Improving Plasticity in Online Continual Learning via Collaborative Learning

Anonymous CVPR submission

Paper ID 7545

Abstract

Online Continual Learning (CL) solves the problem of learning the ever-emerging new classification tasks from a continuous data stream. Unlike its offline counterpart, in online CL, the training data can only be seen once. Most existing online CL research regards catastrophic forgetting (i.e., model stability) as almost the only challenge. In this paper, we argue that the model's capability to acquire new knowledge (i.e., model plasticity) is another challenge in online CL. While replay-based strategies have been shown to be effective in alleviating catastrophic forgetting, there is a notable gap in research attention toward improving model plasticity. To this end, we propose Collaborative Continual Learning (CCL), a collaborative learning based strategy to improve the model's capability in acquiring new concepts. Additionally, we introduce Distillation Chain (DC), a novel collaborative learning scheme to boost the training of the models. We adapted CCL-DC to existing representative online CL works. Extensive experiments demonstrate that even if the learners are well-trained with state-of-the-art online CL methods, our strategy can still improve model plasticity dramatically, and thereby improve the overall performance by a large margin. The source code is included in the supplementary material and will be publicly available upon acceptance.

1. Introduction

Continual Learning (CL) [11, 14, 35, 47] aims to learn a sequence of tasks incrementally and encourage the neural network to gain more performance on the tasks at hand, without forgetting heretofore learned knowledge. CL can be done in two different manners [4, 47]: *offline* continual learning and *online* continual learning. In offline CL, the learner can have infinite access to all the training data of the current task it trains on and may go through the data for any epoch. Contrary to offline CL, in online CL, the training data for each task also comes continually in a data stream, and the learner can only see the training data once. Apart from the learning manner, there are also three different CL scenarios [25, 32, 45]: Task-incremental Learning (TIL), Domain-incremental Learning (DIL), and Class-incremental learning (CIL). In this paper, we focus on the CIL setting in online CL.

Various online CL methods [6, 7, 21, 22, 38, 48] have been proposed to help the models learn continually on one-epoch data stream, with alleviated forgetting. Among them, replay-based methods have shown remarkable success, and current state-of-the-art methods rely heavily on memory replay to mitigate catastrophic forgetting [19, 33]. However, while most existing online CL research almost only focuses on improving model stability (*i.e.*, alleviating catastrophic forgetting) in pursuit of better overall accuracy, the importance of model plasticity (*i.e.*, the capability to acquire new knowledge) is greatly overlooked. Contrary to offline CL, where it is possible to gain high plasticity by iterating several epochs on the current task before proceeding to the subsequent task, the plasticity in online CL is more important because the training data can only be seen once. As shown in Fig. 1, compared to learning without memory replay, the replay-based methods implicitly alleviate the low plasticity issue to some extent. Also, it is possible to improve the plasticity with multiple updates trick on incoming samples [3]. However, the combination of memory replay and multiple updates does not bridge the plasticity gap, and multiple updates trick will also lead to higher catastrophic forgetting. Overall, the plasticity gap hinders the performance of online CL methods.

In this paper, we claim that besides stability, the model's ability to acquire new knowledge (*i.e.,* model plasticity) is also vital in order to have a good overall accuracy. To shed light on how model plasticity and stability will impact the overall performance, we propose a quantitative link between plasticity, stability, and final accuracy, showing that both plasticity and stability play crucial roles in the overall performance.

