Mitigating Forgetting in Continual Learning via Contrasting Semantically Distinct Augmentations

Sheng-Feng Yu[†‡] Wei-Chen Chiu[†]
[†]*National Yang Ming Chiao Tung University, Taiwan* [‡]*Macronix International Co., Ltd.*
robertyu1@mxic.com.tw [‡]walon@nycu.edu.tw

***Abstract*— Online continual learning (OCL) aims to enable model learning from a non-stationary data stream to continuously acquire new knowledge as well as retain the learnt one. Under the constraints of having limited system size and computational cost, in which the main challenge comes from the "catastrophic forgetting" issue – the inability to well remember the learnt knowledge while learning the new ones. With the specific focus on the class-incremental OCL scenario, i.e. OCL for classification, the recent advance incorporates the contrastive learning technologies for learning more generalised feature representation to achieve the state-of-the-art performance but is still unable to fully resolve the catastrophic forgetting. In this paper, we follow the strategy of adopting contrastive learning but further introduce the *semantically distinct augmentation* technique, in which it leverages strong augmentation to generate more data samples, and we show that considering these samples semantically different from their original classes (thus being related to the out-of-distribution samples) in the contrastive learning mechanisms contributes to alleviate forgetting and facilitate model stability. Moreover, in addition to contrastive learning, the typical classification mechanism and objective (i.e. softmax classifier and cross-entropy loss) are included in our model design for utilising the label information, but particularly equipped with a sampling strategy to tackle the tendency of favouring the new classes (i.e. model bias towards the recently learnt classes). Upon conducting extensive experiments on CIFAR-10, CIFAR-100, and Mini-Imagenet datasets, our proposed method is shown to achieve superior performance against various baselines.**

I. Introduction

The ability to continually learning new knowledge is getting more and more important for machine learning models nowadays as the increasing demands of automation and the dynamic nature of our environment, e.g. the visual recognition system of the goods in the intelligent self-checkout system for smart retail should be able to classify the newly-added items or the existing items with new packing. In particular, the model cannot be trained from scratch whenever the classes or recognition targets increase. Instead, it needs to keep adapting itself to learn new knowledge on the fly over time. *Online continual learning* (OCL) [1] is one of the topics getting popular these years to serve such purpose, where the machine learning agent continually learns a few new concepts every once in awhile without forgetting the others (i.e. what the agent has learnt previously).

If the agent continually learns to classify a new set of unseen classes, this problem is named *class-incremental OCL*. It is one of the most prevalent settings in the community of OCL. However, learning on unseen classes would change the model parameters (and the feature representation space) optimised for the old classes. Hence, the model classification accuracy on the old classes inevitably deteriorates. This phenomenon is well-known and called *Catastrophic Forgetting*.

To address this issue, [2] proposes using a small memory to store the learnt examples. When learning new classes, the model is retrained/updated by using not only the recently received samples that belong to new classes but the stored examples from old classes, in which such a strategy attempts to maintain the accuracy for both the old and new classes. Another widely-adopted idea comes from [3], where they propose to regularise the model learning by constraining the update of important parameters in order to alleviate the catastrophic forgetting. However, limiting the model update space deteriorates the learning ability of the model. Recently, the introduction of learning a generalisable representation [4], [5], [6] brings another break. Basically, learning the generalisable representation aims to not only distinguish between the learnt/seen classes but also have higher intrinsic dimension such that the features from unseen classes are more likely to be distributed away from the seen ones, i.e. the feature representations are richer and more transferable for the unseen classes in which the model requires relatively minor adjustments to achieve high accuracy for the new classes.

Among the three ideas described above, learning a generalisable representation has benefited from the recent advance of self-supervised representation learning [7], [8]. In particular, contrastive learning, e.g. SimCLR [9] and SimSiam [10], has shown its effectiveness in learning generalisable image representation. For instance, SimCLR leverages the composition of multiple carefully chosen data augmentations, such as cropping and colour distortion, to generate random views, and the model is trained to align these random views in the representation space with the ones of the same label while pushing away the views with different labels. In contrast, SimSiam is a technique that can learn meaningful representations without the need for separating views of different labels by using stop gradient operation and without the requirement for large batch size, making it suitable for online continual learning.

However, [9] observed that some augmentations (e.g. rotation, noise) are too strong to deteriorate the representation quality if they involve in the view generation, and [11] discovered that the semantic shift caused by these augmentations is too large to align the corresponding random views well. Despite this, it does not imply that strong aug-

mentations [7], [8] cannot provide meaningful semantics for representation learning. For example, [8] propose learning image representation by predicting rotation. Hence, those augmentations can still be utilised for contrastive representation learning, but we should be aware of not encouraging the model to align the views with those strong augmentations which could cause the large semantic shift. In turn, if the diverse views produced by the strong augmentation are treated as belonging to the other classes which are distinct from their original samples, additionally considering them in contrastive learning could help the model learn to extract rich features and represent the unseen classes better (according to the observation made by [12] where the diverse views together are similar to the auxiliary dataset and the unseen classes are analogous to the out-of-distribution samples). This research builds upon the aforementioned concept and aims to explore the potential of contrastive learning and image transformations in enhancing the performance of class incremental OCL. To achieve this, we propose *semantically distinct augmentation (SDA)*: given a mini-batch composed of the training samples, the strong transformations are first applied to these training samples to produce diverse views which are treated as from different/novel classes (cf. Figure 1) and are added back to the mini-batch, then both contrastive learning and softmax classifier are applied to the extended mini-batch (with having both the original samples and their corresponding diverse views). The two-pronged benefits are introduced by such SDA technique: First, learning on a diverse dataset allows the model to get a more generalisable representation and mitigates catastrophic forgetting; Second, as data comes in a stream for the OCL setting such that each data sample ideally can only be adopted once for training, our SDA leverages strong augmentations for the attempt on making best use of every sample.

Furthermore, as online continual learning has a non-stationary data stream, the model is likely to face the imbalanced training set (where the training samples are mostly from the newly added classes), thus the softmax classifier would suffer from the class-imbalanced problem [13]. We hence adopt a specifically-designed sampling strategy to balance the learning between old and new classes. With conducting experiments on several datasets and different settings of online continual learning, our full model equipped with all the aforementioned designs (named as **SDAF**) is demonstrated to provide the state-of-the-art performance in comparison to various baselines.

