Online Prototype Learning for Online Continual Learning

Yujie Wei[1] Jiaxin Ye[1] Zhizhong Huang[2] Junping Zhang[2] Hongming Shan[1,3,4*]
[1] Institute of Science and Technology for Brain-inspired Intelligence, Fudan University
[2] Shanghai Key Lab of Intelligent Information Processing, School of Computer Science
Fudan University
[3] MOE Frontiers Center for Brain Science, Fudan University
[4] Shanghai Center for Brain Science and Brain-inspired Technology
{yjwei22, jxye22}@m.fudan.edu.cn, {zzhuang19, jpzhang, hmshan}@fudan.edu.cn

Abstract

Online continual learning (CL) studies the problem of learning continuously from a single-pass data stream while adapting to new data and mitigating catastrophic forgetting. Recently, by storing a small subset of old data, replay-based methods have shown promising performance. Unlike previous methods that focus on sample storage or knowledge distillation against catastrophic forgetting, this paper aims to understand why the online learning models fail to generalize well from a new perspective of shortcut learning. We identify shortcut learning as the key limiting factor for online CL, where the learned features may be biased, not generalizable to new tasks, and may have an adverse impact on knowledge distillation. To tackle this issue, we present the online prototype learning (OnPro) framework for online CL. First, we propose online prototype equilibrium to learn representative features against shortcut learning and discriminative features to avoid class confusion, ultimately achieving an equilibrium status that separates all seen classes well while learning new classes. Second, with the feedback of online prototypes, we devise a novel adaptive prototypical feedback mechanism to sense the classes that are easily misclassified and then enhance their boundaries. Extensive experimental results on widely-used benchmark datasets demonstrate the superior performance of OnPro over the state-of-the-art baseline methods. Source code is available at https://github.com/weilllllls/OnPro.

1. Introduction

Current artificial intelligence systems [30, 36, 52, 16] have shown excellent performance on the tasks at hand; however, they are prone to forget previously learned knowledge while learning new tasks, known as *catastrophic forgetting* [20, 23, 9]. Continual learning (CL) [46, 44, 14, 19] aims to learn continuously from a non-stationary data stream while adapting to new data and mitigating catastrophic forgetting, offering a promising path to human-like artificial general intelligence. Early CL works consider the task-incremental learning (TIL) setting, where the model selects the task-specific component for classification with task identifiers [1, 41, 50, 14]. However, this setting lacks flexibility in real-world scenarios. In this paper, we focus on a more general and realistic setting—the class-incremental learning (CIL) in the online CL mode [42, 13, 27, 51]—where the model learns incrementally classes in a sequence of tasks from a single-pass data stream and cannot access task identifiers at inference.

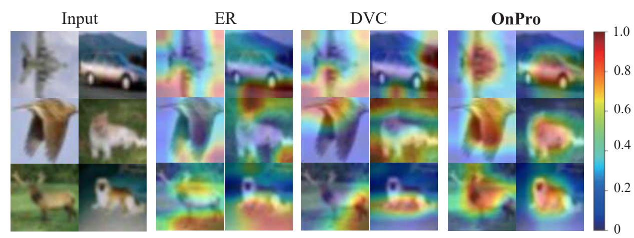


Figure 1. The visual explanations by GradCAM++ on the training set of CIFAR-10 (image size 32×32). Although all methods predict the correct class, shortcut learning still exists in ER and DVC.

Various online CL methods have been proposed to mitigate catastrophic forgetting [51, 43, 25, 28, 11, 5, 13]. Among them, replay-based methods [11, 43, 26, 2, 25] have shown promising performance by storing a subset of data from old classes as exemplars for experience replay. Unlike previous methods that focus on sample storage [51, 3], we are interested in how generalizable the learned features are to new classes, and aim to understand why the online learning models fail to generalize well from a new perspective of shortcut learning.

*Corresponding author

Intuitively, the neural network tends to "take shortcuts" [22] and focuses on simplistic features. *This behavior of shortcut learning is especially serious in online CL*, since the model may learn biased and inadequate features from the single-pass data stream. Specifically, the model may be more inclined to learn trivial solutions *unrelated* to objects, which are hard to generalize and easily forgotten. Take Fig. 1 as an example, when classifying two classes, saying airplanes in the sky and cat on the grass, the model may easily identify the shortcut clue between two classes—blue sky vs. green grass—unfortunately, the learned features are delicate and unrelated to the classes of interest. When new bird and deer classes come, which may also have sky or grass, the model has to be updated due to inapplicable previous knowledge, leading to poor generalization and catastrophic forgetting. Thus, learning *representative* features that best characterize the class is crucial to resist shortcut learning and catastrophic forgetting, especially in online CL.

In addition, the intuitive manifestation of catastrophic forgetting is the confusion between classes. To alleviate class confusion, many works [26, 41, 48, 4, 7, 55] employ self-distillation [17, 32] to preserve previous knowledge. However, the premise for knowledge distillation to succeed is that the model has learned sufficient discriminative features in old classes, and these features still remain discriminative when learning new classes. As mentioned above, the model may learn oversimplified features due to shortcut learning, significantly compromising the generalization to new classes. Thus, distilling these biased features may have an adverse impact on new classes. In contrast, we consider a more general paradigm to maintain discrimination among all seen classes, which can tackle the limitations of knowledge distillation.

In this paper, we aim to learn representative features of each class and discriminative features between classes, both crucial to mitigate catastrophic forgetting. Toward this end, we present the Online Prototype learning (OnPro) framework for online continual learning. The online prototype introduced is defined as "a representative embedding for a group of instances in a mini-batch." There are two reasons for this design: (1) for new classes, the data arrives sequentially from a single-pass stream, and we cannot access all samples of one class at any time step (iteration); and (2) for old classes, computing the prototypes of all samples in the memory bank at each time step is computationally expensive, especially for the online scenario with limited resources. Thus, *our online prototypes only utilize the data available at the current time step (*i.e.*, data within a mini-batch), which is more suitable for online CL.*

To resist shortcut learning in online CL and maintain discrimination among seen classes, we first propose Online Prototype Equilibrium (OPE) to learn representative and discriminative features for achieving an equilibrium status that separates all seen classes well while learning new classes. Second, instead of employing knowledge distillation that may distill unfaithful knowledge from previous models, we devise a novel Adaptive Prototypical Feedback (APF) that can leverage the feedback of online prototypes to first sense the classes—that are easily misclassified—and then adaptively enhance their decision boundaries.

The contributions are summarized as follows.

1) We identify shortcut learning as the key limiting factor for online CL, where the learned features may be biased, not generalizable to new tasks, and may have an adverse impact on knowledge distillation. To the best of our knowledge, this is the first time to identify the shortcut learning issues in online CL, offering new insights into why online learning models fail to generalize well.
2) We present the online prototype learning framework for online CL, in which the proposed online prototype equilibrium encourages learning representative and discriminative features while adaptive prototypical feedback leverages the feedback of online prototypes to sense easily misclassified classes and enhance their boundaries.
3) Extensive experimental results on widely-used benchmark datasets demonstrate the superior performance of our method over the state-of-the-art baseline methods.

