give the average accuracy and average forgetting of the proposed C^{2}ASR collaborated with existing popular self-supervised learning methods on Split CIFAR-10, e.g. MoCo v2 Chen et al. (2020b), BYOL Grill et al. (2020), BarlowTwins Zbontar et al. (2021), as shown in Table 3. Our C^{2}ASR always achieves better results than existing CSSL methods, which shows C^{2}ASR can be well integrated with other self-supervised methods.

Table 3: The results (Average Accuracy and Average Forgetting) of collaboration with existing popular self-supervised learning methods on Split CIFAR-10. All methods are pre-trained with Resnet-18 as backbone for 200 epoches on Split CIFAR-10 and evaluated with KNN classifier Wu et al. (2018). 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.

	MoCo v2 Chen et al. (2020b)		BYOL Grill et al. (2020)	
	Accuracy	Forgetting	Accuracy	Forgetting
FINETUNE	89.27(±0.51)	4.70(±0.81)	88.49(±0.52)	4.93(±0.77)
LUMP Madaan et al. (2022)	91.56(±0.25)	2.24(±0.29)	91.14(±0.48)	2.61(±0.37)
CaSSLe* Fini et al. (2022)	91.31(±0.36)	1.91(±0.42)	91.44(±0.57)	2.57(±0.30)
C^{2}ASR(Ours)	**92.07**(±0.28)	**1.73**(±0.34)	**91.93**(±0.40)	**2.25**(±0.46)
	SimSiam Chen & He (2021)		BarlowTwins Zbontar et al. (2021)	
	Accuracy	Forgetting	Accuracy	Forgetting
FINETUNE	90.11(±0.12)	5.42(±0.08)	87.72(±0.32)	4.08(±0.56)
LUMP Madaan et al. (2022)	91.00(±0.40)	2.92(±0.53)	90.31(±0.30)	1.13(±0.18)
CaSSLe* Fini et al. (2022)	91.23(±0.34)	2.74(±0.39)	90.91(±0.23)	1.35(±0.38)
C^{2}ASR(Ours)	**92.47**(±0.41)	**2.59**(±0.58)	**91.34**(±0.26)	**0.94**(±0.22)

4.5 Ablation study and visualization

The visualization of the augmentation stability: In this part, we give the t-SNE visualization of the pre-trained representations combined with corresponding augmentation stability on Split CIFAR-10, as shown in Figure 1. We only show the categories in the first four tasks, i.e. task 1 (Figure 1(a), 1(b)), task 2 (Figure 1(c), 1(d)), task 3 (Figure 1(e), 1(f)) and task 4 (Figure 1(g), 1(h)), since the last task doesn't need to be replayed. We can see that the samples located in the center of category distribution often have a large augmentation stability value, while the samples located in the boundary of task distribution are low. This phenomenon is common in all tasks, and becomes the initial motivation of our ASR. The underlying mechanism is that self-supervised learning can learn semantically informative representations by encouraging augmentation invariance, even without manual annotations. Thus, this augmentation stability distribution is also encoded into the feature space by self-supervised models.



(a) class-1 (b) class-2 (c) class-3 (d) class-4

(e) class-5 (f) class-6 (g) class-7 (h) class-8

Figure 1: The t-SNE visualization of the pre-trained representations combined with corresponding augmentation stability on Split CIFAR-10. The color bar on the right corresponds the value of the augmentation stability.

The accuracy maps across the task streams: In this part, we report the accuracy maps across the task streams on Split CIFAR-10 in Table 4. Specifically, the accuracy map includes the knn accuracies on all seen tasks after training each task. To allow for simplification, we denote "the accuracy by training on task i and testing on task j" by "Tr_iTe_j $(i \geqslant j)$", i.e. corresponding to the row i and column j in each accuracy map; "the forgetting on task i" by "F_j", i.e. "$\underset{i}{max}\ Tr_iTe_j$ - Tr_5Te_j". Compared with FINETUNE, LUMP and Cassle have alleviated catastrophic forgetting. However, LUMP suffers from the overfitting effect, which reaches 0.84%, 5.32%, 4.79% and 1.91% on F_1, F_2, F_3 and F_4, while Cassle acquires 1.52%, 4.25%, 2.2% and 2.56% and the proposed C^{2}ASR acquires 1.61%, 4.25%, 2.45% and 1.99%, showing a better anti-forgetting ability. In terms of the ability to continuously learn, LUMP drops 3.33%, 1.86%, 2.05% and 0.85% on Tr_2Te_2, Tr_3Te_3, Tr_4Te_4 and Tr_5Te_5 compared with FINETUNE, and Cassle drops 4.23%, 2.16%, 1.68% and 0.27%, while the proposed C^{2}ASR drops 1.03%, 0.71%, 1.21% and 0.07%, showing strong competitiveness.