II. Related Work

Online Continual Learning. The goal of class-incremental online continual learning [14] focuses on how an artificially intelligent agent learns to classify new classes without forgetting its knowledge on the classes previously learnt (where such an issue is the so-called catastrophic forgetting). To tackle against the catastrophic forgetting, one should balance the model learning between the old classes and new classes, where the literature roughly contains three branches:

(1) ***Experience Replay***. [2] suggests that the online learning agent equips a fixed-sized memory to store the learnt examples, then the model repeatedly replays the samples from the memory to alleviate catastrophic forgetting. In particular, they adopt the reservoir sampling [15] strategy to draw the samples from the memory for model training, such strategy ensures the sampling result being equivalent to having uniform sampling from the stream data without knowing the sequence length. Follow-up works [16], [17], [18], [19], [20] assume that every training data has different importance, and remembering a few critical samples is enough for keeping the data distribution.

(2) ***Regularisation and Constraint Optimisation***. [3], [21], [22] regularise the network update to alleviate catastrophic forgetting during learning new classes, and these approaches are often efficient and usually have little extra cost. For instance, [23], [24] constrain the model optimisation such that the loss on past classes never increases. However, both works limit the space for the model optimisation, and hence they suffer from the inability of learning new classes.

(3) ***Improving Representation Learning***. Online aware meta-learning (OML) [25] uses a meta-learning objective to pre-train the sparse representation which easily adapts to new classes to mitigate catastrophic forgetting; Nevertheless, its contribution is focused on having better pre-trained weights, while the main target tackled in this paper is the overall continual learning mechanism thus being different. And other approaches instead aim to learn the generalisable representations [4], [6], [26]. As the generalisable representation ideally should provide better support for not only the seen classes but also the unseen ones, hence it requires fewer tuning to optimise for the new classes and suffers less from forgetting.

There are other approaches that are unable to categorise into the above three branches. First, knowledge distillation [27] keeps an old model as a teacher to preserve the learnt knowledge [28], [5]. Second, the expansion-based online continual algorithm dynamically expands the network capacity upon the arrival of new classes [29], [30]. We do not consider these approaches here, as they need additional resources for computation (e.g. much more additional memory for storing the old model to perform knowledge distillation; and continuously growing model size for the expansion-based methods in which it means that the requirement of memory space also keeps increasing), and we only consider the methods with similar computational cost as our baselines to make comparison. In this work, we focus on mitigating the forgetting issue by continually learning a generalisable representation. The learning system additionally equips a small memory for replay, and every sample only appears once in the entire training trajectory except that it is stored in the memory.

Image Representation Learning. Image representation learning is an essential foundation for various computer vision tasks. Especially, self-supervised learning is one of the most thrilling branches in this field. Self-supervised learning encourages the machine to learn image representation from a pretext task, which is able to automatically generate a su-

pervision signal via a predefined transformation without any human labelling. For example, image permutation and rotation prediction [7], [8] help the model learn the image feature representation. In particular, recent works on self-supervised learning advance to contrast between images to perform representation learning by leveraging the combination of various transformations, called contrastive learning [9], [10], [31]. For instance, SimCLR [9] as a representative work first generates a pair of positive views by applying a sequence of transformations to an image twice, then it learns image representation by attracting the positive pairs and pushing negative views from other images away. SimSiam [10] can be thought of as "SimCLR without negatives", it introduces a predictor network in its forward process on one view and applies a stop-gradient operation in its backward process on the other view. Moreover, as it uses neither the negative sample pairs nor the momentum encoders (what other self-supervised methods, e.g. MoCO [32], would need), it has smaller model size during training (compared to MoCO and BYOL) as well as better support for the small training batch. SimSiam thus becomes suitable for the computational-cost-sensitive problems such as OCL, the main topic of this paper. We argue that representation learning has a high potential to mitigate catastrophic forgetting by increasing the feature generalisability. And our method proposes to utilise a strong data augmentation to boost the feature generalisability learnt by the contrastive learning, which will be detailed later.

Long-Tailed Recognition. As mentioned in the introduction, we adopt a softmax classifier to facilitate the training efficiency, but it is susceptible to the non-stationary input order [33], [34]. Even with a replay buffer to store the exemplars from old classes, the softmax classifier still tends to bias towards the new classes. Long-tailed recognition [13], [35] is the subject that aims to balance the softmax classifier under an imbalanced training dataset. For instance, [13] solves this issue by normalising the classifier according to the weight norm in the softmax classifier. In this work, we balance the model by dynamically adjusting the data distribution for softmax classifier learning.

III. Methodology

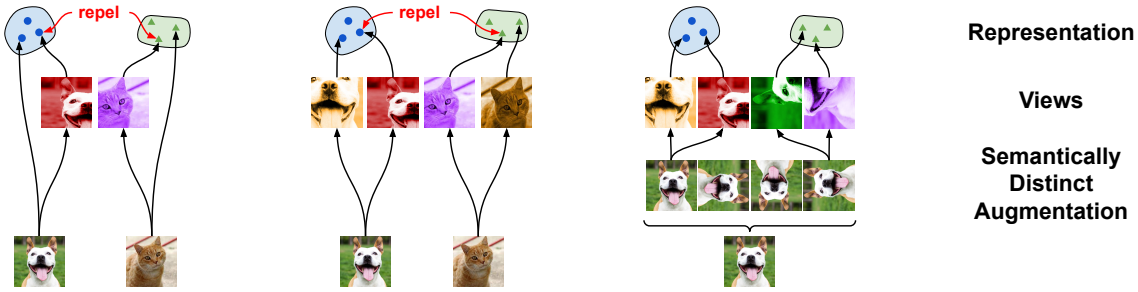
In this work, given a machine learning agent to execute the class-incremental OCL, we assume that it has a fixed size memory $\mathcal{M}$ to store the exemplars of the old/learnt classes (for the purpose of experience replay) and there are T training stages for the entire training process, where for each training stage $t \in \{1 \cdots T\}$ the agent will learn from the training examples that arrive as a data stream (in which it means that each training sample only appears once unless it is stored in the memory) for recognising a set of new classes $\mathcal{C}_t$. Please note that in the following paragraphs we would misuse $\mathcal{M}$ to represent the experience replay memory or its size for simplicity. Following a similar setting as in previous works, the classes learnt at each training stage are assumed to be disjoint for simplicity, i.e. $\mathcal{C}_i \cap \mathcal{C}_j = \phi$ for any $i \neq j$, and we denote the training samples received during the training stage t as $\mathcal{D}_t$ where they belong to the classes $\mathcal{C}_t$. As the target of class-incremental online continual learning at the training stage t is to let the machine learning agent not only learn the new classes $\mathcal{C}_t$ but also maintain its recognition ability for the old classes learnt during previous stages, the learning scenario at the training stage t is a $\{C_{\text{old}} + C_t\}$-ways classification problem where $C_{\text{old}} = \sum_{i=1}^{t-1} C_i$ and C_i denotes the cardinality of $\mathcal{C}_i$. Without loss of generality, we index the old classes $\mathcal{C}_{\text{old}}$ by $\{1, 2, \ldots, C_{\text{old}}\}$ and the new classes $\mathcal{C}_t$ by $\{C_{\text{old}} + 1, \ldots, C_{\text{old}} + C_t\}$. In detail, the data stream at the training stage t is composed of U data batches $\mathcal{B}_t^u$ where $u = 1 \cdots U$, and each batch $\mathcal{B}_t^u$ contains a group of training samples x_i and their class labels y_i where $y_i \in \{C_{\text{old}} + 1, \ldots, C_{\text{old}} + C_t\}$. These batches are disjoint, i.e. every training sample only appears among batches once during the data stream, this setting is called *one epoch setting*.