Guided by the quantitative relationship, we focus ourselves on the former-overlooked plasticity perspective. Inspired by the ability of collaborative learning to accelerate the convergence in non-continual scenarios [5], we incorporated collaborative learning in online CL and observed

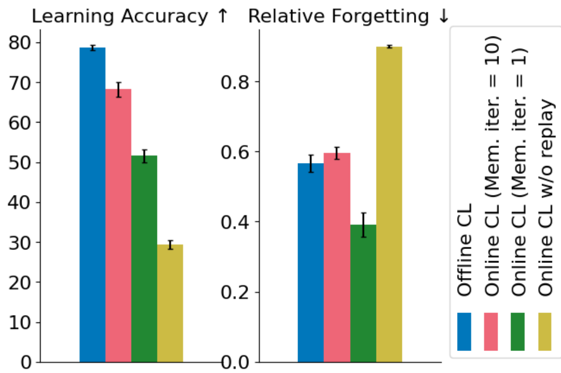


Figure 1. The plasticity (learning accuracy) and stability (relative forgetting, our metric proposed in Sec. 3) comparison of ER under different settings on CIFAR-100. For experiments with memory replay, the size of the memory buffer is set to 2,000. We can witness a plasticity gap between offline CL and online CL, even with memory replay and multiple update trick (memory iteration > 1).

a similar phenomenon. To this end, we propose Collaborative Continual Learning with Distillation Chain (CCL-DC), a collaborative learning scheme that can be adapted to existing online CL methods. CCL-DC comprises two key components: Collaborative Continual Learning (CCL) and Distillation Chain (DC).

CCL involves two peer continual learners to learn from the data stream simultaneously in a peer teaching manner, and it enables us to have more parallelism in optimization and provides more maneuverability to the continual learners. To the best of our knowledge, CCL is the first to involve collaborative learning techniques in online CL research. Moreover, to fully exploit the potential of collaborative learning in online CL scenarios, we proposed DC, an entropy regularization based optimization strategy explicitly designed for online CL.

The main contribution of this paper can be summarized as follows.

1. We identify two important challenges in training online continual learners: plasticity and stability. Moreover, we propose a quantitative link between plasticity, stability, and final performance. Based on this, we find that plasticity is an important obstacle in online CL, which was greatly overlooked in the previous research;
2. To overcome the plasticity issue, we propose CCL-DC, a collaborative learning based strategy that can be seamlessly integrated into the existing methods and improve their performance by enhancing plasticity;
3. Extensive experiments show that CCL-DC can enhance the performance of existing methods by a large margin.

2. Related Work

Continual Learning. Continual Learning methods can be classified into three different categories: regularization-based methods, parameter-isolation-based methods, and replay-based methods. Regularization-based methods [2, 9, 26, 29, 50] add extra regularization terms to balance the old and new tasks. Parameter-isolation-based methods [1, 17, 39–41] solve the problem explicitly by dynamically allocating task-specific parameters. Replay-based methods [6, 7, 10, 15, 21, 22, 34, 38, 48] maintain a small memory buffer that stores a few old training samples.

Among these methods, replay-based strategies have gained huge success due to their impressive performance and simplicity. ER [38] is the fundamental replay-based method that leverages Cross-Entropy loss for classification and a random replay buffer. DER++ [6] stores the logits in the memory buffer and extends ER with the distillation of old stored logits. ER-ACE [7] extends ER with Asymmetric Cross-Entropy loss for classification to suppress the drift of old class representations. OCM [21] leverages a replay-based strategy by maximizing the mutual information between old and new class representations. GSA [22] solves cross-task class discrimination with replay-based strategy and Gradient Self Adaption. OnPro [48] uses online prototype learning to address shortcut learning and alleviate catastrophic forgetting.

These replay-based methods propose different strategies for alleviating catastrophic forgetting and improving the model stability. However, the importance of the model plasticity is greatly neglected in their research, despite their success in terms of final performance. In our work, these methods serve as the baselines and we adapted our strategy to these baselines to show the efficiency of our proposed approach.

Collaborative Learning. Collaborative learning [5, 20, 42, 51, 52] orients from online knowledge distillation (KD). Different from the conventional KD methods, online KD trains a cohort of deep networks from scratch in a peer-teaching manner. During the training process, the model imitates their peers and guides the training of other models simultaneously. DML [51] suggests peer student models learn from each other through the logit distillation between the probability distributions. Codistillation [5] is similar to DML and suggests the ensemble of peer networks can further improve the performance. More importantly, Codistillation shows that online KD can help the model converge faster on non-continual scenarios.