Table 4: The accuracy maps across the task streams on Split CIFAR-10.

FINETUNE					LUMP Madaan et al. (2022)					CaSSLe* Fini et al. (2022)					C^{2}ASR(Ours)				
1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
96.97	89.17	89.16	88.84	93.32	96.32	95.74	95.95	95.77	95.48	96.91	95.93	95.68	95.65	95.39	96.88	95.74	95.68	95.16	95.27
-	89.94	83.83	82.70	82.45	-	86.61	83.37	82.49	81.29	-	85.71	84.42	83.00	81.46	-	88.91	86.33	84.49	84.66
-	-	94.68	90.31	87.57	-	-	92.82	90.25	88.03	-	-	93.52	91.25	90.32	-	-	93.97	90.61	91.52
-	-	-	97.51	92.60	-	-	-	95.46	93.55	-	-	-	95.83	93.27	-	-	-	96.30	94.31
-	-	-	-	97.25	-	-	-	-	96.40	-	-	-	-	96.98	-	-	-	-	97.18

5 CONCLUSION

In this paper, we study how to address catastrophic forgetting in Continual Self-Supervised Learning (CSSL) without bringing some negative effects, e.g. overfitting on the rehearsal samples or hindering from learning fresh knowledge. Specifically, we first propose Augmentation Stability Rehearsal (ASR) store the most representative and discriminative samples for rehearsal, which helps to overcome the overfitting on the rehearsal samples. Secondly, we further propose Contrastive Continuity on Augmentation Stability Rehearsal (C^{2}ASR) based on ASR to preserve the information shared among seen task streams and dismiss the redundant information in previous states, which helps to free up the ability to continuously learn. We expect the contributions to be helpful for the CSSL community.

References

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

Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. In Advances in Neural Information Processing Systems, pp. 11816–11825, Vancouver, BC, Canada, December 2019.

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, pp. 15920–15930, virtual, December 2020.

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, pp. 9912–9924, Virtual, December 2020.

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, pp. 9650–9660, Virtual, October 2021.

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, pp. 233–248, Munich, Germany, September 2018.

Hyuntak Cha, Jaeho Lee, and Jinwoo Shin. Co2l: Contrastive continual learning. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9516–9525, Virtual, October 2021.

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, pp. 532–547, Munich, Germany, September 2018.

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.

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, pp. 1597–1607, Virtual, July 2020a.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 15750–15758, Virtual, June 2021.

Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. arXiv:2003.04297, 2020b.

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, pp. 248–255, Miami, Florida, USA, June 2009.

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.

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 International Conference on Computer Vision, pp. 9588–9597, Virtual, October 2021.

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, pp. 9621–9630, Virtual, June 2022.

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.

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, pp. 21271–21284, Virtual, December 2020.

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

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, pp. 770–778, Las Vegas, NV, USA, June 2016.

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, pp. 9729–9738, Virtual, June 2020.

Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. arXiv:1503.02531, 2015.

Khurram Javed and Martha White. Meta-learning representations for continual learning. In Advances in Neural Information Processing Systems, pp. 1818–1828, Vancouver, BC, Canada, December 2019.

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, pp. 16071–16080, Virtual, June 2022.

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.

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

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

Yann LeCun. The mnist database of handwritten digits. `http://yann.lecun.com/exdb/mnist/`, 1998.

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, pp. 3925–3934, Long Beach, California, USA, June 2019.

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

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.

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, pp. 72–88, Munich, Germany, September 2018.

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

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

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, pp. 2536–2544, Las Vegas, NV, USA, June 2016.

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, pp. 2001–2010, Honolulu, HI, USA, July 2017.

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.

Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In Proceedings of IEEE Information Theory Workshop (ITW), pp. 1–5, Jerusalem, Israel, April 2015.

Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. arXiv preprint physics/0004057, 2000.

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, pp. 3733–3742, Salt Lake City, UT, USA, June 2018.

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

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.

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, pp. 12310–12320, Virtual, July 2021.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In Proceedings of the International Conference on Machine Learning, pp. 3987–3995, Sydney, Australia, August 2017.

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.