Without loss of generality, here we summarise a generic algorithmic procedure for the class-incremental OCL methods with the experience replay memory in Algorithm 1. Basically, when the machine learning agent receives a data batch B_t^u during the training stage t, it will use the data from B_t^u (belonging to the new classes) as well as the samples from the experience replay memory (mostly belonging to the old classes) to train itself for I iterations (for simplicity, we assume that I is a constant). Please note that, we just train the same sample for I times which does not violate the one epoch setting. In particular, at each iteration, the agent utilises the data $B_t^u \cup B_M$ to perform the training where B_M denotes the m samples retrieved from the experience replay memory. After I iterations of training based on the batch B_t^u and the memory, the MemoryUpdate operation is performed to replace some exemplars in the memory $\mathcal{M}$ with the ones sampled from B_t^u, where the reservoir sampling algorithm [15] is adopted for such a MemoryUpdate operation in our work following the practise in [4].

A. Preliminary

As motivated previously that our proposed method stems from the idea of learning a generalisable representation via contrastive learning, where a recent work [4] adopts such idea to achieve the state-of-the-art in class-incremental OCL, here we hence review several key references (e.g. SimCLR [9] and SCL [36]) for traversing the main ideas behind [4] in order to build the preliminary of our method.

First, SimCLR [9], as a representative approach of contrastive learning, consists of three parts: a random transformation module $\mathcal{H}$, an encoder network $\mathcal{F}$, and a projection network $\mathcal{G}$. Basically, the transformation module $\mathcal{H}$ adopts a sequence of random transformations (e.g. sequentially applying random crop, random horizontal flip, and random colour distortion) to generate different views $\tilde{x}_i^{(j)}$ for every image x_i in a training batch $\{x_i\}_{i=1\ldots\mathbb{B}}$, where $\mathbb{B}$ is the number of samples in a batch, $\tilde{x}_i^{(j)} = H^{(j)}(x_i)$, and the transformation operation $H^{(j)}$ is re-sampled from $\mathcal{H}$ for each x_i. Based on such random transformations, we construct a set of views $\tilde{x}_i^{(j)}$ together with their corresponding image



(a) SCR (b) SCL (c) SDA

Fig. 1: The comparison among **(a)** supervised contrastive replay (SCR), **(b)** supervised contrastive learning (SCL), and **(c)** our proposed SDA model. SCR contrasts the original image to a random view, and SCL contrasts between two random views. Our proposed SDA first augments an image into K instances in which each instance is treated as belonging to different classes, then every instance generates two views. In results, there are $2K$ views in total. Specifically, since the K instances augmented via SDA are now treated as K novel classes, the classification scenario here is analogous to a $\{K(C_{old}+C_t)\}$-ways classification problem. Noting that, for every representation on the top, the points with the same colour are encouraged to form a compact group during the learning. Moreover, the repulsion between different classes is only performed in SCR/SCL but not used in our SDA (as SDA adopts the contrastive mechanism from SimSiam).

index i:

$$\mathbf{V}=\bigcup_{i=1}^{\mathbb{B}}\bigcup_{j=1}^{2}\{(\tilde{x}_i^{(j)},i)\mid \tilde{x}_i^{(j)}=H^{(j)}(x_i)\} \tag{1}$$

With denoting $z_i^{(j)}=\mathcal{G}(\mathcal{F}(\tilde{x}_i^{(j)}))$, the goal of SimCLR training is to learn the feature encoder $\mathcal{F}$ via the objective of encouraging two vectors $z_i^{(j)}$ and $z_i^{(j')}$ obtained from the same x_i but under different transformations (i.e. positive pairs) to attract each other while enforcing the z vectors to repel once they originate from different images (i.e. negative pairs).

Algorithm 1 Generic class-incremental online continual learning algorithm with the experience replay memory

Input: Learning rate α; The number of iterations for SGD update I; Training objective $\mathcal{L}$
Parameter: θ

Memory $\mathcal{M}\leftarrow\{\}$
for $t=1$ to T **do**
  while $B_t^u\sim D_t$ **do**
    for $i=1$ to I **do**
      $B_M\leftarrow$ MemoryRetrieval$(\mathcal{M},m)$
      $\theta\leftarrow$ SGD$(B_t^u\cup B_M,\mathcal{L},\theta,\alpha)$
    end for
    $\mathcal{M}\leftarrow$ MemoryUpdate$(B_t^u,\mathcal{M})$
  end while
end for
return θ

In comparison to SimCLR which is self-supervised as the positive and negative pairs are simply determined by their image indexes, Supervised Contrastive Learning (SCL) [36] takes the class labels y into consideration thus being supervised, in which the training views are constructed by:

$$\mathbf{V}_{\text{SCL}}=\bigcup_{i=1}^{\mathbb{B}}\bigcup_{j=1}^{2}\{(\tilde{x}_i^{(j)},y_i)\mid \tilde{x}_i^{(j)}=H^{(j)}(x_i)\} \tag{2}$$

where the transformation module $\mathcal{H}$ here is the same as the one used in SimCLR. Similarly, given the training views $\mathbf{V}_{\text{SCL}}$, the objective of SCL is to encourage the attraction between z vectors from the same class and enforce the repulsion between the ones from different classes, for learning the extractor $\mathcal{F}$.

