Supervised Contrastive Replay: Revisiting the Nearest Class Mean Classifier in Online Class-Incremental Continual Learning

Zheda Mai[1], Ruiwen Li[1], Hyunwoo Kim[2], Scott Sanner[1]
[1]University of Toronto
[2]LG AI Research
{zheda.mai, ruiwen.li}@mail.utoronto.ca, hwkim@lgresearch.ai, ssanner@mie.utoronto.ca

Abstract

Online class-incremental continual learning (CL) studies the problem of learning new classes continually from an online non-stationary data stream, intending to adapt to new data while mitigating catastrophic forgetting. While memory replay has shown promising results, the recency bias in online learning caused by the commonly used Softmax classifier remains an unsolved challenge. Although the Nearest-Class-Mean (NCM) classifier is significantly undervalued in the CL community, we demonstrate that it is a simple yet effective substitute for the Softmax classifier. It addresses the recency bias and avoids structural changes in the fully-connected layer for new classes. Moreover, we observe considerable and consistent performance gains when replacing the Softmax classifier with the NCM classifier for several state-of-the-art replay methods.

To leverage the NCM classifier more effectively, data embeddings belonging to the same class should be clustered and well-separated from those with a different class label. To this end, we contribute Supervised Contrastive Replay (SCR), which explicitly encourages samples from the same class to cluster tightly in embedding space while pushing those of different classes further apart during replay-based training. Overall, we observe that our proposed SCR substantially reduces catastrophic forgetting and outperforms state-of-the-art CL methods by a significant margin on a variety of datasets.

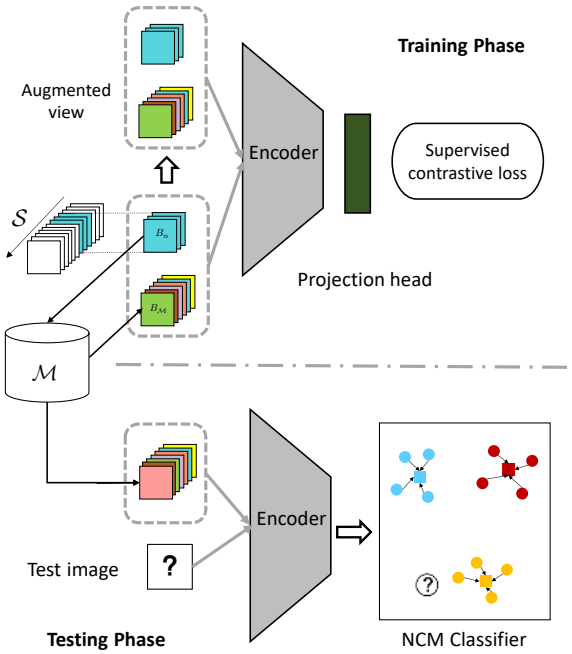
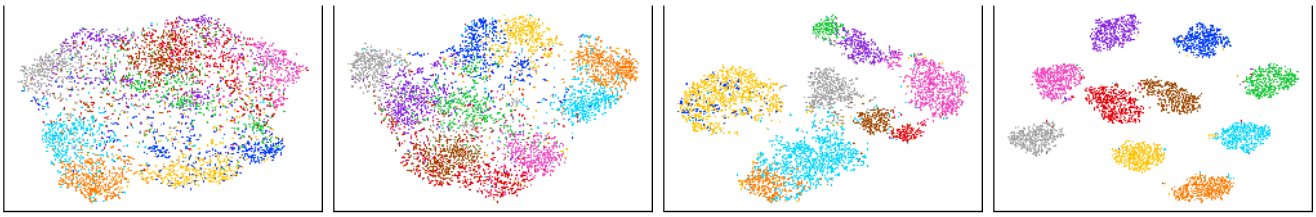


Figure 1: An overview of SCR. During training, an input batch is created by concatenating the minibatch B_n from the data stream with another minibatch $B_{\mathcal{M}}$ from the memory buffer $\mathcal{M}$. The input batch and its augmented view are encoded by a shared encoder and projection head before the representations are evaluated by the supervised contrastive loss. During testing, the projection head is discarded and all the buffered samples are used to compute the class means for the NCM classifier.

1. Introduction

With the ubiquity of personal smart devices and image-related applications, a massive amount of image data is generated daily. A practical online learning system is expected to learn incrementally without storing all streaming data and retraining over it due to space and computational resource limitations. However, a well-documented drawback of deep neural networks that prevents it from learning continually is called *catastrophic forgetting* [40] — the inability to retain previously learned knowledge after learning new tasks. To address this challenge, *Continual Learning* (CL) studies the problem of learning from a non-i.i.d stream of data, intending to preserve and extend the acquired knowledge while minimizing storage, computation, and time.

Most early CL approaches considered *task-incremental*



(a) ER (b) MIR (c) ASER$_\mu$ (d) **SCR(Ours)**

Figure 2: 2D t-SNE [53] visualization of data embeddings in the memory buffer M by the end of the training (CIFAR-100). Note that ER [9], MIR [2] and ASER [50] are three state-of-the-art methods that use categorical cross-entropy loss to train the network. By using supervised contrastive loss, the embeddings of our proposed SCR are better clustered and separated based on labels, which provides a solid foundation for using distance-based classifiers such as NCM [41] and cosine-similarity-based classifier [16].

settings, in which new data arrives one task at a time, and the model can utilize task-IDs during both training and inference time [28, 35, 36]. This setting implicitly simplifies the CL problem as the model just needs to classify labels within a task with the help of task-IDs. Meanwhile, this simplification diminishes the applicability of this setting when task-IDs are not available. In this work, we consider a more realistic but challenging setting, known as *online class-incremental*, where a model is required to learn new classes continually from an online data stream (each sample is seen only once) and classify all labels without task-IDs.