Despite the success of collaborative learning in non-continual scenarios, due to the lack of focus on plasticity, the research on collaborative learning in CL is still limited. To the best of our knowledge, there is no existing research using the collaborative learning technique to boost the training of online CL. Moreover, in our work, we propose DC, an entropy regularization based optimization strategy, which is designed to exploit the full potential of collaborative learning in online CL scenarios.

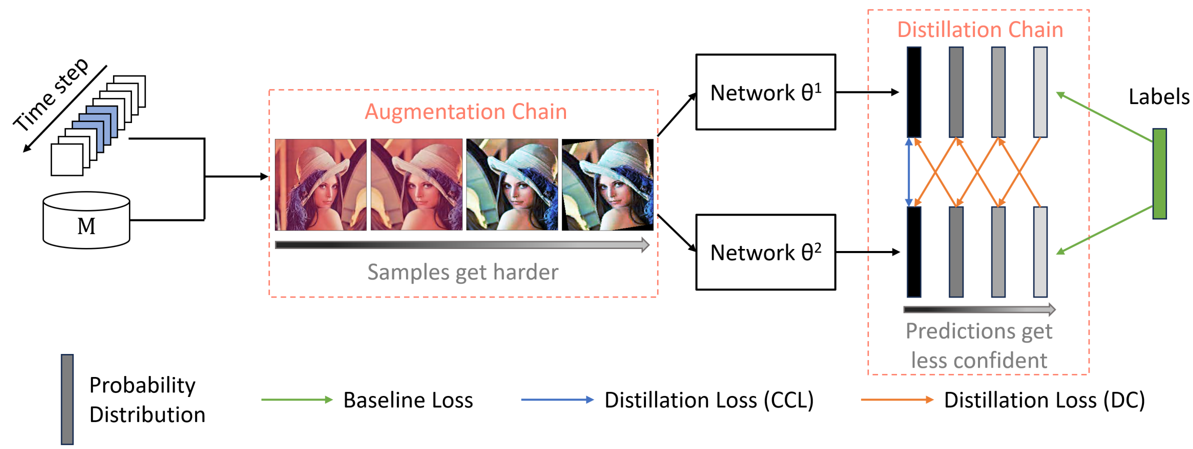


Figure 2. Overview of the proposed CCL-DC framework applied to a baseline online CL method. The proposed CCL-DC framework has two main components. The first one is CCL, which involves two peer continual learners that simultaneously learn from the data stream in a peer teaching manner. The second component is DC, which generates a chain of samples with varying levels of difficulty and feeds them to models to obtain a chain of logit distribution of different confidence levels. Then, in a collaborative learning approach, DC conducts distillation from *less confident* predictions to *more confident* predictions, to serve as a learned entropy regularization.

3. Plasticity and Stability in online CL

In this section, we revise the metric for model plasticity and propose a novel metric for model stability. In addition, we quantitatively derived the impact of model plasticity and stability on the final performance.

3.1. Model Plasticity

The model plasticity measures the learner's capability to learn new knowledge when a new task arrives. Several different metrics have been proposed to measure the model plasticity [9, 30, 37, 47]. In our work, we evaluate the model plasticity with Learning Accuracy (LA) [37]. Formally, the Learning Accuracy for the j-th task is defined as:

$$l_j = a_j^j, \tag{1}$$

where a_j^i is the accuracy evaluated on the test set of task j after training the network from task 1 to task i. For an overall metric normalized against all tasks, the averaged Learning Accuracy is written as $LA = \frac{1}{T}\sum_{j=1}^{T} l_j$, and T is the number of tasks in total.