Supervised contrastive replay (SCR) [4] adapts SCL for the problem of class-incremental OCL and provides the state-of-the-art performance, where the main difference between SCR and SCL comes from the transformation module, as visualised in Figure 1a and Figure 1b respectively: For SCL, its transformation module $\mathcal{H}$ applies two distinct transformations on the input image to construct the positive pair, while a positive pair in SCR is built upon an input image x_i and its random view $\tilde{x}_i$, thus SCR in general has lower randomness than SCL. With denoting the original images as $\mathbf{V}_{\text{ori}}=\{(\tilde{x}_i,y_i)\mid\tilde{x}_i=x_i\}$, the training views for SCR are:

$$\mathbf{V}_{\text{SCR}}=\bigcup_{i=1}^{\mathbb{B}}(\{(\tilde{x}_i,y_i)\mid\tilde{x}_i=H(x_i)\}\cup\mathbf{V}_{\text{ori}}) \tag{3}$$

In particular, when SCR and SCL are both applied in the class-incremental OCL scenario, as SCR has lower randomness than SCL during the constructing training views, it is more likely to provide higher classification accuracy than SCL in the first few training stages; however, such lower randomness of SCR in turn sacrifices the potential for learning more diverse (thus more generalised) representations hence leading to lower accuracy of SCR compared to SCL in the later training stages.

B. Semantically Distinct Augmentation

The performance difference versus training stages caused by the aforementioned randomness between SCR and SCL

motivates us to conduct further research on the impact of the random transformations upon the representation learning. As found by the work of SimCLR [9], adding some particular transformations (e.g. rotation or blur) into the transformation module would instead hurt the quality of learnt representations as these transformations cause more significant distortion to the input image (thus having the semantic shift). Following such an empirical observation, we propose the mechanism named *Semantically Distinct Augmentation (SDA)* which is applied on the input image x before the transformation module $\mathcal{H}$. The SDA consists of multiple deterministic augmentations, and every augmentation would cause a distinct semantics change of the input images. In results, if SDA are adopted during the learning, the feature space tends to have higher intrinsic dimension (which leads to more generalised features) for handling the diverse semantics produced by SDA. Later in experiments, we demonstrate that using such an SDA mechanism benefits the OCL to learn more generalised representations, thus leading to superior performance.

In detail, we assume that there are K strong deterministic increases $\mathcal{S} = \{S_1, S_2, \ldots, S_K\}$ in the SDA mechanism, and every augmentation in $\mathcal{S}$ applies to each sample x_i, that is $S_k(x_i)$. Then, similarly to SCL, $H^{(j)}$ is used to generate random views, $\tilde{x}_{ik}^{(j)} = H^{(j)}(S_k(x_i))$, as visualised in Figure 1c, and the extended label space is defined by:

$$\tilde{y}_{ik} = K(y_i - 1) + k \tag{4}$$

where the original label y_i extends to K different classes. Based on such an extended label space, the batch of views for training is defined as:

$$\mathbf{V}_{\text{SDA}} = \bigcup_{i=1}^{\mathbb{B}} \bigcup_{j=1}^{2} \bigcup_{k=1}^{K} \{(\tilde{x}_{ik}^{(j)}, \tilde{y}_{ik})\} \tag{5}$$

We then adopt the contrastive learning mechanism of SimSiam [10] to peform the learning upon $\mathbf{V}_{\text{SDA}}$, where the loss function for each single view $z_{ik}^{(j)} = \mathcal{G}(\mathcal{F}(\tilde{x}_{ik}^{(j)}))$ is defined as follows to encourage the anchor view $\tilde{x}_{ik}^{(j)}$ being grouped up with its corresponding positive views:

$$\mathcal{L}_{\text{vw}}(z_{ik}^{(j)}) = -\sum_{j' \neq j} \text{CosineSimilarity}(\mathcal{P}(z_{ik}^{(j)}), \textit{stopgrad}(z_{ik}^{(j')})) \tag{6}$$

in which $\mathcal{P}$ is the predictor network and *stopgrad* denotes the stop-gradient operation. Finally, the self-supervised objective function averaged over $\mathcal{L}_{\text{vw}}$ of all views is adopted in each iteration:

$$\mathcal{L}_{\text{SS}} = \sum_{z_{ik}^{(j)} \in A} \mathcal{L}_{\text{vw}}(z_{ik}^{(j)}) \tag{7}$$

where $A = \{z_{ik}^{(j)} | z_{ik}^{(j)} = \mathcal{G}(\mathcal{F}(\tilde{x}_{ik}^{(j)})), \quad \forall(\tilde{x}_{ik}^{(j)}, \tilde{y}_{ik}) \in \mathbf{V}_{\text{SDA}}\}$ includes the features of all views in a batch.

Despite the contrastive-learning-based loss $\mathcal{L}_{\text{SS}}$, we also leverage the label information by including the softmax classifier and the cross-entropy loss $\mathcal{L}_{\text{CE}}$:

$$\mathcal{L}_{\text{CE}} = -\sum_i \sum_k \mathbf{1}(\tilde{y}_{ik})^T \log p_{ik}^{(j)} \tag{8}$$

where given a view $\tilde{x}_{ik}^{(j)}$, $p_{ik}^{(j)} = \text{softmax}(W^T \mathcal{F}(\tilde{x}_{ik}^{(j)}) + b)$ is a probability vector with length $K(C_{old} + C_t)$, W is a weight matrix, b is a bias vector, the one-hot vector $\mathbf{1}(\tilde{y}_{ik})$ has value 1 for the element indexed by $\tilde{y}_{ik}$ and zero everywhere else.

C. Weight-Aware Balanced Sampling

The softmax classifier is likely to be biased towards the classes with more training samples [13], [37], [38]. In the OCL scenario, the learning agent accesses more samples related to new classes because only a small fraction of the old examples are stored. Thus, the model tends to classify samples into new classes. To tackle such an issue, as every column of the weight matrix W for the softmax classifier represents the weights for the corresponding class (hence being related to the degree of bias), we propose weight-aware balanced sampling (WABS) which adaptively decides the sample ratio between old and new classes to balance the classifier based on degree of bias. We first define a sampling rate γ as follows:

$$\gamma = \min\left(1, \frac{2 \times \exp(w_{\text{old}}/\tau_w)}{\exp(w_{\text{old}}/\tau_w) + \exp(w_{\text{new}}/\tau_w)}\right) \tag{9}$$

where w_{new} is the mean over all the weights related to the new classes (i.e. average over the columns in W corresponding to the new classes) and w_{old} is defined similarly for the old classes, and τ_w is a hyperparameter.