Current CL methods can be taxonomized into three major categories: regularization, parameter-isolation, and replay methods [43, 14]. The replay approach has been shown to be simple and efficient compared to other approaches in the online class-incremental setting [2, 3] . However, A key challenge of replay methods is the imbalance between old and new classes, as only a small amount of old class data are stored in the replay buffer. Recent works have revealed that the Softmax classifier and its associated fully-connected (FC) layer are seriously affected by the class imbalance, which leads to *task-recency bias* — the tendency of a model to be biased towards classes from the most recent task [39, 37, 24, 57]. Although the Nearest-Class-Mean (NCM) classifier [41] is significantly undervalued in the CL community, we demonstrate that it is a simple yet effective substitute for the Softmax classifier as it not only addresses the recency bias but also avoids structural changes in the FC layer when new classes are observed. Moreover, we observe considerable and consistent performance gains when replacing the Softmax classifier with the NCM classifier for five methods with memory buffers. Since [61] also observed similar gains in methods without memory buffer, we advocate using the NCM classifier instead of the commonly used Softmax classifier for future study.

Furthermore, to exploit the NCM classifier more effectively, the data embeddings belonging to the same class should be clustered and well-separated from those with different class labels. To this end, we contribute Supervised Contrastive Replay (SCR), which leverages the *supervised contrastive loss* [26] to explicitly encourage samples from the same class to cluster tightly in embedding space and push those of different classes further apart when replaying buffered samples with the new samples. Through extensive experiments on three commonly used benchmarks in the CL literature, we demonstrate that SCR outperforms state-of-the-art methods by significant margins with three different memory buffer sizes.

2. Related Work

2.1. Continual Learning

Online Class-Incremental Learning Following the recent CL literature [2, 3, 32, 12], we consider the online supervised class-incremental learning setting where a model needs to learn new classes continually from an online data stream (each sample is seen only once). Formally, we define a data stream $\mathcal{D} = \{D_1, \dots, D_N\}$ over $X \times Y$, where X and Y are input/output random variables respectively and N is the number of tasks. Note that tasks do not overlap in classes, meaning $\{Y_i\} \cap \{Y_j\} = \emptyset$ if $i \neq j$ (where $\{Y_k\}$ represents the set of data for task k). We consider a classification model with two components: a encoder $f : X \mapsto \mathbb{R}^d$ that maps an input image to a compact d-dimensional vectorial embedding, and a classifier $g : \mathbb{R}^d \mapsto \mathbb{R}^c$ which maps the embedding to output predictions (c is the number of classes observed so far). A CL algorithm A is defined with the following signature:

$$A_t : \langle (f, g)_{t-1}, B_t^n, M_{t-1} \rangle \rightarrow \langle (f, g)_t, M_t \rangle \quad (1)$$

The model receives a small batch B_t^n of size b from task D_n at time t. f and g will be updated based on B_t^n and data in M_{t-1}, a bounded memory that can be used to store

a subset of the training samples or other useful data [35, 9]. Moreover, we adopt the single-head evaluation setup [7] where the classifier has no access to task-IDs during inference and hence must choose among all labels. Our goal is to train the model (f, g) to continually learn new classes from the data stream without forgetting.

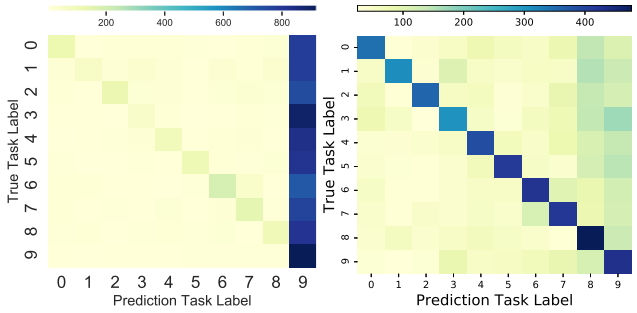
Approaches As previously discussed, current CL methods can be classified into three major categories: regularization, parameter-isolation, and replay methods [43, 14]. **Regularization** methods constraint the updates of some important network parameters to mitigate catastrophic forgetting. This is done by either incorporating additional penalty terms into the loss function [33, 1, 62, 48] or modifying the gradient of parameters during optimization [36, 8, 22]. Other regularization methods imposed knowledge distillation [23] techniques to penalize the feature drift on previous tasks [35, 57, 45]. **Parameter-isolation** methods bypass interference by allocating different parameters to each task [38, 32, 60]. **Replay** methods deploy a memory buffer to store a subset of data from previous tasks for replay [47, 9]. Regularization methods mostly protect the model's ability to classify within a task, and thus they do not work well in our setting, which requires the ability to classify from all labels the model has seen before [34]. Also, most parameter isolation methods require task-IDs during inference, which violates our setting. Therefore, in this work, we will focus on replay methods, which have been shown to be efficient and effective compared to other approaches in the online class-incremental setting [2, 3].

Metrics We use the *average accuracy* of the test sets from observed tasks to measure the overall performance [7, 9]. In Average Accuracy, $a_{i,j}$ is the accuracy evaluated on the held-out test set of task j after training the network from task 1 to i. By the end of training all N tasks, the average accuracy can be calculated as follows:

$$\text{Average Accuracy}(A_N) = \frac{1}{N}\sum_{j=1}^{N} a_{N,j} \qquad (2)$$

2.2. Contrastive Learning

The general goal of contrastive learning is intuitive: the representation of "similar" samples should be mapped close together in the embedding space, while that of "dissimilar" samples should be further away [25, 30]. When labels are not available (self-supervised), similar samples are often formed by data augmentations of the target sample while dissimilar samples are often drawn randomly from the same batch of the target sample [10] or from the memory bank/queue that stores feature vectors [21, 58]. When labels are provided (supervised), similar samples are those from the same class and dissimilar samples are those from different classes [26]. Contrastive learning has recently attracted a surge of interest and shown promising results in various areas including computer vision [18, 6], natural language processing [11, 15], audio processing [49, 42], graph [20, 44] and multimodal data [4, 51].



(a) ER (b) SCR

Figure 3: Confusion matrices for ER and SCR on CIFAR100 with a memory buffer of size 2,000. ER suffers seriously from the task-recency bias as it tends to predict most samples as classes in the most recent task, while SCR is clearly less biased because of the more discernible embeddings and NCM classifier.