3.2. Model Stability

The stability measures how much the model forgets given its current state. The most commonly used metric in previous CL research is the Forgetting Measure (FM) [9]. Intuitively, FM for the j-th task fm_j^k reveals how much performance the model loses on a given task j, after training on task k, compared with its maximum performance obtained in the past:

$$fm_j^k = \max_{i\in\{1,\dots,k-1\}} (a_j^i - a_j^k), \forall j < k. \tag{2}$$

For the overall metric obtained across all tasks, FM can be expressed as:

$$FM = \frac{1}{T-1}\sum_{j=1}^{T-1} fm_j^T. \tag{3}$$

In our work, instead of using FM as the stability metric, we propose forgetting measure on a relative basis, which we call Relative Forgetting (RF). There are two reasons for shifting from absolute forgetting to relative forgetting:

1. RF is more fair for methods with higher plasticity. This is because methods with poor plasticity will never have a large FM, and FM is capped by the maximum performance obtained by the learner in the past. Moreover, even if two learners lose the same absolute performance, the more plastic learner can still be regarded as forgetting less, because it has a higher peak performance and loses less proportion of its performance;
2. RF helps quantitatively derive the relationship between the model stability and final performance.

Intuitively, RF measures how much *proportion* of performance the model forgets. And RF for the j-th task after training on task k, can be defined as:

$$f_j^k = \max_{i\in\{1,\dots,k\}} \left(1 - \frac{a_j^k}{a_j^i}\right), \forall j \le k. \tag{4}$$

The overall metric averaged across all tasks can be written as:

$$RF = \frac{1}{T}\sum_{j=1}^{T} f_j^T. \tag{5}$$

3.3. Impact on the Overall Performance

For online CL, the model's final average accuracy (AA) is the most vital metric. In this subsection, we try to show how the model plasticity and stability will impact the final performance quantitatively.

The model's final average accuracy can be calculated by:

$$AA = \frac{1}{T}\sum_{j=1}^{T} a_j^T. \tag{6}$$

With the definition of our plasticity metric (LA) and stability metric (RF), we can easily find the relationship between learning accuracy, relative forgetting, and accuracy:

$$a_j^i \geq l_j \times (1 - f_j^i), \tag{7}$$

where we take the equal sign when $a_j^j = \max_{i\in\{1,...,i\}} a_j^i$. When generalizing the class-wise final accuracy a_j^T to the average accuracy (AA), we need to take the dot product of the LA vector $[l_1, ..., l_T]$ with RF vector $[f_1^T, ..., f_T^T]$ which is trivial. More intuitively, in practice, we can make the approximation with:

$$AA \gtrapprox LA \times (1 - RF). \tag{8}$$

As indicated by Eq. 8, the lower bound of the final performance is proportional to LA and $1-RF$, which suggests that both plasticity (LA) and stability (RF) play a crucial role in the final accuracy. Our findings reveal the importance of the model plasticity which was neglected in the past. And it can serve as a good guide for future online CL research.

4. Proposed Method

In this section, we first justify our motivation with the findings in Sec. 3. Then, we introduce our proposed strategy: Collaborative Continual Learning and Distillation Chain. Finally, we show how to adapt our proposed strategy to the existing online CL methods and boost their plasticity.

4.1. Motivation Justification

Online continual learners aim to continuously adapt to non-stationary data streams, efficiently acquiring new knowledge while retaining previously learned information. In current online CL research, almost all of the efforts focus on alleviating catastrophic forgetting, and the importance of "learning capability" on new knowledge is greatly overlooked. However, our finding in Sec. 3 shows both plasticity and stability play important roles in achieving decent final performance.

While replay-based methods were originally designed to tackle forgetting issues, Fig. 1 demonstrates that they can implicitly mitigate the plasticity gap. Nonetheless, as shown in Fig. 1, the limited plasticity is still a significant barrier to the performance, even with replay. To this end, we explicitly focus on the plasticity perspective.

The potential of collaborative learning to improve convergence in non-continual scenarios [5] positions it as a promising candidate for enhancing plasticity. With the apparent lack of focus on plasticity, collaborative learning has yet to be leveraged to boost convergence of online continual learners. In our research, we propose to exploit collaborative learning convergence properties for improving plasticity. We find that similar to non-continual scenarios, collaborative learning strategy can boost convergence by allowing more parallelism in the training and more maneuverability of the continual learners. Moreover, to fully take advantage of collaborative learning, we also propose Distillation Chain (DC), an entropy regularization based optimization strategy in collaborative learning specifically designed for online CL.