Then we reformulate the cross-entropy loss as follows:

$$\mathcal{L}_{\text{WABS}} = -\sum_i \sum_k \mathbb{1}\!\!\!\!\mathbb{1}_{\text{WABS}}(\tilde{y}_{ik}) \mathbf{1}(\tilde{y}_{ik})^T \log p_{ik}^{(j)} \tag{10}$$

where $\mathbb{1}\!\!\!\!\mathbb{1}_{\text{WABS}}$ is defined as below, with uniformly drawing Γ from $[0, 1]$ for each sample:

$$\mathbb{1}\!\!\!\!\mathbb{1}_{\text{WABS}}(\tilde{y}_{ik}) = \begin{cases} 1, & \text{if } \tilde{y}_{ik} \leq KC_{\text{old}} \\ 1, & \text{if } \tilde{y}_{ik} > KC_{\text{old}},\ \Gamma < \gamma \\ 0, & \text{if } \tilde{y}_{ik} > KC_{\text{old}},\ \Gamma \geq \gamma \end{cases} \tag{11}$$

in which we keep all those views belonging to the old classes while for each of the views belonging to new classes it has γ probability to be kept, for the use in the cross-entropy loss. Please note that, the WABS only applies on $\mathcal{L}_{\text{WABS}}$ while $\mathcal{L}_{\text{SS}}$ uses all views without any sampling.

The overall objective for our full model (named **SDAF**) combines the proposed $\mathcal{L}_{\text{SS}}$ to reduce forgetting by learning on diverse views and the cross-entropy loss $\mathcal{L}_{\text{WABS}}$ with adaptive sampling to utilise the label information: $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{WABS}} + \lambda \mathcal{L}_{\text{SS}}$, where λ controls the balance between two losses.

D. Inference

We adopt nearest-centre-mean (NCM) classifier for inference. For any test sample x, we average its distance to all class centres over K augmentations. First, we calculate the centres m_{ck} for all $K(C_{old} + C_t)$ classes,

$$m_{ck} = \frac{1}{|R_{ck}|} \sum_{r_{ik} \in R_{ck}} r_{ik} \tag{12}$$

where $R_{ck} = \{r_{ik} = \mathcal{F}(S_k(x_i)) | y_i = c,\ (x_i, y_i) \in \mathcal{M}\}$. We define the prediction function for a test sample x as:

$$\hat{y} = \underset{c}{\operatorname{argmin}} \frac{1}{K} \sum_{k} d(\mathcal{F}(S_k(x)), m_{ck}) \tag{13}$$

The distance metric $d(x, m) = \sqrt{(x-m)^T \Sigma^{-1} (x-m)}$ is based on the Mahalanobis distance and $\Sigma^{-1} = \text{Cov}^{-1}(\mathbf{R})$ is the pseudo-inverse of the covariance matrix of the set $\mathbf{R} = \bigcup_{c \in \mathcal{C}_{\text{old}} \cup \mathcal{C}_t} \bigcup_{k=1}^{K} R_{ck}$. Noting that for previous methods (e.g. SCR) they typically adopt Euclidean distance for $d(x, m)$ in the nearest-centre-mean classifier. The reason behind our using Mahalanobis distance is that it takes the feature distribution into consideration via covariance matrix, while Euclidean distance only computes the distance from every individual sample to the mean of exemplars.

IV. Experimental Results

Datasets. We experiment on three benchmarks, including CIFAR-10, CIFAR-100, and Mini-ImageNet. We split CIFAR-10 into 5 incremental stages, and each stage contains 2 classes; We split CIFAR-100 into 10 incremental stages, and each stage contains 10 classes; We split Mini-ImageNet into 10 incremental stages and each stage contains 10 classes.
Metrics. We adopt several well-known metrics to assess the performance of online continual learning, including: average incremental accuracy ($\mathbb{A}$), end accuracy ($\mathbb{E}$), and forgetting measure ($\mathbb{F}$), where their definitions could be found in [22].
Architecture. The full architecture of our SDAF model is illustrated in Figure 2. First, the semantically distinct augmentation $\mathcal{S}$ generates K different images $S_k(x)$ from the input image x. Then, we sample $2K$ random transformations $H \sim \mathcal{H}$ to create $2K$ views for those K images, followed by using the feature extractor $\mathcal{F}$ to project the $2K$ images into the latent representation space. The network $\mathcal{G}$ further projects the resultant feature representation into another low-dimensional space to perform the contrastive learning. On the other hand, the softmax classifier $\mathcal{G}_{\text{soft}}$ is responsible for computing the cross-entropy loss, which is equipped with the weight-aware balanced sampling strategy ($\mathcal{L}_{\text{WABS}}$).
Implementation – Hyperparameters. The network architecture for the components used in our proposed method basically follows the ones in [4]. For all experiments, we adopt a reduced ResNet18 as our feature extractor $\mathcal{F}$ with resultant feature dimension set to 160, the projection head $\mathcal{G}$ is a two-layer multilayer perceptron (MLP) with width 160 and 128 respectively, and the predictor $\mathcal{P}$ is also a two-layer MLP with both input and output width being 128. The transformation module $\mathcal{H}$ of contrastive learning consists of random cropping, random horizontal flip, random colour distortion, and random grey scale. The detailed setting for transformation module $\mathcal{H}$ is described later in the next paragraph. We adopt the SGD optimiser with learning rate 0.1. The batch size $|B_t^u|$ is 10, and the retrieval batch size $|B_M|$ is 10. We adopt the rotation as the SDA strategy $\mathcal{S}$, where $\mathcal{S}$ consists of four different degree of rotation ($K = 4$), i.e. $0°, 90°, 180°$, and $270°$. We empirically set λ in the total loss function $\mathcal{L}_{\text{total}}$ to 1.5, and the temperature τ_w in our WABS (cf. Eq. 9) to 0.5. We use reservoir sampling and uniform random sampling for operations MemoryUpdate and MemoryRetrieval in Algorithm 1.
Implementation – Transformation Module. The transformation module $\mathcal{H}$ for contrastive learning composes of a long series of random transformations and another short series of random transformations, where each series is responsible for generating one view in a positive pair (following the common practice as used in FixMatch [39]). The long series of random transformations includes a uniformly random cropping, a random horizontal flip, a random colour distortion, and a random grey scale operator. Uniformly random cropping keeps the original image from 50% to 100% in terms of area, and it has been implemented in Pytorch as "torchvision.transforms.RandomResizedCrop". Random horizontal flip has 50% chance to flip the image. Random colour distortion has also been realized in Pytorch as "torchvision.transforms.ColorJitter", and we set the factors for brightness, contrast, saturation to 0.4, and the factor for hue is set to 0.1. Last, with a chance of 20%, the image is converted to gray scale, and this operator is implemented as "torchvision.transforms.RandomGrayscale" in Pytorch. On the other hand, the short series of random transformations comprises of a uniformly random cropping and a random colour distortion. The implementation is the same as the long series of random transformation, except the cropping area is set to 75% to 100%, the factors for brightness, contrast, saturation to 0.2, and the factor for hue is set to 0.05.