3. Method

3.1. Softmax Classifier vs. NCM Classifier

Softmax classifier Softmax classifier with cross-entropy loss has been a standard approach for classification tasks for neural networks [17]. Although this combination also dominates the CL for image classification, it may not be the best choice for CL due to the following deficiencies.

- **Architecture modification for new classes** When the model receives new classes, the Softmax classifier requires the model to stop training and add weights in the FC classification layer to accommodate the new classes.
- **Decoupled representation and classification** In the class-incremental setting, as mentioned in [46], it is problematic that the weights in the classification layers are decoupled from the encoder since whenever the encoder changes, weights in the classification layer must also be updated.
- **Task-recency bias** Multiple previous works [57, 24, 31, 5] have observed that a model with the Softmax classifier has a strong prediction bias towards the most recent task due to the imbalance of new and old classes, which is the primary source of catastrophic

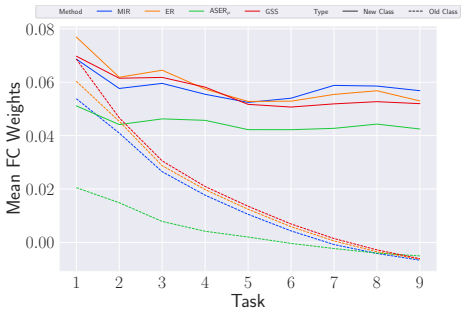


Figure 4: The means of the weights in the FC layer for new and old classes on CIFAR100. The mean of new classes is much higher than that of old classes, which leads to task-recency bias.

forgetting. Figure 3 (a) shows the confusion matrix after training task 10, which shows that the model tends to predict most samples as classes in the most recent task. As illustrated in Figure 4, the means of weights for the new classes in the FC layer are much higher than those for the old classes and hence the model assigns a larger probability mass for predicting a sample as a new class vs. an old class.

Nearest Class Mean (NCM) Classifier The NCM classifier and its variants have been widely used in few-shot or zero-shot learning [56, 59]. Concretely, after the embedding network f is trained, the NCM classifier computes a class mean (prototype) vector for each class using all the embeddings of this class. To predict a label for a new sample $\mathbf{x}$, NCM compares the embedding of $\mathbf{x}$ with all the prototypes and assigns the class label with the most similar prototype:

$$\mu_c = \frac{1}{n_c} \sum_i f(\mathbf{x}_i) \cdot \mathbb{1}\{\mathbf{y}_i = c\} \tag{3}$$

$$y^* = \underset{c=1,\ldots,t}{\operatorname{argmin}} \|f(\mathbf{x}) - \mu_c\| \tag{4}$$

where n_c is the number of samples for class c and $\mathbb{1}\{\mathbf{y}_i = c\}$ is the indicator for $\mathbf{y}_i = c$. The embedding network f keeps being updated in CL, and the true prototype vector for each class cannot be exactly computed with the updated f due to the unavailability of the training data for previous tasks. iCaRL [46] approximates the prototype vectors using the data in the memory buffer, while SDC [61] proposes a drift compensation to update previously computed prototypes without using a memory buffer and .

Although the NCM classifier is significantly undervalued in the CL community, we argue that it is a simple yet effective substitute for the Softmax classifier as it not only resolves the deficiencies of the Softmax classifier mentioned above but also demonstrates a considerable improvement.

- Since the NCM classifier simply compares the embedding of the test sample with prototypes, it does not require an additional FC layer, and therefore, new classes can be added without any architecture modification.
- As the prototypes change instinctively based on the encoder, the NCM classifier is more robust against changes of the encoder.
- The biased weights in the FC layer result in the task-recency bias, but since the NCM classifier does not involve the FC layer, it is intrinsically less prone to the task-recency bias.

Figure 5 shows the average accuracy comparison of a Softmax classifier and an NCM classifier on five methods. NCM classifiers show significant improvements over the commonly used Softmax classifier across all five methods and three datasets, which suggests that the dominance of the Softmax classifier in online continual learning should be revisited.

Although the NCM classifier has shown impressive results, the embedding quality greatly and directly impacts the performance of the NCM classifier. To effectively exploit the NCM classifier, the data embeddings belonging to the same class should be clustered and well-separated from those with a different class label. However, the binary cross-entropy loss used in iCaRL may not be capable of addressing the relationship between classes, and the commonly used categorical cross-entropy loss may not be effective in creating discernible patterns in the embedding space, as shown in Figure 2.

3.2. Supervised Contrastive Replay

Supervised Contrastive Learning To improve accuracy of the vanilla NCM classifier, we propose to leverage contrastive learning, which has shown promising progress in self-supervised learning to obtain more discernible patterns in the embedding space. Specifically, we will focus on the *supervised contrastive learning* (SCL) [26, 19] as labels are available in the online class-incremental setting. Intuitively, SCL aims to tightly cluster embeddings of samples from the same class while pushing those of different classes further apart. Concretely, following the framework proposed in [10, 52], SCL consists of three main components. $Aug(\cdot)$ create an augmented view $\tilde{x}$ of a data sample x, $\widetilde{x} = Aug(x)$. Encoder network $Enc(\cdot)$ maps an image sample $\boldsymbol{x}$ to a vectorial embedding $r = Enc(x) \in \mathcal{R}^{D_E}$ (with r normalized to the unit hypersphere in $\mathcal{R}^{D_E}$).Projection network $Proj(\cdot)$ maps r to a projected vector $z = Proj(r) \in \mathcal{R}^{D_P}$ followed by a L2 normalization step. For an incoming batch with b samples $B = \{x_k, y_k\}_{k=1\ldots b}$, we create a multiviewed batch with 2b samples: the original incoming batch and its augmented view, $B_I = B \cup \widetilde{B}$ where