2. Related Work

Continual learning. Continual learning methods can be roughly summarized into three categories: regularization-based, parameter-isolation-based, and replay-based methods. Regularization-based methods [9, 1, 39, 31] add extra regularization constraints on network parameters to mitigate forgetting. Parameter-isolation-based methods [49, 50, 38, 18] avoid forgetting by dynamically allocating parameters or modifying the architecture of the network. Replay-based methods [11, 2, 3, 10, 4, 47] maintain and update a memory bank (buffer) that stores exemplars of past tasks. Among them, replay-based methods are the most popular for their simplicity yet efficiency. Experience Replay [11] randomly samples from the buffer. MIR [2] retrieves buffer samples by comparing the interference of losses. Furthermore, in the online setting, ASER [51] introduces a buffer management theory based on the Shapley value. SCR [43] utilizes supervised contrastive loss [34] for training and the nearest-class-mean classifier for testing. OCM [26] prevents forgetting through mutual information maximization.

Unlike these methods that focus on selecting which samples to store or learning features only by instances, our work rethinks the catastrophic forgetting from a new shortcut learning perspective, and proposes to learn representative and discriminative features through online prototypes.

Knowledge distillation in continual learning. Another solution to catastrophic forgetting is to preserve previ-

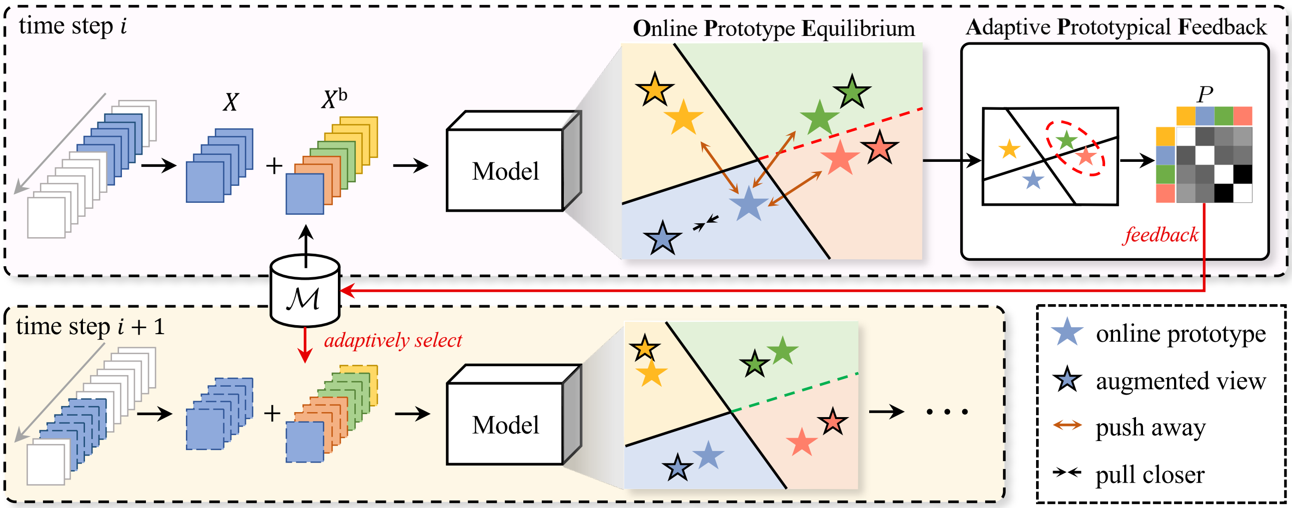


Figure 2. Illustration of the proposed OnPro framework. At time step (iteration) i, the incoming data X and replay data X^{b} are augmented and fed to the model to learn features with OPE. Then, the proposed APF senses easily misclassified classes from all seen classes and enhances their decision boundaries. Concretely, APF adaptively selects more data for mixup according to the probability distribution P.

ous knowledge by self-distillation [48, 4, 41, 7, 55, 26]. iCaRL [48] constrains changes of learned knowledge by distillation and employs class prototypes for nearest neighbor prediction. Co^{2}L [7] proposes a self-distillation loss to preserve learned features. PASS [55] maintains the decision boundaries of old classes by distilling old prototypes. However, it is hard to distill useful knowledge when previous models are not learned well. In contrast, we propose a general feedback mechanism to enhance the discrimination of classes that are prone to misclassification, which overcomes the limitations on knowledge distillation.

Prototypes in continual learning. Some previous methods [48, 43, 55] attempt to utilize prototypes to mitigate catastrophic forgetting. As mentioned above, iCaRL and SCR employ class prototypes as classifiers, and PASS distills old prototypes to retain learned knowledge. Nevertheless, computing prototypes with all samples is extremely expensive for training. There are also some works considering the use of prototypes in the online scenario. CoPE [15] designs the prototypes with a high momentum-based update for each observed batch. A recent work [28] estimates class prototypes on all seen data using mean update criteria. However, regardless of momentum update or mean update, accumulating previous features as prototypes may be detrimental to future learning, since the features learned in old classes may not be discriminative when encountering new classes due to shortcut learning. In contrast, the proposed online prototypes only utilize the data visible at the current time step, which significantly decreases the computational cost and is more suitable for online CL.

Contrastive learning. Inspired by breakthroughs in self-supervised learning [45, 29, 12, 24, 6, 33], many studies [43, 5, 26, 7, 28] in CL use contrastive learning to learn generalized features. An early work [21] analyzes and reveals the impact of contrastive learning on online CL. Among them, the work most related to ours is PCL [40], which calculates infoNCE loss [45] between instance and prototype. The most significant difference is that the loss in OPE only considers online prototypes, and there is no involvement of instances. Please refer to the supplementary material for detailed comparisons between our OPE and PCL.

3. Method

Fig. 2 presents the illustration of the proposed OnPro. In this section, we start by providing the problem definition of online CIL. Then, we describe the definition of the online prototype, the proposed online prototype equilibrium, and the proposed adaptive prototypical feedback. Finally, we propose an online prototype learning framework.

3.1. Problem Definition

Formally, online CIL considers a continuous sequence of tasks from a single-pass data stream $\mathfrak{D} = \{\mathcal{D}_1, \ldots, \mathcal{D}_T\}$, where $\mathcal{D}_t = \{x_i, y_i\}_{i=1}^{N_t}$ is the dataset of task t, and T is the total number of tasks. Dataset $\mathcal{D}_t$ contains N_t labeled samples, y_i is the class label of sample x_i and $y_i \in \mathcal{C}_t$, where $\mathcal{C}_t$ is the class set of task t and the class sets of different tasks are disjoint. For replay-based methods, a memory bank is used to store a small subset of seen data, and we also maintain a memory bank $\mathcal{M}$ in our method. At each time step of task t, the model receives a mini-batch data $X \cup X^{\mathrm{b}}$ for training, where X and X^{b} are drawn from the i.i.d distribution $\mathcal{D}_t$ and the memory bank $\mathcal{M}$, respectively. Moreover, we adopt the single-head evaluation setup [9], where a unified classifier must choose labels from all seen classes at inference due to unavailable task identifiers. The goal of online CIL is to train a unified model on data seen only once while predicting well on both new and old classes.