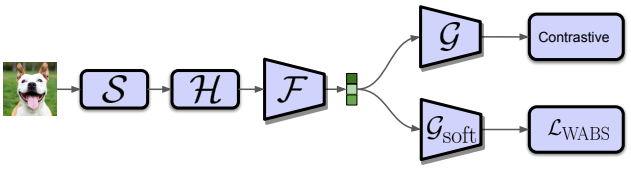


Fig. 2: Model architecture. $\mathcal{S}$ is semantically distinct augmentation. $\mathcal{H}$ is the transformation module for contrastive learning. $\mathcal{F}$ is a feature extractor. $\mathcal{G}$ is a multilayer perceptron (MLP). $\mathcal{G}_{\text{soft}}$ is a single-layer softmax classifier (i.e. one fully-connected layer).

Results in End Accuracy $\mathbb{E}$ and Average Incremental Accuracy $\mathbb{A}$. We compare our full model (SDAF) with respect to several state-of-the-art baselines, including EWC++ [22], ER [2], AGEM [24], GSS [17], MIR [16], ASER [18], DualNet [6], DVC [26], SCR [4], and SCL [36]. As our proposed method multiplies the size of a batch by 8 times (i.e. firstly adopting semantically distinct augmentation to augment an image into $K = 4$ instances, followed by generating 2 views for each of the instances), in order to have the fair comparison among our proposed method and the baselines in terms of the same computational cost, we set the number of SGD update I to 1 for our proposed method, I to 4 for the baselines based on contrastive learning (e.g. SCR, SCL, DualNet, and DVC), and I to 8 for the other methods. Please note that, as DualNet contains two components (i.e. a slow learner adopting self-supervised learning and a fast learner adopting supervised learning), both of these two

TABLE I: Evaluation results in terms of end accuracy $\mathbb{E}$ (average over 3 random orders of class arrival in the data stream, results later shown in Table II and III follow the same setting). All methods are trained with the similar computational cost.

Methods	Mini-ImageNet			CIFAR-100			CIFAR-10		
	$\mathcal{M}$=1000	$\mathcal{M}$=2000	$\mathcal{M}$=5000	$\mathcal{M}$=1000	$\mathcal{M}$=2000	$\mathcal{M}$=5000	$\mathcal{M}$=200	$\mathcal{M}$=500	$\mathcal{M}$=1000
EWC++	-	4.5 ± 0.2	-	-	5.8 ± 0.3	-	-	18.1 ± 0.3	-
ER	9.3 ± 0.8	12.1 ± 1.5	20.1 ± 1.8	11.1 ± 0.1	14.2 ± 0.7	20.6 ± 0.9	24.1 ± 3.0	29.1 ± 4.1	38.2 ± 3.4
AGEM	5.0 ± 0.8	5.1 ± 0.9	5.2 ± 0.6	6.1 ± 0.5	6.1 ± 0.5	6.1 ± 0.5	18.1 ± 1.4	18.2 ± 1.2	18.3 ± 0.9
GSS	8.4 ± 0.8	11.1 ± 2.7	14.9 ± 2.5	10.4 ± 0.4	12.6 ± 0.7	16.9 ± 1.1	20.3 ± 1.7	24.7 ± 2.9	32.0 ± 5.2
MIR	8.8 ± 0.5	10.9 ± 1.0	18.5 ± 1.2	10.9 ± 0.4	13.6 ± 0.7	19.0 ± 0.9	22.9 ± 3.2	29.6 ± 4.0	37.2 ± 4.2
ASER	13.7 ± 1.4	16.8 ± 1.7	24.8 ± 1.3	13.2 ± 0.8	17.3 ± 0.8	23.3 ± 1.0	22.4 ± 3.2	28.0 ± 4.3	32.5 ± 3.2
DualNet	15.8 ± 0.6	22.9 ± 1.3	27.0 ± 3.0	16.7 ± 2.2	21.5 ± 1.6	25.0 ± 1.6	44.3 ± 2.7	52.8 ± 2.4	56.0 ± 3.1
DVC	22.2 ± 0.7	27.4 ± 0.9	**33.4 ± 0.5**	25.4 ± 0.7	30.5 ± 0.6	36.6 ± 1.6	48.2 ± 3.0	55.6 ± 2.6	59.8 ± 4.1
SCR	15.8 ± 1.5	16.4 ± 2.0	17.7 ± 1.8	20.9 ± 1.2	22.1 ± 1.4	24.1 ± 0.9	44.6 ± 6.6	58.4 ± 5.1	65.7 ± 2.6
SCL	14.6 ± 0.8	15.8 ± 1.2	16.6 ± 1.3	18.8 ± 1.1	20.4 ± 1.3	22.0 ± 0.9	49.9 ± 5.8	61.0 ± 1.8	66.6 ± 1.5
SDAF	**22.7 ± 0.5**	**28.3 ± 0.3**	33.2 ± 0.5	**29.3 ± 2.1**	**35.3 ± 0.7**	**39.0 ± 0.3**	**52.9 ± 3.5**	**66.4 ± 1.0**	**70.1 ± 0.5**

components will run for $I = 4$ SGD updates before receiving the next new batch. We follow the aforementioned settings of I for SGD updates in all of our experiments unless otherwise specified. The results in terms of end accuracy $\mathbb{E}$ is shown in Table I, in which it is clear to observe that our proposed SDAF method outperforms almost of the baselines on different datasets with various settings of memory size $\mathcal{M}$ except being slightly worse than DVC on Mini-ImageNet with $\mathcal{M} = 5000$ (noting that DVC has a specific strategy to draw the most informative samples from memory, where such strategy benefits more when memory size gets larger).