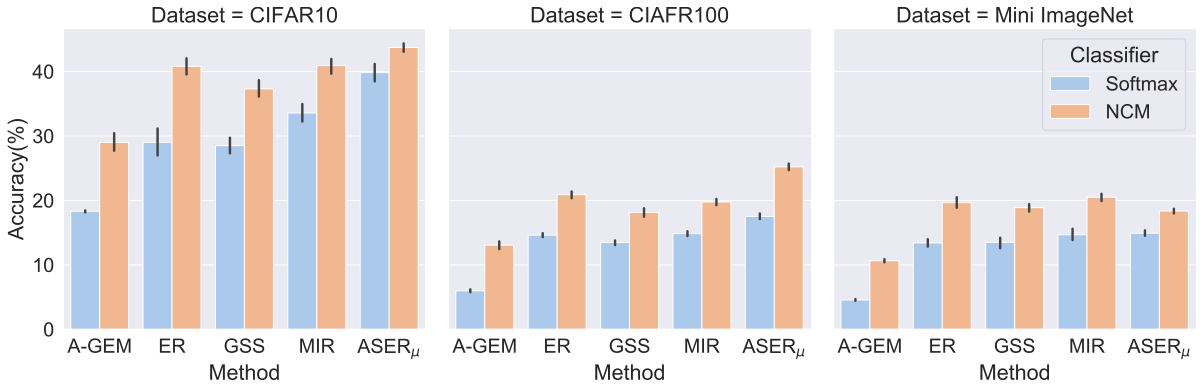


Figure 5: The average accuracy comparison of Softmax classifier and NCM classifier on five methods with memory buffers. We set the memory size to 2,000 for Mini-ImageNet and CIFAR-100 and 500 for CIFAR-10. (Refer to Table. 1 for a more detailed comparison with different memory sizes). Methods with NCM classifiers show significant improvements over the commonly used Softmax classifier across all three datasets, which suggests that the dominance of the Softmax classifier in online continual learning should be revisited.

$\widetilde{B} = \{\tilde{x_k} = Aug(x_k), y_k\}_{k=1\ldots b}$. The SCL loss takes the following form:

$$\mathcal{L}_{\text{SCL}}(Z_I) = \sum_{i\in I} \frac{-1}{|P(i)|} \sum_{p\in P(i)} \log \frac{\exp\left(z_i \cdot z_p/\tau\right)}{\sum_{j\in A(i)} \exp\left(z_i \cdot z_j/\tau\right)} \tag{5}$$

where I is the set of indices of B_I and $A(i) = I\backslash\{i\}$, represents the set of indices of all samples in B_I except for sample i. $P(i) \equiv \left\{p \in A(i) : \boldsymbol{y}_p = \boldsymbol{y}_i\right\}$ is the set of all positives (i.e., samples with the same labels as sample i) in B_I excluding sample i, and $|P(i)|$ is its cardinality. $Z_I = \{z_i\}_{i\in I} = \{Proj(Enc(x_i)\}_{i\in I}$; $\tau \in \mathcal{R}^+$ is an adjustable temperature parameter controlling the separation of classes; the $\cdot$ indicates the dot product.

Supervised Contrastive Replay (SCR) An overview of SCR can be found in Figure 1. As mentioned in Section 2.1, during the training phase, the model receives one small batch B_n at a time from task D_n in the data stream $\mathcal{D}$. An input batch is created by concatenating B_n with another batch $B_{\mathcal{M}}$ selected from the memory buffer $\mathcal{M}$. The input batch and its augmented view are encoded by a shared encoder network $Enc(\cdot)$ and a projection network $Proj(\cdot)$ before the representations are evaluated by the supervised contrastive loss $\mathcal{L}_{\text{SCL}}$. After updating both $Enc(\cdot)$ and $Proj(\cdot)$ with the gradient from $\mathcal{L}_{\text{SCL}}$, the memory buffer $\mathcal{M}$ will be updated with B_n.

During the testing phase, $Proj(\cdot)$ is discarded. All the buffered samples are fed into $Enc(\cdot)$ to obtain the embeddings, which are used to compute the class means (prototypes) for the NCM classifier. As SCR builds much more discernible patterns in the embedding space with the contrastive loss, the NCM classifier is able to unleash its capability in our method. Algorithm 1 summarizes the training and inference procedures.

Algorithm 1: Supervised Contrastive Replay

Initialize: Memory $\mathcal{M} \leftarrow \{\} * M$; $Aug(\cdot)$; $Enc_\theta(\cdot)$; $Proj_\phi(\cdot)$

for $n \in \{1, \ldots, N\}$ **do**

Training phase:

for $B_n \sim D_n$ **do**

$B_{\mathcal{M}} \leftarrow$ *MemoryRetrieval*$(B_n, \mathcal{M})$

$B_{n\mathcal{M}} \leftarrow B_n \cup B_{\mathcal{M}}$

$B_I \leftarrow B_{n\mathcal{M}} \cup Aug(B_{n\mathcal{M}})$

$Z_I \leftarrow Proj_\phi(Enc_\theta(B_I))$

$\theta, \phi \leftarrow \text{SGD}(\mathcal{L}_{\text{SCL}}(Z_I), \theta, \phi)$ `// Eq. 5`

$\mathcal{M} \leftarrow$ *MemoryUpdate*$(B_n, \mathcal{M})$

Testing phase:

`//` $C \leftarrow$ `number of observed classes`

for $c \in \{1, \ldots, C\}$ **do**

`//` $n_c \leftarrow$ `number of class` c `samples`

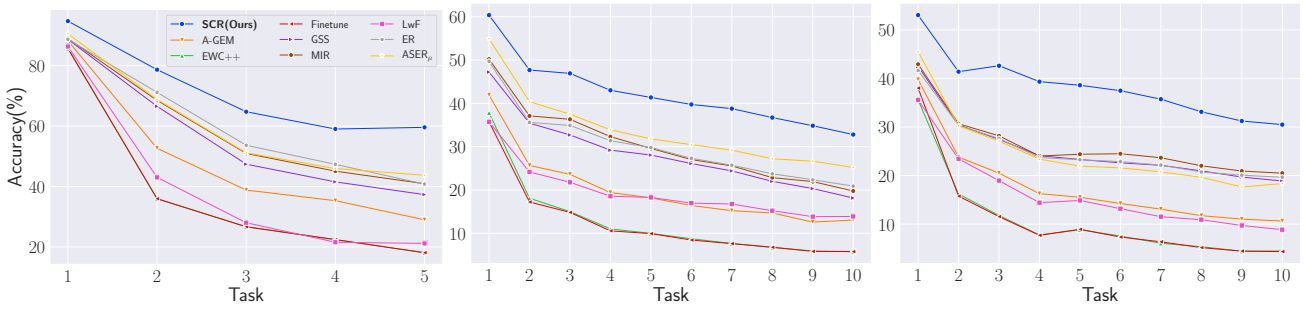
$\mu_c = \frac{1}{n_c} \sum_i^{|\mathcal{M}|} Enc_\theta(x_i) \cdot \mathbb{1}\{\mathbf{y}_i = c\}$