3.2. Online Prototype Definition

Prior to introducing the online prototypes, we first present the network architecture of our OnPro. Suppose that the model consists of three components: an encoder network f, a projection head g, and a classifier φ. Each sample x in incoming data X (a mini-batch data from new classes) is mapped to a projected vectorial embedding (representation) $\mathbf{z}$ by encoder f and projector g:

$$\mathbf{z} = g(f(\mathrm{aug}(x); \theta_f); \theta_g), \tag{1}$$

where aug represents the data augmentation operation, θ_f and θ_g represent the parameters of f and g, respectively, and $\mathbf{z}$ is ℓ_2-normalized. Similar to Eq. (1), we use $\mathbf{z}^{\mathrm{b}}$ to denote the representation of replay data X^{b} (a mini-batch data from seen classes in the memory bank).

At each time step of task t, the online prototype of each class is defined as the mean representation in a mini-batch:

$$\mathbf{p}_i = \frac{1}{n_i} \sum\nolimits_j \mathbf{z}_j \cdot \mathbb{1}\{y_j = i\}, \tag{2}$$

where n_i is the number of samples for class i in a mini-batch, and $\mathbb{1}$ is the indicator function. We can get a set of K online prototypes in X, $\mathcal{P} = \{\mathbf{p}_i\}_{i=1}^{K}$, and a set of K^{b} online prototypes in X^{b}, $\mathcal{P}^{\mathrm{b}} = \left\{\mathbf{p}_i^{\mathrm{b}}\right\}_{i=1}^{K^{\mathrm{b}}}$. Note that $K = |\mathcal{P}| \le |\mathcal{C}_t|$ and $K^{\mathrm{b}} = |\mathcal{P}^{\mathrm{b}}| \le \sum_{i=1}^{t} |\mathcal{C}_i|$, where $|\cdot|$ denotes the cardinal number.

3.3. Online Prototype Equilibrium

The introduced online prototypes can provide representative features and avoid class-unrelated information. These characteristics are exactly the key to counteracting shortcut learning in online CL. Besides, maintaining the discrimination among seen classes is also essential to mitigate catastrophic forgetting. Based on these, we attempt to learn representative features of each class by pulling online prototypes $\mathcal{P}$ and their augmented views $\widehat{\mathcal{P}}$ closer in the embedding space, and learn discriminative features between classes by pushing online prototypes of different classes away, formally defined as a contrastive loss:

$$\ell(\mathcal{P}, \widehat{\mathcal{P}}) = \frac{-1}{|\mathcal{P}|} \sum_{i=1}^{|\mathcal{P}|} \log \frac{\exp\left(\frac{\mathbf{p}_i^{\mathrm{T}} \widehat{\mathbf{p}}_i}{\tau}\right)}{\sum\limits_j \exp\left(\frac{\mathbf{p}_i^{\mathrm{T}} \widehat{\mathbf{p}}_j}{\tau}\right) + \sum\limits_{j \ne i} \exp\left(\frac{\mathbf{p}_i^{\mathrm{T}} \mathbf{p}_j}{\tau}\right)}, \tag{3}$$

where τ is the temperature hyper-parameter, $\mathcal{P}$ and $\widehat{\mathcal{P}}$ are ℓ_2-normalized. To compute the contrastive loss across all positive pairs in both $(\mathcal{P}, \widehat{\mathcal{P}})$ and $(\widehat{\mathcal{P}}, \mathcal{P})$, we define $\mathcal{L}_{\mathrm{pro}}$ as the final contrastive loss over online prototypes:

$$\mathcal{L}_{\mathrm{pro}}(\mathcal{P}, \widehat{\mathcal{P}}) = \frac{1}{2}\left[\ell(\mathcal{P}, \widehat{\mathcal{P}}) + \ell(\widehat{\mathcal{P}}, \mathcal{P})\right]. \tag{4}$$

Considering the learning of new classes and the consolidation of learned knowledge simultaneously in online CL, we propose Online Prototype Equilibrium (OPE) to learn representative and discriminative features on both new and seen classes by employing $\mathcal{L}_{\mathrm{pro}}$:

$$\mathcal{L}_{\mathrm{OPE}} = \mathcal{L}_{\mathrm{pro}}^{\mathrm{new}}(\mathcal{P}, \widehat{\mathcal{P}}) + \mathcal{L}_{\mathrm{pro}}^{\mathrm{seen}}(\mathcal{P}^{\mathrm{b}}, \widehat{\mathcal{P}}^{\mathrm{b}}), \tag{5}$$

where $\mathcal{L}_{\mathrm{pro}}^{\mathrm{new}}$ focuses on learning knowledge from *new* classes, and $\mathcal{L}_{\mathrm{pro}}^{\mathrm{seen}}$ is dedicated to preserving learned knowledge of all *seen* classes. *This process is similar to a zero-sum game, and OPE aims to achieve the equilibrium to play a win-win game.* Concretely, as the model learns, the knowledge of new classes is gained and added to the prototypes over the memory bank $\mathcal{M}$, causing $\mathcal{L}_{\mathrm{pro}}^{\mathrm{seen}}$ gradually changes to the equilibrium that separates all seen classes well, including new ones. This variation is crucial to mitigate forgetting and is consistent with the goal of CIL.

3.4. Adaptive Prototypical Feedback

Although OPE can bring an overall equilibrium, it tends to treat each class *equally*. In fact, the degree of confusion varies among classes, and the model should focus purposefully on confused classes to consolidate learned knowledge. To this end, we propose Adaptive Prototypical Feedback (APF) with the feedback of online prototypes to sense the classes that are prone to be misclassified and then enhance their decision boundaries.

For each class pair in the memory bank $\mathcal{M}$, APF calculates the distances between online prototypes of all seen classes from the previous time step, showing the class confusion status by these distances. The closer the two prototypes are, the easier to be misclassified. Based on this analysis, our idea is to enhance the boundaries for those classes. Therefore, we convert the prototype distance matrix to a probability distribution P over the classes via a symmetric Gaussian kernel, defined as follows:

$$P_{i,j} \propto \exp(-\|\mathbf{p}_i^{\mathrm{b}} - \mathbf{p}_j^{\mathrm{b}}\|_2^2), \tag{6}$$

where $i, j \in \{1, \ldots, |\mathcal{P}^{\mathrm{b}}|\}$ and $i \ne j$. Then, all probabilities are normalized to a probability mass function that sums to one. APF returns probabilities to $\mathcal{M}$ for guiding the next sampling process and enhancing decision boundaries of easily misclassified classes.