The results on average incremental accuracy $\mathbb{A}$ is shown in Table II, in which it is clear to again observe that our SDAF method outperforms almost of the baselines on different datasets with various settings of memory size $\mathcal{M}$.

Results in Forgetting Measure $\mathbb{F}$. Table III shows the forgetting measure ($\mathbb{F}$) of several methods, where our SDAF is shown to effectively alleviate catastrophic forgetting. It is contributed to our using strong augmentations for generating diverse samples as novel classes, which leads to better learning the generalised representation and the model is able to handle the newly arriving classes with less adjustment.

Ablation Study on SDA. To evaluate the effectiveness of our proposed component, we conducted an ablation study on CIFAR-10 and CIFAR-100. In this study, we adjusted the number of SGD updates, denoted as I, to maintain a similar computational cost. The results, measured in terms of the end accuracy $\mathbb{E}$, is presented in Table IV. Starting from an experience replay model, which means that we only use cross-entropy loss and a memory $\mathcal{M}$ to store exemplars for replay, then we sequentially add the proposed modules to analyse their contributions. "+ Contrastive" indicates the design of adding the contrastive loss with the transformation module $\mathcal{H}$ based on the experience replay model, and "+ Rotation (align)" indicates the design that we apply the rotation transformation to the input images but these samples are treated as belonging to their original classes, while "+ SDA" instead indicates the design that we treat samples generated by $\mathcal{S}$ as new classes as described in Section III-B. Finally, "+ WABS" indicates that the proposed WABS loss is used to replace the original cross-entropy loss. As shown in Table IV, all the proposed components have a positive impact on end accuracy $\mathbb{E}$, showing the effectiveness of our designs.

V. Conclusion

We propose a class-incremental online continual learning approach which stems from the idea of learning generalised representation to alleviate the issue of catastrophic forgetting, where we particularly utilise the SDA for producing the semantically distinct classes to enhance the generalised representation learning as well as additionally adopt the softmax classifier and the WABS strategy to tackle the imbalanced dataset. With a similar computational cost, our method performs better with respect to several baselines.

References

[1] Matthias Delange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Greg Slabaugh, and Tinne Tuytelaars, "A continual learning survey: Defying forgetting in classification tasks," *T-PAMI*, 2021.

[2] Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato, "On tiny episodic memories in continual learning," 2019.

[3] 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," *PNAS*, 2017.

[4] Zheda Mai, Ruiwen Li, Hyunwoo Kim, and Scott Sanner, "Supervised contrastive replay: Revisiting the nearest class mean classifier in online class-incremental continual learning," in *CVPR Workshops*, 2021.

[5] Hyuntak Cha, Jaeho Lee, and Jinwoo Shin, "Co2l: Contrastive continual learning," in *ICCV*, 2021.

[6] Quang Pham, Chenghao Liu, et al., "Dualnet: Continual learning, fast and slow," in *NeurIPS*, 2021.

[7] Mehdi Noroozi and Paolo Favaro, "Unsupervised learning of visual representations by solving jigsaw puzzles," in *ECCV*, 2016.

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

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

[10] Xinlei Chen and Kaiming He, "Exploring simple siamese representation learning," in *CVPR*, 2021.

[11] Yifei Wang, Zhengyang Geng, Feng Jiang, Chuming Li, Yisen Wang, Jiansheng Yang, and Zhouchen Lin, "Residual relaxation for multi-view representation learning," in *NeurIPS*, 2021.

[12] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich, "Deep anomaly detection with outlier exposure," in *ICLR*, 2019.

TABLE II: Evaluation results in terms of average incremental accuracy $\mathbb{A}$. All methods are trained with the similar computational cost.

Methods	Mini-ImageNet			CIFAR-100			CIFAR-10		
	M=1k	M=2k	M=5k	M=1k	M=2k	M=5k	M=0.2k	M=0.5k	M=1k
EWC++	-	6.4 ± 0.8	-	-	8.9 ± 1.1	-	-	35.8 ± 4.6	-
ER	21.6 ± 2.4	25.2 ± 3.0	32.2 ± 2.9	24.8 ± 1.3	29.3 ± 1.6	34.5 ± 1.8	48.2 ± 2.0	53.7 ± 2.8	59.7 ± 3.1
AGEM	10.4 ± 1.2	10.5 ± 1.0	10.6 ± 0.9	14.9 ± 0.6	15.0 ± 0.5	14.9 ± 0.5	40.8 ± 1.3	40.8 ± 1.3	40.8 ± 1.4
GSS	20.3 ± 3.2	23.7 ± 3.4	27.4 ± 3.4	24.3 ± 1.7	27.1 ± 2.0	31.0 ± 2.1	43.4 ± 0.9	46.5 ± 0.8	53.9 ± 2.9
MIR	20.9 ± 2.1	24.0 ± 1.4	31.1 ± 2.9	24.5 ± 1.6	28.1 ± 1.7	33.6 ± 2.5	46.9 ± 2.0	53.6 ± 2.0	58.4 ± 2.3
ASER	24.8 ± 2.7	29.5 ± 2.0	35.9 ± 2.1	26.2 ± 0.6	30.8 ± 1.3	36.4 ± 1.9	44.6 ± 1.4	48.3 ± 1.4	53.0 ± 2.6
DualNet	27.2 ± 1.5	30.5 ± 2.5	32.6 ± 1.4	28.9 ± 2.0	31.9 ± 2.4	33.7 ± 2.3	61.3 ± 0.9	65.2 ± 2.4	67.4 ± 2.0
DVC	33.4 ± 1.0	37.0 ± 0.8	**41.5 ± 0.9**	36.9 ± 0.6	40.4 ± 1.0	43.8 ± 0.6	64.1 ± 1.6	67.1 ± 2.2	70.4 ± 2.3
SCR	25.2 ± 2.6	25.9 ± 2.4	26.8 ± 2.4	31.2 ± 3.7	32.5 ± 3.5	33.2 ± 3.6	65.9 ± 3.6	73.6 ± 2.1	78.4 ± 1.2
SCL	23.3 ± 2.4	23.8 ± 2.0	25.0 ± 2.4	28.7 ± 3.5	29.2 ± 3.6	30.6 ± 3.1	68.7 ± 2.6	75.5 ± 1.6	78.2 ± 0.7
SDAF	**34.2 ± 2.0**	**38.8 ± 2.0**	41.3 ± 1.5	**40.2 ± 2.1**	**44.4 ± 1.3**	**46.6 ± 1.8**	**71.3 ± 1.2**	**78.5 ± 0.8**	**80.7 ± 0.6**

TABLE III: Evaluation results in terms of forgetting measure $\mathbb{F}$ (noting that $\mathbb{F}$ is the smaller the better).