$y^* = \underset{c=1,\ldots,t}{\text{argmin}} \|Enc_\theta(\mathbf{x}) - \mu_c\|$ `// classify` $\mathbf{x}$

4. Experiment

4.1. Experiment Setup

Datasets **Split CIFAR-10** is constructed by splitting the CIFAR-10 dataset [29] into 5 different tasks with non-overlapping classes and 2 classes in each task, similarly as in [2]. **Split CIFAR-100** splits the CIFAR-100 dataset



(a) CIFAR10 (b) CIFAR100 (c) Mini ImageNet

Figure 6: Average accuracy on observed tasks on CIFAR10 (M=0.2k), CIFAR100 (M=2k) and Mini-ImageNet (M=2k). SCR consistently outperform all the compared methods by an enormous margin. Note that all the compared methods on the plots use the NCM classifier.

[29] into 10 disjoint tasks, and each task has 10 classes. **Split Mini-ImageNet** divides the Mini-ImageNet dataset [54] into 10 disjoint tasks with 10 classes per task.

Baselines We compare our proposed SCR against several state-of-the-art continual learning algorithms:

- **A-GEM** (ICLR'19) [8]: Averaged Gradient Episodic Memory, that utilizes the samples in the memory buffer to constrain the parameter updates.
- **ASER**$_\mu$ (AAAI'21) [50]: Adversarial Shapley Value Experience Replay that leverages Shapley value adversarially in memory retrieval.
- **ER** (ICML-W'19)[9]: Experience replay, a replay method with random sampling in memory retrieval and reservoir sampling in memory update.
- **EWC++** (ECCV'18) [7]: An online version of EWC [28], a regularization method that limits the update of parameters that were crucial to the past tasks.
- **GSS** (NeurIPS'19) [3]: Gradient-Based Sample Selection, a replay method that diversifies the gradients of the samples in the replay memory.
- **LwF** (TPAMI'18) [35] Learning Without Forgetting, a regularization method that utilizes knowledge distillation to penalize the feature drifts on previous tasks.
- **MIR** (NeurIPS'19) [2]: Maximally Interfered Retrieval, a replay method that retrieves memory samples with loss increases given the estimated parameter update based on the current batch.
- **offline**: This is not a CL method, but rather an upper bound; offline trains the model over multiple epochs on the whole dataset with iid sampled mini-batches. We use 50 epochs for offline training.

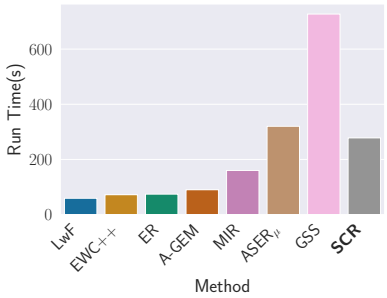


Figure 7: Run time (training + inference) comparison. SCR achieves state-of-the-art performance without sacrificing computation efficiency.

- **fine-tune**: A lower-bound method that simply trains the model when new data is presented without any measure for forgetting avoidance.

Implementation Detail Following [9, 36, 8, 2], we use a reduced ResNet18 as the backbone model for all datasets. We use stochastic gradient descent with a learning rate of 0.1, and the model receives a batch with size 10 at a time from the data stream. All the methods except for SCR are trained with cross-entropy loss and classify with the Softmax classifier. The projection network of SCR is a Multi-Layer Perceptron (MLP) [17] with one hidden layer (ReLU) and an output size 128, and we set the temperature τ to 0.1. We use reservoir sampling [55] for memory update and random sampling for memory retrieval and use a memory batch size 100. The ablation study of the variables mentioned above will be discussed in Section. 4.4.

4.2. Evaluation of NCM Classifier

To assess the effectiveness of the NCM classifier, we compare five methods that employ memory buffers