Our adaptive prototypical feedback is implemented as a sampling-based mixup. Specifically, APF adaptively selects more samples from easily misclassified classes in $\mathcal{M}$

for mixup [54] according to the probability distribution P. Considering not over-penalizing the equilibrium of current online prototypes, we introduce a two-stage sampling strategy for replay data X^{b} of size m. First, we select n_{APF} samples with P, and a larger $P_{a,b}$ means more sampling from classes a and b. Here, $n_{\mathrm{APF}} = \alpha \cdot m$, and α is the ratio of APF. Second, the remaining $m - n_{\mathrm{APF}}$ samples are uniformly randomly selected from the entire memory bank to avoid the model only focusing on easily misclassified classes and disrupting the established equilibrium.

3.5. Overall Framework of OnPro

The overall structure of OnPro is shown in Fig. 2. OnPro comprises two key components based on proposed online prototypes: Online Prototype Equilibrium (OPE) and Adaptive Prototypical Feedback (APF). With the two components, the model can learn representative features against shortcut learning, and all seen classes maintain discriminative when learning new classes. However, classes may not be compact, because the online prototypes cannot cover full instance-level information. To further achieve intra-class compactness, we employ supervised contrastive learning [34] to learn instance-wise representations:

$$\begin{aligned}\mathcal{L}_{\mathrm{INS}} = &\sum_{i=1}^{2N} \frac{-1}{|I_i|} \sum_{j \in I_i} \log \frac{\exp\left(\operatorname{sim}(\mathbf{z}_i, \mathbf{z}_j)/\tau'\right)}{\sum\limits_{k \neq i} \exp\left(\operatorname{sim}(\mathbf{z}_i, \mathbf{z}_k)/\tau'\right)} \\ &+ \sum_{i=1}^{2m} \frac{-1}{|I_i^{\mathrm{b}}|} \sum_{j \in I_i^{\mathrm{b}}} \log \frac{\exp(\operatorname{sim}(\mathbf{z}_i^{\mathrm{b}}, \mathbf{z}_j^{\mathrm{b}})/\tau')}{\sum\limits_{k \neq i} \exp\left(\operatorname{sim}(\mathbf{z}_i^{\mathrm{b}}, \mathbf{z}_k^{\mathrm{b}})/\tau'\right)},\end{aligned} \tag{7}$$

where $I_i = \{j \in \{1, \ldots, 2N\} \mid j \neq i, y_j = y_i\}$ and $I_i^{\mathrm{b}} = \left\{j \in \{1, \ldots, 2m\} \mid j \neq i, y_j^{\mathrm{b}} = y_i^{\mathrm{b}}\right\}$ are the set of positive samples indexes to $\mathbf{z}_i$ and $\mathbf{z}_i^{\mathrm{b}}$, respectively. y_i^{b} is the class label of input x_i^{b} from X^{b}. N is the batch size of X. τ' is the temperature hyperparameter. The similarity function sim is computed in the same way as Eq. (9) in OCM [26].

Thus, the total loss of our OnPro framework is given as:

$$\mathcal{L}_{\mathrm{OnPro}} = \mathcal{L}_{\mathrm{OPE}} + \mathcal{L}_{\mathrm{INS}} + \mathcal{L}_{\mathrm{CE}}, \tag{8}$$

where $\mathcal{L}_{\mathrm{CE}} = \mathrm{CE}(y^{\mathrm{b}}, \varphi(f(\operatorname{aug}(x^{\mathrm{b}}))))$ is the cross-entropy loss; see the supplementary material for detailed training algorithms.

Following other replay-based methods [11, 43, 26], we update the memory bank in each time step by uniformly randomly selecting samples from X to push into $\mathcal{M}$ and, if $\mathcal{M}$ is full, pulling an equal number of samples out of $\mathcal{M}$.

4. Experiments

4.1. Experimental Setup

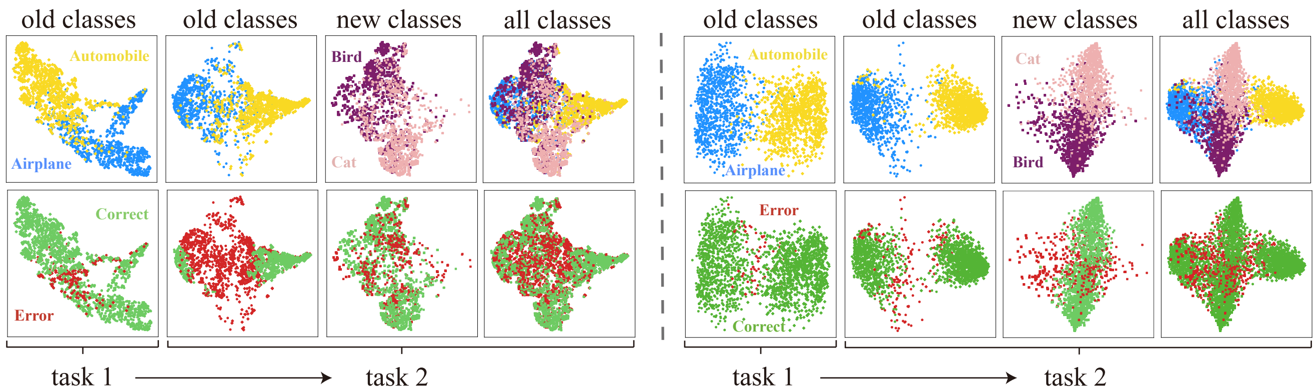
Datasets. We use three image classification benchmark datasets, including **CIFAR-10** [35], **CIFAR-100** [35], and **TinyImageNet** [37], to evaluate the performance of online CIL methods. Following [51, 43, 25], we split CIFAR-10 into 5 disjoint tasks, where each task has 2 disjoint classes, 10,000 samples for training, and 2,000 samples for testing, and split CIFAR-100 into 10 disjoint tasks, where each task has 10 disjoint classes, 5,000 samples for training, and 1,000 samples for testing. Following [26], we split TinyImageNet into 100 disjoint tasks, where each task has 2 disjoint classes, 1,000 samples for training, and 100 samples for testing. Note that the order of tasks is fixed in all experimental settings.

Baselines. We compare our OnPro with 13 baselines, including 10 replay-based online CL baselines: AGEM [10], MIR [2], GSS [3], ER [11], GDumb [47], ASER [51], SCR [43], CoPE [15], DVC [25], and OCM [26]; 3 offline CL baselines that use knowledge distillation by running them in one epoch: iCaRL [48], DER++ [4], and PASS [55]. Note that PASS is a non-exemplar method.