4.2. Collaborative Continual Learning

The introduced Collaborative Continual Learning (CCL) enables more parallelism and flexibility in training online continual learners, and it is the key to improving the model plasticity and the final performance. As shown in Fig. 2, CCL involves two peer continual learners of the same architecture and optimizer setting training in a peer-teaching manner. In the training phase, networks are supervised with both the ground truth label and the predictions of their peers. In the inference phase, models can either make predictions collaboratively with ensemble methods [5] to get a better performance or predict independently for the sake of computation efficiency. If we denote two networks in CCL as θ^1 and θ^2, we formulate our loss to network θ^1 as:

$$\begin{aligned}\mathcal{L}^1_{CCL} =& \lambda_1 \cdot \mathcal{L}_{cls}(\theta^1(X), y) \\ &+ \lambda_2 \cdot D_{KL}(\theta^1(X)/\tau, \theta^2(X)/\tau),\end{aligned} \tag{9}$$

where (X, y) is the data-label pair, $\mathcal{L}_{cls}(\cdot)$ is the classification loss in the baseline method CCL adapts to, $D_{KL}(\cdot)$ is the Kullback-Leibler divergence, λ_1 and λ_2 are balancing hyperparameters and τ is the temperature hyperparameter. Note that the network θ^2 should be trained with $\mathcal{L}^2_{CCL}$, respectively.

4.3. Distillation Chain

To fully take advantage of CCL, we propose Distillation Chain (DC), an entropy regularizaion based strategy explicitly designed for online CL. As illustrated in Fig. 2, DC comprises two steps: (1) generating a chain of samples with different levels of difficulty [43] using data augmentation, and (2) distillation of logit distribution from *harder* samples to *easier* samples in a collaborative learning way.

The main motivation of DC originates from the idea of entropy regularization-based optimization strategies, like label smoothing [44], knowledge distillation [49], and confidence penalty [36], where we find that overconfidence will hurt performance in non-continual scenarios. As shown in the supplementary material, we observed a similar phenomenon in online CL. To tackle the problem, DC uses data augmentation strategies to generate samples with different levels of difficulty and produces logit distribution with different confidence. The distillation from *less confident* predictions to *more confident* predictions weakens the overall confidence of the network and benefits the performance by improving the generalization capability.

In our work, we use a geometric distortion comprised of RandomCrop and RandomHorizontalFlip as the first step of DC augmentation. After that, we use RandAugment [13] for the subsequent augmentations and we involve two hyperparameters N and M for RandAugment. We take three augmentation steps and distill the logit distribution from the teacher with *harder* samples to the student with *easier* samples. We formulate our loss with DC to network θ^1 as:

$$\begin{aligned} \mathcal{L}^1_{DC} =& \lambda_1 \sum_{i=1}^{3} \mathcal{L}_{cls}(\theta^1(X_i), y) \\ &+ \lambda_2 \sum_{i=1}^{3} D_{KL}(\theta^1(X_{i-1})/\tau, \theta^2(X_i)/\tau), \end{aligned} \tag{10}$$

where X_i is the augmentation of input sample X after i augmentation steps. More discussion about why and how DC works can be found in the supplementary material.

4.4. Apply CCL-DC to online CL methods

The overall loss to network θ^1 when adapting CCL-DC can be written as:

$$\mathcal{L}^1 = \mathcal{L}_{Baseline} + \mathcal{L}^1_{CCL} + \mathcal{L}^1_{DC}, \tag{11}$$

where $\mathcal{L}_{Baseline}$ is the loss function of the baseline model CCL-DC adapts to. Note that the model θ^2 should be trained similarly. In Algorithm 1, we provide a Pytorch-like pseudo-code demonstrating how to incorporate CCL-DC into a given baseline. For simplicity, we only show the loss function for model θ^1. Also, we omitted the memory buffer in the pseudo-code. However, the training should be consistent with the baseline, using both streaming and memory data.