Method	M=1k	M=2k	M=5k	M=1k	M=2k	M=5k	M=0.2k	M=0.5k	M=1k
fine-tune		4.3 ± 0.2			5.8 ± 0.3			18.1 ± 0.3	
iid offline		51.4 ± 0.2			49.6 ± 0.2			81.7 ± 0.1	
EWC++		4.5 ± 0.2			5.8 ± 0.3			18.1 ± 0.3	
LwF		8.9 ± 0.5			13.9 ± 0.5			21.2 ± 0.9	
AGEM	4.5 ± 0.4	4.6 ± 0.2	4.6 ± 0.2	5.8 ± 0.3	6.0 ± 0.3	5.9 ± 0.2	18.2 ± 0.3	18.3 ± 0.2	18.2 ± 0.2
AGEM-NCM	9.5 ± 0.3	10.6 ± 0.3	11.6 ± 0.5	11.5 ± 0.8	13.1 ± 0.8	14.3 ± 0.4	28.1 ± 1.8	29.0 ± 1.8	29.1 ± 0.9
ER	10.3 ± 0.7	13.4 ± 0.7	16.4 ± 1.5	11.2 ± 0.6	14.6 ± 0.4	21.0 ± 0.9	22.4 ± 1.1	29.0 ± 2.5	37.7 ± 2.0
ER-NCM	16.8 ± 0.8	19.7 ± 1.0	21.1 ± 0.8	16.8 ± 0.5	20.9 ± 0.6	28.3 ± 1.0	30.8 ± 2.0	40.8 ± 1.5	49.4 ± 0.9
GSS	10.5 ± 0.6	13.5 ± 1.1	14.5 ± 2.2	10.6 ± 0.4	13.5 ± 0.4	18.0 ± 1.1	23.0 ± 0.9	28.5 ± 1.5	34.6 ± 2.3
GSS-NCM	15.2 ± 0.9	18.9 ± 0.7	20.9 ± 1.3	13.2 ± 0.7	18.1 ± 0.9	25.8 ± 0.7	28.5 ± 1.2	37.3 ± 1.6	46.6 ± 2.0
MIR	10.7 ± 0.7	14.7 ± 1.1	17.3 ± 1.6	11.7 ± 0.3	14.9 ± 0.5	21.6 ± 1.2	23.8 ± 0.9	33.6 ± 1.7	43.0 ± 1.6
MIR-NCM	17.8 ± 0.5	20.5 ± 0.7	22.1 ± 0.9	16.4 ± 0.4	19.8 ± 0.6	27.9 ± 1.0	31.2 ± 1.5	40.9 ± 1.5	49.9 ± 1.0
ASER_μ	12.5 ± 0.8	14.9 ± 0.5	18.2 ± 0.9	14.4 ± 0.6	17.5 ± 0.6	21.7 ± 1.0	28.5 ± 1.3	39.8 ± 1.7	46.7 ± 1.3
ASER_μ-NCM	16.6 ± 0.7	18.4 ± 0.5	21.1 ± 0.3	22.0 ± 0.6	25.2 ± 0.7	29.6 ± 0.4	34.1 ± 0.8	43.7 ± 0.8	50.3 ± 0.9
SCR	$\mathbf{24.1 \pm 0.6}$	$\mathbf{30.6 \pm 0.5}$	$\mathbf{35.4 \pm 0.5}$	$\mathbf{26.6 \pm 0.5}$	$\mathbf{32.8 \pm 0.7}$	$\mathbf{37.8 \pm 0.3}$	$\mathbf{48.6 \pm 1.1}$	$\mathbf{59.6 \pm 1.2}$	$\mathbf{65.7 \pm 0.6}$
Gains	6.3 ↑	10.1 ↑	13.3 ↑	4.6 ↑	7.6 ↑	8.2 ↑	14.5 ↑	15.9 ↑	15.4 ↑
	(a) Mini-ImageNet			(b) CIFAR-100			(c) CIFAR-10		

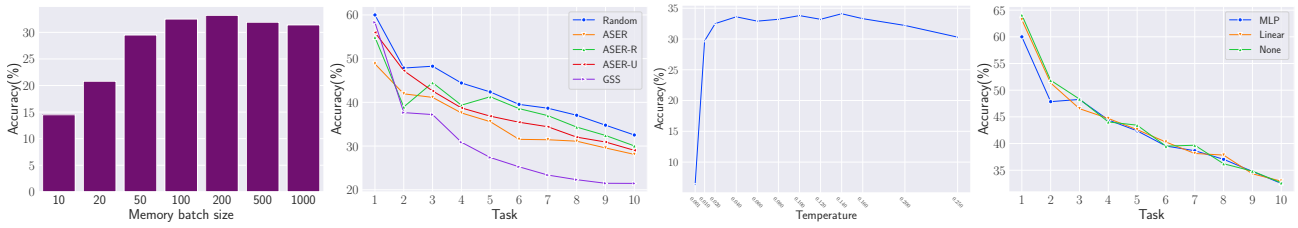
Table 1: Average Accuracy by the end of training. M is the memory buffer size and all numbers are the average of 10 runs. SCR considerably and consistently outperforms all the compared methods by large margins in different datasets and memory sizes.

(AGEM, ER, GSS, MIR, ASER_μ) with their variants equipped with the NCM classifier. As we can see in Figure 5 and Table 1, methods with the NCM classifier show significant improvements over those with the default Softmax classifier. For instance, in CIFAR100, the NCM classifier helps ASER_μ with 1k memory achieve 22%, which requires *five times more memory* to achieve when using the Softmax classifier. Generally, we also observe that the performance gain is more notable when the memory buffer is small. For example, in Mini-ImageNet, MIR obtains 66.4% relative improvement (10.3% $\rightarrow$ 17.8%) with M=1k, which is only improved by 27.7% relatively (17.3% $\rightarrow$ 22.1%) with M=5k. Furthermore, the NCM gains are less obvious for GSS, and we find out that it's because some classes only have a few or sometimes zero samples in the GSS buffer, which makes it hard to estimate the correct prototypes for those classes. Moreover, ASER_μ has better NCM gains in general, and it's because ASER_μ tends to learn more discernible embeddings, as we can see in Figure 1.

To sum up, we observe considerable and consistent performance gains when replacing the Softmax classifier with the NCM classifier for five methods on three different datasets and memory sizes. Since [61] also observed similar gains in methods without memory buffer, we advocate using the NCM classifier instead of the commonly used Softmax classifier for future study.

4.3. Evaluation of SCR

To evaluate the performance of SCR, we compare it with several state-of-the-art CL methods described in Section 4.1. As we can see in Figure 6, SCR consistently outperforms all the compared methods by enormous margins along the whole data streams of three different datasets. Note that all the compared methods on the plots have already been NCM-augmented. Table 1 shows the detailed comparison of SCR with all the compared methods on different datasets and memory sizes. The last row of the table shows the absolute improvements over the second-best methods. SCR consistently achieves state-of-the-art results across all settings and outperforms the compared methods by large margins. SCR achieves 35.4% (13.3%↑), 37.8% (8.2%↑) and 65.7% (15.4%↑) respectively in Mini-ImageNet, CIFAR100 and CIFAR10 respectively. The success of SCR comes from (i) the NCM classifier, which has shown impressive performance over the Softmax classifier in Section 4.2, and (ii) the contrastive loss, which enables the model to learn more discernible embeddings and provides a solid foundation for the NCM classifier. Moreover, we observe SCR benefits from a large memory buffer in general, as contrastive learning desires more diverse negative samples. For example, SCR achieves 60.2% relative gain with M=5k (21.1% (MIR-NCM) $\rightarrow$ 35.4%), while obtains 35.4% relative improvement with M=1k (16.6% (MIR-NCM) $\rightarrow$ 24.1%). In terms of task-recency bias, we



(a) Impact of memory batch size (b) Impact of memory management (c) Impact of temperature (d) Impact pf projection network

Figure 8: Average accuracy of SCR with M=2k on CIFAR100 for ablation study.

can see in Figure 3 (b) that SCR is clearly much less biased than ER even though a slight bias is still observed. Furthermore, SCR does not sacrifice its computation efficiency, as shown in Figure 7. Its running time (combined training and inference) is shorter than ASER$_\mu$ and only slightly longer than MIR.