Evaluation metrics. We use Average Accuracy and Average Forgetting [51, 25] to measure the performance of our framework in online CIL. Average Accuracy evaluates the accuracy of the test sets from all seen tasks, defined as Average Accuracy $= \frac{1}{T}\sum_{j=1}^{T} a_{T,j}$, where $a_{i,j}$ is the accuracy on task j after the model is trained from task 1 to i. Average Forgetting represents how much the model forgets about each task after being trained on the final task, defined as Average Forgetting $= \frac{1}{T-1}\sum_{j=1}^{T-1} f_{T,j}$, where $f_{i,j} = \max_{k\in\{1,\ldots,i-1\}} a_{k,j} - a_{i,j}$.

Implementation details. We use ResNet18 [30] as the backbone f and a linear layer as the projection head g like [43, 26, 7]; the hidden dim in g is set to 128 as [12]. We also employ a linear layer as the classifier φ. We train the model from scratch with Adam optimizer and an initial learning rate of 5×10^{-4} for all datasets. The weight decay is set to 1.0×10^{-4}. Following [51, 25], we set the batch size N as 10, and following [26] the replay batch size m is set to 64. For CIFAR-10, we set the ratio of APF $\alpha = 0.25$. For CIFAR-100 and TinyImageNet, α is set to 0.1. The temperature $\tau = 0.5$ and $\tau' = 0.07$. For baselines, we also use ResNet18 as their backbone and set the same batch size and replay batch size for fair comparisons. We reproduce all baselines in the same environment with their source code and default settings; see the supplementary material for implementation details about all baselines. We report the average results across 15 runs for all experiments.

Data augmentation. Similar to data augmentations used in SimCLR [12], we use resized-crop, horizontal-flip, and gray-scale as our data augmentations. For all baselines, we



(a) ER (b) OnPro

Figure 3. t-SNE [53] visualizations of features learned from ER and OnPro on the test set of CIFAR-10. When learning new classes, ER suffers serious class confusion probably because shortcut learning. In contrast, OnPro significantly mitigates the forgetting.

Method	CIFAR-10			CIFAR-100			TinyImageNet		
	$M = 0.1k$	$M = 0.2k$	$M = 0.5k$	$M = 0.5k$	$M = 1k$	$M = 2k$	$M = 1k$	$M = 2k$	$M = 4k$
iCaRL [48]	31.0±1.2	33.9±0.9	42.0±0.9	12.8±0.4	16.5±0.4	17.6±0.5	5.0±0.3	6.6±0.4	7.8±0.4
DER++ [4]	31.5±2.9	39.7±2.7	50.9±1.8	16.0±0.6	21.4±0.9	23.9±1.0	3.7±0.4	5.1±0.8	6.8±0.6
PASS [55]	33.7±2.2	33.7±2.2	33.7±2.2	7.5±0.7	7.5±0.7	7.5±0.7	0.5±0.1	0.5±0.1	0.5±0.1
AGEM [10]	17.7±0.3	17.5±0.3	17.5±0.2	5.8±0.1	5.9±0.1	5.8±0.1	0.8±0.1	0.8±0.1	0.8±0.1
GSS [3]	18.4±0.2	19.4±0.7	25.2±0.9	8.1±0.2	9.4±0.5	10.1±0.8	1.1±0.1	1.5±0.1	2.4±0.4
ER [11]	19.4±0.6	20.9±0.9	26.0±1.2	8.7±0.3	9.9±0.5	10.7±0.8	1.2±0.1	1.5±0.2	2.0±0.2
MIR [2]	20.7±0.7	23.5±0.8	29.9±1.2	9.7±0.3	11.2±0.4	13.0±0.7	1.4±0.1	1.9±0.2	2.9±0.3
GDumb [47]	23.3±1.3	27.1±0.7	34.0±0.8	8.2±0.2	11.0±0.4	15.3±0.3	4.6±0.3	6.6±0.2	10.0±0.3
ASER [51]	20.0±1.0	22.8±0.6	31.6±1.1	11.0±0.3	13.5±0.3	17.6±0.4	2.2±0.1	4.2±0.6	8.4±0.7
SCR [43]	40.2±1.3	48.5±1.5	59.1±1.3	19.3±0.6	26.5±0.5	32.7±0.3	8.9±0.3	14.7±0.3	19.5±0.3
CoPE [15]	33.5±3.2	37.3±2.2	42.9±3.5	11.6±0.7	14.6±1.3	16.8±0.9	2.1±0.3	2.3±0.4	2.5±0.3
DVC [25]	35.2±1.7	41.6±2.7	53.8±2.2	15.4±0.7	20.3±1.0	25.2±1.6	4.9±0.6	7.5±0.5	10.9±1.1
OCM [26]	47.5±1.7	59.6±0.4	70.1±1.5	19.7±0.5	27.4±0.3	34.4±0.5	10.8±0.4	15.4±0.4	20.9±0.7
OnPro (**ours**)	**57.8**±1.1	**65.5**±1.0	**72.6**±0.8	**22.7**±0.7	**30.0**±0.4	**35.9**±0.6	**11.9**±0.3	**16.9**±0.4	**22.1**±0.4

Table 1. Average Accuracy (higher is better) on three benckmark datasets with different memory bank sizes M. All results are the average and standard deviation of 15 runs.

also use these augmentations. In addition, for DER++[4], SCR [43], and DVC [25], we follow their default settings and use their own extra data augmentations. OCM [26] uses extra rotation augmentations, which are also used in OnPro.

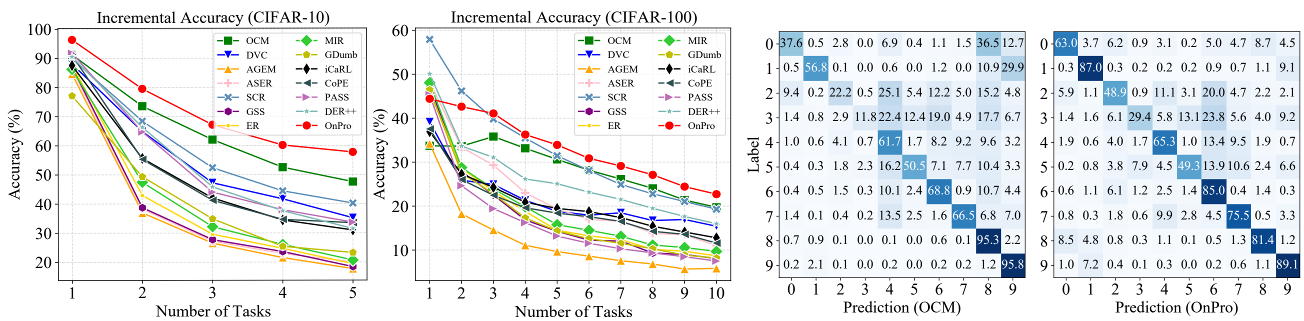
4.2. Motivation Justification

Shortcut learning in online CL. Shortcut learning is severe in online CL since the model cannot learn sufficient representative features due to the single-pass data stream. To intuitively demonstrate this issue, we conduct Grad-CAM++ [8] on the training set of CIFAR-10 ($M = 0.2k$) after the model is trained incrementally, as shown in Fig. 1. Each row in Fig. 1 represents a task with two classes. We can observe that although ER and DVC predict the correct class, the models actually take shortcuts and focus on some object-unrelated features. An interesting phenomenon is that ER tends to take shortcuts in each task. For example, ER learns the sky on both the airplane class in task 1 (the first row) and the bird class in task 2 (the second row) . Thus, ER forgets almost all the knowledge of the old classes. DVC maximizes the mutual information between instances like contrastive learning [12, 29], which only partially alleviates shortcut learning in online CL. In contrast, OnPro focuses on the representative features of the objects themselves. The results confirm that learning representative features is crucial against shortcut learning; see the supplementary material for more visual explanations.