Methods	Mini-ImageNet			CIFAR-100			CIFAR-10		
	$\mathcal{M}$=1000	$\mathcal{M}$=2000	$\mathcal{M}$=5000	$\mathcal{M}$=1000	$\mathcal{M}$=2000	$\mathcal{M}$=5000	$\mathcal{M}$=200	$\mathcal{M}$=500	$\mathcal{M}$=1000
DVC	29.1 ± 0.3	23.1 ± 1.3	15.6 ± 0.7	33.1 ± 0.7	27.6 ± 0.3	17.7 ± 1.1	**34.6 ± 2.8**	27.8 ± 1.6	22.3 ± 3.4
SCR	14.0 ± 1.1	13.2 ± 0.9	11.3 ± 2.3	16.8 ± 2.1	15.3 ± 2.2	13.5 ± 1.1	44.2 ± 3.4	27.6 ± 2.1	21.6 ± 0.5
SCL	20.0 ± 1.7	18.0 ± 1.4	16.9 ± 1.1	14.8 ± 2.0	12.9 ± 1.7	11.2 ± 1.2	34.7 ± 3.5	22.4 ± 1.0	16.8 ± 0.8
SDAF	**12.6 ± 1.1**	**11.7 ± 0.3**	**10.9 ± 0.1**	**13.2 ± 1.3**	**11.0 ± 1.4**	**9.7 ± 0.3**	35.3 ± 6.0	**19.5 ± 0.8**	**14.0 ± 0.3**

TABLE IV: Ablation study on model designs (experiments based on CIFAR-100 with $\mathcal{M} = 2000$ and CIFAR-10 with $\mathcal{M} = 500$).

End accuracy $\mathbb{E}$	CIFAR-100	CIFAR-10
Experience replay	14.2	29.1
+ Contrastive	26.8	61.0
+ Rotation (align)	31.6	64.6
+ SDA	34.4	66.1
+ WABS (proposed SDAF)	35.3	66.4

[13] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis, "Decoupling representation and classifier for long-tailed recognition," in *ICLR*, 2020.

[14] Zhiyuan Chen and Bing Liu, "Lifelong machine learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 2018.

[15] Jeffrey S. Vitter, "Random sampling with a reservoir," *ACM Trans. Math. Softw.*, 1985.

[16] Rahaf Aljundi, Eugene Belilovsky, Tinne Tuytelaars, Laurent Charlin, Massimo Caccia, Min Lin, and Lucas Page-Caccia, "Online continual learning with maximal interfered retrieval," in *NeurIPS*, 2019.

[17] Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio, "Gradient based sample selection for online continual learning," in *NeurIPS*, 2019.

[18] Dongsub Shim, Zheda Mai, Jihwan Jeong, Scott Sanner, Hyunwoo Kim, and Jongseong Jang, "Online class-incremental continual learning with adversarial shapley value," in *AAAI*, 2021.

[19] Huiwei Lin, Shanshan Feng, Xutao Li, Wentao Li, and Yunming Ye, "Anchor assisted experience replay for online class-incremental learning," *TCSVT*, 2022.

[20] Qinghua Hu, Yucong Gao, and Bing Cao, "Curiosity-driven class-incremental learning via adaptive sample selection," *TCSVT*, 2022.

[21] Zhizhong Li and Derek Hoiem, "Learning without forgetting," *TPAMI*, 2017.

[22] Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr, "Riemannian walk for incremental learning: Understanding forgetting and intransigence," in *ECCV*, 2018.

[23] David Lopez-Paz and Marc'Aurelio Ranzato, "Gradient episodic memory for continual learning," in *NeurIPS*, 2017.

[24] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny, "Efficient lifelong learning with a-gem," in *ICLR*, 2019.

[25] Khurram Javed and Martha White, "Meta-learning representations for continual learning," in *NeurIPS*, 2019.

[26] Yanan Gu, Xu Yang, Kun Wei, and Cheng Deng, "Not just selection, but exploration: Online class-incremental continual learning via dual view consistency," in *CVPR*, 2022.

[27] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al., "Distilling the knowledge in a neural network," in *NeurIPS Workshops*, 2015.

[28] Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara, "Dark experience for general continual learning: a strong, simple baseline," in *NeurIPS*, 2020.

[29] Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang, "Lifelong learning with dynamically expandable networks," in *ICLR*, 2018.

[30] Soochan Lee, Junsoo Ha, Dongsu Zhang, and Gunhee Kim, "A neural dirichlet process mixture model for task-free continual learning," in *ICLR*, 2020.

[31] Adrien Bardes, Jean Ponce, and Yann LeCun, "Vicreg: Variance-invariance-covariance regularization for self-supervised learning," 2021.

[32] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick, "Momentum contrast for unsupervised visual representation learning," in *CVPR*, 2020.

[33] Pietro Buzzega, Matteo Boschini, Angelo Porrello, and Simone Calderara, "Rethinking experience replay: a bag of tricks for continual learning," in *ICPR*, 2021.

[34] Matthias De Lange and Tinne Tuytelaars, "Continual prototype evolution: Learning online from non-stationary data streams," in *ICCV*, 2021.

[35] Peng Wang, Kai Han, Xiu-Shen Wei, Lei Zhang, and Lei Wang, "Contrastive learning based hybrid networks for long-tailed image classification," in *CVPR*, 2021.

[36] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan, "Supervised contrastive learning," in *NeurIPS*, 2020.

[37] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu, "Large scale incremental learning," in *CVPR*, 2019.

[38] Bowen Zhao, Xi Xiao, Guojun Gan, Bin Zhang, and Shu-Tao Xia, "Maintaining discrimination and fairness in class incremental learning," in *CVPR*, 2020.

[39] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel, "Fixmatch: Simplifying semi-supervised learning with consistency and confidence," in *NeurIPS*, 2020.