In summary, by evaluating on three standard CL datasets and comparing to the state-of-the-art CL methods, we have strongly demonstrated the effectiveness and efficiency of SCR in overcoming catastrophic forgetting, which brings online CL much closer to its ultimate goal of matching offline training while maintaining a low computation footprint.

4.4. Ablation Study

In this subsection, we aim to explore the impact of various SCR configurations on its performance. We use SCR with M=2k on CIFAR100 as the study case to analyze the impacts of components of SCR.

Impact of memory batch size $B_\mathcal{M}$. Figure 8 (a) shows the impact of the memory batch size. Generally speaking, contrastive learning benefits from larger batch sizes as it means more negative samples [10, 26]. Nevertheless, in online CL, accuracy improvement is more obvious with the increase of $B_\mathcal{M}$ when $B_\mathcal{M}$ is smaller than 200. The performance drops when $B_\mathcal{M}$ continues to increase. We suspect the decrease is due to the overfitting of the memory samples as 500/1,000 are 25%/50% of the whole memory buffer in this study case.

Impact of memory buffer management. We compare random retrieval + reservoir update(Random), ASER retrieval + update (ASER), ASER retrieval (ASER-R), ASER update (ASER-U) and GSS. As we can see in Figure 8 (b), the random option is much better than GSS and slightly better than others. We observed that some classes have only a few or zero samples in the memory for GSS, which is undesirable for SCR. Although random seems reasonable for the balanced CIFAR100 dataset, when facing imbalanced datasets, combing SCR with other memory management methods may yield better performance [27, 13].

Impact of temperature variable τ. We can see from Figure 8 (c) that the performance deteriorates when the τ is too low and too high. SCR with τ ranging from 0.02 to 0.16 achieves stable results.

Impact of projection network $Proj(\cdot)$. We tried Multi-Layer Perceptron (MLP), linear and no projection network (None). Although [10] suggests that a nonlinear projection network improves the representation quality, we find that the choice of projection network is insignificant in online CL as shown in Figure 8 (d).

5. Conclusion

In this paper, we first demonstrated that the NCM classifier is a simple yet effective substitute for the Softmax classifier in the online CL. It resolves several deficiencies of the Softmax classifier and shows considerable and consistent performance gains across a variety of CL methods. Based on these results, we advocate using the NCM classifier instead of the commonly used Softmax classifier for future study of CL methods. Moreover, to leverage the NCM classifier more effectively, we proposed SCR that explicitly encourages samples from the same class to cluster tightly in embedding space while pushing samples of different classes further apart during experience replay-based training.

Empirically, we observe that our proposed SCR substantially reduces catastrophic forgetting in comparison to state-of-the-art CL methods and outperforms them all by a significant margin on various datasets and memory settings. In summary, leveraging a simple randomized experience replay method while using a supervised contrastive loss (in place of cross-entropy) combined with an NCM classifier bring us closer to realizing the ultimate goal of continual learning to perform as well as offline training methods.

References

[1] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 139–154, 2018. 3

[2] Rahaf Aljundi, Eugene Belilovsky, Tinne Tuytelaars, Laurent Charlin, Massimo Caccia, Min Lin, and Lucas Page-Caccia. Online continual learning with maximal interfered retrieval. In *Advances in Neural Information Processing Systems 32*, pages 11849–11860. 2019. 2, 3, 5, 6

[3] Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. In *Advances in Neural Information Processing Systems 32*, pages 11816–11825. 2019. 2, 3, 6

[4] Relja Arandjelovic and Andrew Zisserman. Objects that sound. In *Proceedings of the European conference on computer vision (ECCV)*, pages 435–451, 2018. 3

[5] Eden Belouadah and Adrian Popescu. Il2m: Class incremental learning with dual memory. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 583–592, 2019. 3

[6] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*, 2020. 3

[7] 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 (ECCV)*, pages 532–547, 2018. 3, 6

[8] Arslan Chaudhry, MarcAurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-GEM. In *International Conference on Learning Representations*, 2019. 3, 6

[9] Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K. Dokania, Philip H. S. Torr, and Marc'Aurelio Ranzato. On tiny episodic memories in continual learning, 2019. 2, 3, 6

[10] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 3, 4, 8

[11] Zewen Chi, Li Dong, Furu Wei, Nan Yang, Saksham Singhal, Wenhui Wang, Xia Song, Xian-Ling Mao, Heyan Huang, and Ming Zhou. Infoxlm: An information-theoretic framework for cross-lingual language model pre-training. *arXiv preprint arXiv:2007.07834*, 2020. 3

[12] Aristotelis Chrysakis and Marie-Francine Moens. Online continual learning from imbalanced data. *Proceedings of Machine Learning and Systems*, pages 8303–8312, 2020. 2

[13] Aristotelis Chrysakis and Marie-Francine Moens. Online continual learning from imbalanced data. In *International Conference on Machine Learning*, pages 1952–1961. PMLR, 2020. 8

[14] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. Continual learning: A comparative study on how to defy forgetting in classification tasks. *arXiv preprint arXiv:1909.08383*, 2019. 2, 3

[15] Hongchao Fang and Pengtao Xie. Cert: Contrastive self-supervised learning for language understanding. *arXiv preprint arXiv:2005.12766*, 2020. 3

[16] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4367–4375, 2018. 2

[17] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016. 3, 6

[18] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020. 3

[19] Beliz Gunel, Jingfei Du, Alexis Conneau, and Ves Stoyanov. Supervised contrastive learning for pre-trained language model fine-tuning. *arXiv preprint arXiv:2011.01403*, 2020. 4

[20] Kaveh Hassani and Amir Hosein Khasahmadi. Contrastive multi-view representation learning on graphs. In *International Conference on Machine Learning*, pages 4116–4126. PMLR, 2020. 3

[21] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020. 3

[22] Xu He and Herbert Jaeger. Overcoming catastrophic interference using conceptor-aided backpropagation.