Class confusion in online CL. Fig. 3 provides the t-SNE [53] visualization results for ER and OnPro on the test set of CIFAR-10 ($M = 0.2k$). We can draw intuitive observations as follows. (1) There is serious class confusion in ER. When the new task (task 2) arrives, features learned in task 1 are not discriminative for task 2, leading to class confusion and decreased performance in old classes. (2) Shortcut learning may cause class confusion. For example, the

Method	CIFAR-10			CIFAR-100			TinyImageNet		
	$M=0.1k$	$M=0.2k$	$M=0.5k$	$M=0.5k$	$M=1k$	$M=2k$	$M=1k$	$M=2k$	$M=4k$
iCaRL [48]	52.7±1.0	49.3±0.8	38.3±0.9	16.5±1.0	11.2±0.4	10.4±0.4	9.9±0.5	10.1±0.5	9.7±0.6
DER++ [4]	57.8±4.1	46.7±3.6	33.6±3.5	41.0±1.1	34.8±1.1	33.2±1.2	77.8±1.0	74.9±0.6	73.2±0.8
PASS [55]	21.2±2.2	21.2±2.2	21.2±2.2	10.6±0.9	10.6±0.9	10.6±0.9	27.0±2.4	27.0±2.4	27.0±2.4
AGEM [10]	64.8±0.7	64.8±0.7	64.5±0.5	41.7±0.8	41.8±0.7	41.7±0.6	73.9±0.7	73.1±0.7	72.9±0.5
GSS [3]	67.1±0.6	65.8±0.6	61.2±1.2	48.7±0.8	46.7±1.3	44.7±1.1	78.9±0.7	77.0±0.5	75.2±0.7
ER [11]	64.7±1.1	62.9±1.0	57.5±1.8	47.0±1.0	46.4±0.8	44.7±1.5	79.1±0.6	77.7±0.6	76.3±0.5
MIR [2]	62.6±1.0	58.5±1.4	51.1±1.1	45.7±0.9	44.2±1.3	42.3±1.0	75.3±0.9	71.5±1.0	66.8±0.8
GDumb [47]	28.5±1.4	28.4±1.0	28.1±1.0	25.0±0.4	23.2±0.4	20.7±0.3	22.7±0.3	18.4±0.2	17.0±0.2
ASER [51]	64.8±1.0	62.6±1.1	53.2±1.5	52.8±0.8	50.4±0.9	46.8±0.7	78.9±0.5	75.4±0.7	68.2±1.1
SCR [43]	43.2±1.5	35.5±1.8	24.1±1.0	29.3±0.9	20.4±0.6	11.5±0.6	44.8±0.6	26.8±0.5	20.1±0.4
CoPE [15]	49.7±1.6	45.7±1.5	39.4±1.8	25.6±0.9	17.8±1.3	14.4±0.8	11.9±0.6	10.9±0.4	9.7±0.4
DVC [25]	40.2±2.6	31.4±4.1	21.2±2.8	32.0±0.9	32.7±2.0	28.0±2.2	59.8±2.2	52.9±1.3	45.1±1.9
OCM [26]	35.5±2.4	23.9±1.4	13.5±1.5	18.3±0.9	15.2±1.0	10.8±0.6	23.6±0.5	26.2±0.5	23.8±1.0
OnPro (**ours**)	23.2±1.3	17.6±1.4	12.5±0.7	15.0±0.8	10.4±0.5	6.1±0.6	21.3±0.5	17.4±0.4	16.8±0.4

Table 2. Average Forgetting (lower is better) on three benckmark datasets. All results are the average and standard deviation of 15 runs.



(a) Average incremental performance

(b) Confusion matrix of OCM and OnPro

Figure 4. Incremental accuracy on tasks observed so far and confusion matrix of accuracy (%) in the test set of CIFAR-10.

performance of ER decreases more on airplanes compared to automobiles, probably because birds in the new task have more similar backgrounds to airplanes, as shown in Fig. 1. (3) OnPro achieves better discrimination both on task 1 and task 2. The results demonstrate that OnPro can maintain discrimination of all seen classes and significantly mitigate forgetting by combining the proposed OPE and APF.

4.3. Results and Analysis

Performance of average accuracy. Table 1 presents the results of average accuracy with different memory bank sizes (M) on three benchmark datasets. Our OnPro consistently outperforms all baselines on three datasets. Remarkably, the performance improvement of OnPro is more significant when the memory bank size is relatively small; this is critical for online CL with limited resources. For example, compared to the second-best method OCM, OnPro achieves about 10% and 6% improvement on CIFAR-10 when M is 100 and 200, respectively. The results show that our OnPro can learn more representative and discriminative features with a limited memory bank. Compared to baselines that use knowledge distillation (iCaRL, DER++, PASS, OCM), our OnPro achieves better performance by leveraging the feedback of online prototypes. Besides, OnPro significantly outperforms PASS and CoPE that also use prototypes, showing that online prototypes are more suitable for online CL.

We find that the performance improvement tends to be gentle when M increases. The reason is that as M increases, the samples in the memory bank become more diverse, and the model can extract sufficient information from massive samples to distinguish seen classes. In addition, many baselines perform poorly on CIFAR-100 and TinyImageNet due to a dramatic increase in the number of tasks. In contrast, OnPro still performs well and improves accuracy over the second best.