5. Experiments

5.1. Experimental Setup

Datasets. We use four image classification benchmark datasets to evaluate the effectiveness of our method, including CIFAR-10 [27], CIFAR-100 [27], TinyImageNet [28], and ImageNet-100 [24]. More detailed information about the dataset split and task allocation is given in the supplementary material.

Algorithm 1 PyTorch-like pseudo-code of CCL-DC to integrate to other baselines.

```
# model1: student model
# model2: teacher model
# optim1: optimizer for student model
# cls: classification loss in baseline
for x, y in dataloader:
  # Baseline loss
  loss_baseline = criterion_baseline(model1, x, y)

  # DC Augmentation
  x1 = geometric_distortion(x)
  x2 = RandAugment(x1, N, M)
  x3 = RandAugment(x2, N, M)

  # CCL-DC loss
  ls, ls1, ls2, ls3 = model1(x, x1, x2, x3)
  lt, lt1, lt2, lt3 = model2(x, x1, x2, x3) # no grad

  loss_cls = cls(ls, y) + cls(ls1, y) + cls(ls2, y) +
  ↪  cls(ls3, y)
  loss_ccl = kl_div(ls/t, lt/t) # temperature t
  loss_dc = kl_div(ls/t, lt1/t) + kl_div(ls1/t, lt2/t) +
  ↪  kl_div(ls2/t, lt3/t)

  loss_ours = lam1*loss_cls + lam2*(loss_ccl + loss_dc)
  loss = loss_baseline + loss_ours

  optim1.zero_grad()
  loss.backward()
  optim1.step()
```

Baselines. To show the effectiveness of our strategy, we applied CCL-DC to six typical and state-of-the-art online CL methods, including ER [38], DER++ [6], ER-ACE [7], OCM [21], GSA [22], and OnPro [48].

Implementation details. We use full ResNet-18 [23] (not pre-trained) as the backbone for every method. For each baseline method, we perform a hyperparameter search on CIFAR-100, M=2k, and apply the hyperparameter to all of the settings. For fair comparison, we use the same optimizer and hyperparameter setting when adapting CCL-DC to the baselines. For hyperparameters unique to CCL-DC, we conduct another hyperparameter search as stated in the supplementary material. We set the streaming batch size to 10 and the memory batch size to 64. We do not use the multiple update trick as described in [3]. More detailed information about data augmentation, hyperparameter search, and hardware environments is given in the supplementary material.

5.2. Results and Analysis

Final average accuracy. Table 1 presents the results of average accuracy (AA) at the end of the training on four datasets. As indicated in Sec. 4, to fully take advantage of collaborative learning, we show the results with the ensemble of two models, with the independent model perfor-