In *International Conference on Learning Representations*, 2018. 3

[23] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 3

[24] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 831–839, 2019. 2, 3

[25] Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. A survey on contrastive self-supervised learning. *Technologies*, 9(1):2, 2021. 3

[26] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33, 2020. 2, 3, 4, 8

[27] Chris Dongjoo Kim, Jinseo Jeong, and Gunhee Kim. Imbalanced continual learning with partitioning reservoir sampling. In *European Conference on Computer Vision*, pages 411–428. Springer, 2020. 8

[28] 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 of the United States of America*, 114 13:3521–3526, 2017. 2, 6

[29] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, April 2009. 5, 6

[30] Phuc H Le-Khac, Graham Healy, and Alan F Smeaton. Contrastive representation learning: A framework and review. *IEEE Access*, 2020. 3

[31] Kibok Lee, Kimin Lee, Jinwoo Shin, and Honglak Lee. Overcoming catastrophic forgetting with unlabeled data in the wild. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 312–321, 2019. 3

[32] Soochan Lee, Junsoo Ha, Dongsu Zhang, and Gunhee Kim. A neural dirichlet process mixture model for task-free continual learning. In *International Conference on Learning Representations*, 2020. 2, 3

[33] Sang-Woo Lee, Jin-Hwa Kim, Jaehyun Jun, Jung-Woo Ha, and Byoung-Tak Zhang. Overcoming catastrophic forgetting by incremental moment matching. In *Advances in neural information processing systems*, pages 4652–4662, 2017. 3

[34] Timothée Lesort, Andrei Stoian, and David Filliat. Regularization shortcomings for continual learning. *arXiv preprint 1912.03049*, 2019. 3

[35] Zhizhong Li and Derek Hoiem. Learning without forgetting. In *ECCV*, pages 614–629. Springer, 2016. 2, 3, 6

[36] David Lopez-Paz and Marc' Aurelio Ranzato. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems 30*, pages 6467–6476. 2017. 2, 3, 6

[37] Zheda Mai, Ruiwen Li, Jihwan Jeong, David Quispe, Hyunwoo Kim, and Scott Sanner. Online continual learning in image classification: An empirical survey. *arXiv preprint arXiv:2101.10423*, 2021. 2

[38] Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7765–7773, 2018. 3

[39] Marc Masana, Xialei Liu, Bartlomiej Twardowski, Mikel Menta, Andrew D Bagdanov, and Joost van de Weijer. Class-incremental learning: survey and performance evaluation. *arXiv preprint arXiv:2010.15277*, 2020. 2

[40] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989. 1

[41] Thomas Mensink, Jakob Verbeek, Florent Perronnin, and Gabriela Csurka. Distance-based image classification: Generalizing to new classes at near-zero cost. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2624–2637, 2013. 2

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

[43] German I. Parisi, Ronald Kemker, Jose L. Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54 – 71, 2019. 2, 3

[44] Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. Gcc: Graph contrastive coding for graph neural network pre-training. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1150–1160, 2020. 3

[45] Amal Rannen, Rahaf Aljundi, Matthew B Blaschko, and Tinne Tuytelaars. Encoder based lifelong learning. In *Proceedings of the IEEE International Con-*

ference on Computer Vision, pages 1320–1328, 2017. 3

[46] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017. 3, 4

[47] Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and Gerald Tesauro. Learning to learn without forgetting by maximizing transfer and minimizing interference. In *International Conference on Learning Representations*, 2018. 3

[48] Hippolyt Ritter, Aleksandar Botev, and David Barber. Online structured laplace approximations for overcoming catastrophic forgetting. In *Advances in Neural Information Processing Systems*, pages 3738–3748, 2018. 3

[49] Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. wav2vec: Unsupervised pre-training for speech recognition. *arXiv preprint arXiv:1904.05862*, 2019. 3

[50] Dongsub Shim, Zheda Mai, Jihwan Jeong, Scott Sanner, Hyunwoo Kim, and Jongseong Jang. Online class-incremental continual learning with adversarial shapley value. *arXiv e-prints*, pages arXiv–2009, 2020. 2, 6

[51] Chen Sun, Fabien Baradel, Kevin Murphy, and Cordelia Schmid. Learning video representations using contrastive bidirectional transformer. *arXiv preprint arXiv:1906.05743*, 2019. 3

[52] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019. 4

[53] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008. 2

[54] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, koray kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems 29*, pages 3630–3638. Curran Associates, Inc., 2016. 6

[55] Jeffrey S Vitter. Random sampling with a reservoir. *ACM Transactions on Mathematical Software (TOMS)*, 11(1):37–57, 1985. 6

[56] Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM Computing Surveys (CSUR)*, 53(3):1–34, 2020. 4

[57] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 374–382, 2019. 2, 3

[58] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3733–3742, 2018. 3

[59] Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *IEEE transactions on pattern analysis and machine intelligence*, 41(9):2251–2265, 2018. 4

[60] Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. In *International Conference on Learning Representations*, 2018. 3

[61] Lu Yu, Bartlomiej Twardowski, Xialei Liu, Luis Herranz, Kai Wang, Yongmei Cheng, Shangling Jui, and Joost van de Weijer. Semantic drift compensation for class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6982–6991, 2020. 2, 4, 7

[62] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. *Proceedings of machine learning research*, 70:3987, 2017. 3