Performance of average forgetting. We report the Average Forgetting results of our OnPro and all baselines on three benchmark datasets in Table 2. The results confirm that OnPro can effectively mitigate catastrophic forgetting. For CIFAR-10 and CIFAR-100, OnPro achieves the lowest average forgetting compared to all replay-based baselines. For TinyImageNet, our result is a little higher than iCaRL

Method	CIFAR-10 Acc ↑(Forget ↓)	CIFAR-100 Acc ↑(Forget ↓)
baseline	46.4±1.2(36.0±2.1)	18.8±0.8(18.5±0.7)
w/o OPE	53.1±1.4(24.7±2.0)	19.3±0.7(15.9±0.9)
w/o APF	52.0±1.5(34.6±2.4)	21.5±0.5(16.3±0.8)
w/o $\mathcal{L}_{\text{pro}}^{\text{new}}$	54.8±1.2(**22.1**±3.0)	19.6±0.8(19.9±0.7)
w/o $\mathcal{L}_{\text{pro}}^{\text{seen}}$	55.7±1.4(25.5±1.5)	20.1±0.4(16.2±0.6)
$\mathcal{L}_{\text{pro}}^{\text{seen}}$ w/o $\mathcal{C}^{\text{new}}$	56.2±1.2(26.4±2.3)	20.8±0.6(17.9±0.7)
OnPro **(ours)**	**57.8**±1.1(23.2±1.3)	**22.7**±0.7(**15.0**±0.8)

Table 3. Ablation studies on CIFAR-10 ($M = 0.1k$) and CIFAR-100 ($M = 0.5k$). "baseline" means $\mathcal{L}_{\text{INS}} + \mathcal{L}_{\text{CE}}$. "$\mathcal{L}_{\text{pro}}^{\text{seen}}$ w/o $\mathcal{C}^{\text{new}}$" means $\mathcal{L}_{\text{pro}}^{\text{seen}}$ do not consider new classes in current task.

and CoPE but better than the latest methods DVC and OCM. The reason is that iCaRL uses a nearest class mean classifier, but we use softmax and FC layer during the test phase, and CoPE slowly updates prototypes with a high momentum. However, as shown in Table 1, OnPro provides more accurate classification results than iCaRL and CoPE. It is a fact that when the maximum accuracy of a task is small, the forgetting on this task is naturally rare, even if the model completely forgets what it learned.

Performance of each incremental step. We evaluate the average incremental performance [4, 25] on CIFAR-10 ($M = 0.1k$) and CIFAR-100 ($M = 0.5k$), which indicates the accuracy over all seen tasks at each incremental step. Fig. 4a shows that OnPro achieves better accuracy and effectively mitigates forgetting while the performance of most baselines degrades rapidly with the arrival of new classes.

Confusion matrices at the end of learning. We report the confusion matrices of our OnPro and the second-best method OCM, as shown in Fig. 4b. After learning the last task (*i.e.*, the last two classes), OCM forgets the knowledge of early tasks (classes 0 to 3). In contrast, OnPro performs relatively well in all classes, especially in the first task (classes 0 and 1), outperforming OCM by 27.8% average improvements. The results show that learning representative and discriminative features is crucial to mitigate catastrophic forgetting; see the supplementary material for extra experimental results.

4.4. Ablation Studies

Effects of each component. Table 3 presents the ablation results of each component. Obviously, OPE and APF can consistently improve the average accuracy of classification. We can observe that the effect of OPE is more significant on more tasks while APF plays a crucial role when the memory bank size is limited. Moreover, when combining OPE and APF, the performance is further improved, which indicates that both can benefit from each other. For example,

Method	$M = 0.1k$	$M = 0.2k$	$M = 0.5k$
Random	53.5±2.7	62.9±2.5	70.8±2.2
APF **(ours)**	**57.8**±1.1	**65.5**±1.0	**72.6**±0.8

Table 4. Comparison of Random Mixup and APF on CIFAR-10.

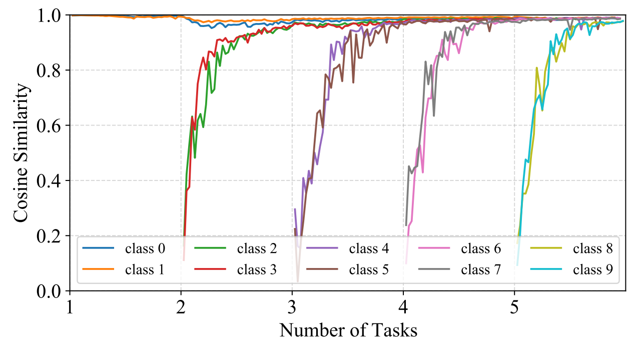


Figure 5. The cosine similarity between online prototypes and prototypes of the entire memory bank.

APF boosts OPE by about 6% improvements on CIFAR-10 ($M = 0.1k$), and the performance of APF is improved by about 3% on CIFAR-100 ($M = 0.5k$) by combining OPE.

Equilibrium in OPE. When learning new classes, the data of new classes is involved in both $\mathcal{L}_{\text{pro}}^{\text{new}}$ and $\mathcal{L}_{\text{pro}}^{\text{seen}}$ of OPE, where $\mathcal{L}_{\text{pro}}^{\text{new}}$ only focuses on learning new knowledge while $\mathcal{L}_{\text{pro}}^{\text{seen}}$ tends to alleviate forgetting on seen classes. To explore the best way of learning new classes, we consider three scenarios for OPE in Table 3. The results show that only learning new knowledge (w/o $\mathcal{L}_{\text{pro}}^{\text{seen}}$) or only consolidating the previous knowledge (w/o $\mathcal{L}_{\text{pro}}^{\text{new}}$) can significantly degrade the performance, which indicates that both are indispensable for online CL. Furthermore, when $\mathcal{L}_{\text{pro}}^{\text{seen}}$ only considers old classes and ignores new classes ($\mathcal{L}_{\text{pro}}^{\text{seen}}$ w/o $\mathcal{C}^{\text{new}}$), the performance also decreases. These results show that the equilibrium of all seen classes (OPE) can achieve the best performance and is crucial for online CL.

Effects of APF. To verify the advantage of APF, we compare it with the completely random mixup in Table 4. APF outperforms random mixup in all three scenarios. Notably, APF works significantly when the memory bank size is small, which shows that the feedback can prevent class confusion due to a restricted memory bank; see the supplementary material for extra ablation studies.

4.5. Validation of Online Prototypes

Fig. 5 shows the cosine similarity between online prototypes and global prototypes (prototypes of the entire memory bank) at each time step. For the first mini-batch of each task, online prototypes are equal to global prototypes (similarity is 1, omitted in Fig. 5). In the first task, online and global prototypes are updated synchronously with the model updates, resulting in high similarity. In subsequent tasks, the model initially learns inadequate features of new classes, causing online prototypes to be inconsistent with global prototypes and low similarity, which shows that accumulating early features as prototypes may be harmful to new tasks. However, the similarity will improve as the model learns, because the model gradually learns representative features of new classes. Furthermore, the similarity on old classes is only slightly lower, showing that online prototypes are resistant to forgetting.

5. Conclusion

This paper identifies shortcut learning as the key limiting factor for online CL, where the learned features are biased and not generalizable to new tasks. It also sheds light on why the online learning models fail to generalize well. Based on these, we present a novel online prototype learning (OnPro) framework to address shortcut learning and mitigate catastrophic forgetting. Specifically, by taking full advantage of introduced online prototypes, the proposed OPE aims to learn representative features of each class and discriminative features between classes for achieving an equilibrium status that separates all seen classes well when learning new classes, while the proposed APF is able to sense easily misclassified classes and enhance their decision boundaries with the feedback of online prototypes. Extensive experimental results on widely-used benchmark datasets validate the effectiveness of the proposed OnPro as well as its components. In the future, we will try more efficient alternatives, such as designing a margin loss to ensure discrimination between classes further.

Acknowledgement This work was supported in part by STI2030-Major Projects (No. 2021ZD0200204), National Natural Science Foundation of China (Nos. 62101136 and 62176059), Shanghai Municipal Science and Technology Major Project (No. 2018SHZDZX01), ZJ Lab, Shanghai Municipal of Science and Technology Project (No. 20JC1419500), and Shanghai Center for Brain Science and Brain-inspired Technology.

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. 1, 2

[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*, volume 32, 2019. 1, 2, 5, 6, 7

[3] Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. *Advances in Neural Information Processing Systems*, 32, 2019. 1, 2, 5, 6, 7

[4] Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. *Advances in Neural Information Processing Systems*, 33:15920–15930, 2020. 2, 3, 5, 6, 7, 8

[5] Lucas Caccia, Rahaf Aljundi, Nader Asadi, Tinne Tuytelaars, Joelle Pineau, and Eugene Belilovsky. New insights on reducing abrupt representation change in online continual learning. *arXiv:2203.03798*, 2022. 1, 3

[6] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 33:9912–9924, 2020. 3

[7] Hyuntak Cha, Jaeho Lee, and Jinwoo Shin. Co^{2}L: Contrastive continual learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9516–9525, 2021. 2, 3, 5

[8] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-CAM++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 839–847, 2018. 6

[9] Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 532–547, 2018. 1, 2, 4

[10] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with A-GEM. *arXiv:1812.00420*, 2018. 2, 5, 6, 7

[11] 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. *arXiv:1902.10486*, 2019. 1, 2, 5, 6, 7

[12] 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, 2020. 3, 5, 6

[13] Aristotelis Chrysakis and Marie-Francine Moens. Online continual learning from imbalanced data. In *International Conference on Machine Learning*, pages 1952–1961, 2020. 1

[14] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, et al. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3366–3385, 2021. 1

[15] Matthias De Lange and Tinne Tuytelaars. Continual prototype evolution: Learning online from non-stationary data streams. In *Proceedings of the IEEE/CVF International Con-*

ference on Computer Vision, pages 8250–8259, 2021. 3, 5, 6, 7

[16] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv:2010.11929*, 2020. 1

[17] Zhiyuan Fang, Jianfeng Wang, Lijuan Wang, Lei Zhang, Yezhou Yang, and Zicheng Liu. Seed: Self-supervised distillation for visual representation. *arXiv:2101.04731*, 2021. 2

[18] Chrisantha Fernando, Dylan Banarse, Charles Blundell, Yori Zwols, David Ha, Andrei A. Rusu, Alexander Pritzel, and Daan Wierstra. PathNet: Evolution channels gradient descent in super neural networks. *arXiv 1701.08734*, 2017. 2

[19] Enrico Fini, Victor G Turrisi Da Costa, Xavier Alameda-Pineda, Elisa Ricci, Karteek Alahari, and Julien Mairal. Self-supervised models are continual learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9621–9630, 2022. 1

[20] Robert M French. Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, 3(4):128–135, 1999. 1

[21] Jhair Gallardo, Tyler L Hayes, and Christopher Kanan. Self-supervised training enhances online continual learning. *arXiv:2103.14010*, 2021. 3

[22] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020. 2

[23] Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv:1312.6211*, 2013. 1

[24] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 33:21271–21284, 2020. 3

[25] Yanan Gu, Xu Yang, Kun Wei, and Cheng Deng. Not just selection, but exploration: Online class-incremental continual learning via dual view consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7442–7451, 2022. 1, 5, 6, 7, 8

[26] Yiduo Guo, Bing Liu, and Dongyan Zhao. Online continual learning through mutual information maximization. In *International Conference on Machine Learning*, pages 8109–8126, 2022. 1, 2, 3, 5, 6, 7

[27] Jiangpeng He, Runyu Mao, Zeman Shao, and Fengqing Zhu. Incremental learning in online scenario. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13926–13935, 2020. 1

[28] Jiangpeng He and Fengqing Zhu. Exemplar-free online continual learning. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 541–545, 2022. 1, 3

[29] 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, 6

[30] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 1, 5

[31] Xu He and Herbert Jaeger. Overcoming catastrophic interference using conceptor-aided backpropagation. In *International Conference on Learning Representations*, 2018. 2

[32] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv:1503.02531*, 2015. 2

[33] Zhizhong Huang, Jie Chen, Junping Zhang, and Hongming Shan. Learning representation for clustering via prototype scattering and positive sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(6):7509–7524 2023. 3

[34] 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*, 33:18661–18673, 2020. 2, 5

[35] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 5

[36] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017. 1

[37] Ya Le and Xuan Yang. Tiny ImageNet visual recognition challenge. *CS 231N*, 7(7):3, 2015. 5

[38] 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

[39] 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. 2

[40] Junnan Li, Pan Zhou, Caiming Xiong, and Steven C. H. Hoi. Prototypical contrastive learning of unsupervised representations. In *International Conference on Learning Representations*, 2021. 3

[41] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, 2017. 1, 2, 3

[42] Zheda Mai, Ruiwen Li, Jihwan Jeong, David Quispe, Hyunwoo Kim, and Scott Sanner. Online continual learning in image classification: An empirical survey. *Neurocomputing*, 469:28–51, 2022. 1

[43] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 3589–3599, 2021. 1, 2, 3, 5, 6, 7

[44] Marc Masana, Xialei Liu, Bartłomiej Twardowski, Mikel Menta, Andrew D Bagdanov, and Joost van de Weijer. Class-incremental learning: survey and performance evaluation on image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 1

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

[46] 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. 1

[47] Ameya Prabhu, Philip HS Torr, and Puneet K Dokania. GDumb: A simple approach that questions our progress in continual learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 524–540, 2020. 2, 5, 6, 7

[48] 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. 2, 3, 5, 6, 7

[49] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv:1606.04671*, 2016. 2

[50] Joan Serra, Didac Suris, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. In *International Conference on Machine Learning*, pages 4548–4557, 2018. 1, 2

[51] Dongsub Shim, Zheda Mai, Jihwan Jeong, Scott Sanner, Hyunwoo Kim, and Jongseong Jang. Online class-incremental continual learning with adversarial Shapley value. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 9630–9638, 2021. 1, 2, 5, 6, 7

[52] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*, 2014. 1

[53] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(11), 2008. 6

[54] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv:1710.09412*, 2017. 5

[55] Fei Zhu, Xu-Yao Zhang, Chuang Wang, Fei Yin, and Cheng-Lin Liu. Prototype augmentation and self-supervision for incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5871–5880, 2021. 2, 3, 5, 6, 7