Dataset	CIFAR10		CIFAR100			Tiny-ImageNet			ImageNet-100
Memory Size M	500	1000	1000	2000	5000	2000	5000	10000	5000
ER [38]	56.68±1.89	62.32±4.13	24.47±0.72	31.89±1.45	39.41±1.81	10.82±0.79	19.16±1.42	24.71±2.52	33.30±1.74
ER + Ours	**66.43±2.48**	**74.10±1.71**	**33.43±1.06**	**44.45±1.04**	**53.81±1.16**	**16.56±1.63**	**29.39±1.23**	**37.73±0.85**	**43.11±1.49**
DER++ [6]	58.04±2.30	64.02±1.92	25.09±1.41	32.33±2.66	38.31±2.28	8.73±1.58	17.95±2.49	19.40±3.71	34.75±2.23
DER++ + Ours	**68.79±1.42**	**74.25±1.10**	**34.36±0.89**	**43.52±1.35**	**52.95±0.86**	**10.99±1.39**	**21.68±1.94**	**28.01±2.46**	**45.70±1.32**
ER-ACE [7]	53.26±3.04	59.94±2.40	28.36±1.99	34.21±1.53	39.39±1.31	13.56±1.00	20.84±0.43	25.92±1.07	38.37±1.20
ER-ACE + Ours	**70.08±1.38**	**75.56±1.14**	**37.20±1.15**	**45.14±1.00**	**53.92±0.48**	**18.32±1.49**	**26.22±2.01**	**32.23±1.70**	**45.15±1.94**
OCM [21]	68.19±1.75	73.15±1.05	28.02±0.74	35.69±1.36	42.22±1.06	18.36±0.95	26.74±1.02	31.94±1.19	23.67±2.36
OCM + Ours	**74.14±0.85**	**77.66±1.46**	**35.00±1.15**	**43.34±1.51**	**51.43±1.37**	**23.36±1.18**	**33.17±0.97**	**39.25±0.88**	**43.19±0.98**
GSA [22]	60.34±1.97	66.54±2.28	27.72±1.57	35.08±1.37	41.41±1.65	12.44±1.17	19.59±1.30	25.34±1.43	41.03±0.99
GSA + Ours	**68.91±1.68**	**75.78±1.16**	**35.56±1.39**	**44.74±1.32**	**55.39±1.09**	**16.70±1.66**	**28.11±1.70**	**37.13±1.75**	**44.28±1.16**
OnPro [48]	70.47±2.12	74.70±1.51	27.22±0.77	33.33±0.93	41.59±1.38	14.32±1.40	21.13±2.12	26.38±2.18	38.75±1.03
OnPro + Ours	**74.49±2.14**	**78.64±1.42**	**34.76±1.12**	**41.89±0.82**	**50.01±0.85**	**21.81±1.02**	**32.00±0.72**	**38.18±1.02**	**47.93±1.26**

Table 1. Average Accuracy (%, higher is better) on four benchmark datasets with difference memory buffer size M, with and without our proposed CCL scheme. The result of our method is given by the ensemble of two peer models. All values are averages of 10 runs.

Dataset	CIFAR10		CIFAR100			Tiny-ImageNet			ImageNet-100
Memory Size M	500	1000	1000	2000	5000	2000	5000	10000	5000
ER	83.13±1.60	78.15±3.60	53.77±1.51	51.53±1.66	50.79±0.71	68.15±1.47	64.99±1.22	64.44±1.45	53.95±1.51
ER + Ours	**90.60±1.50**	**89.99±1.50**	**72.38±0.66**	**70.86±0.72**	**68.84±1.05**	**85.24±0.53**	**81.75±0.83**	**79.54±0.74**	**68.73±1.21**
DER++	77.14±2.96	78.00±2.16	56.13±3.75	55.33±3.26	56.32±3.44	70.01±1.83	66.87±1.30	70.28±2.42	60.65±2.97
DER++ + Ours	**88.85±1.88**	**89.00±1.67**	**72.85±1.37**	**71.54±1.99**	**69.52±2.37**	**82.83±1.27**	**78.80±1.62**	**77.79±0.86**	**70.16±1.03**
ER-ACE	57.66±4.16	61.59±3.35	38.53±1.61	39.95±2.00	41.56±1.44	5.60±1.45	4.83±0.78	4.92±0.95	49.82±1.05
ER-ACE + Ours	**88.37±1.39**	**88.40±1.15**	**69.47±0.88**	**68.39±1.32**	**66.63±0.90**	**21.91±5.16**	**21.88±4.39**	**18.88±3.12**	**68.52±0.82**
OCM	78.71±3.66	81.33±2.06	40.87±1.60	42.00±1.48	42.43±1.80	18.56±2.87	15.86±2.01	15.03±2.02	20.77±1.88
OCM + Ours	**82.39±2.23**	**84.53±1.63**	**48.89±2.04**	**49.83±2.01**	**49.94±2.16**	**31.69±1.81**	**29.54±2.35**	**28.10±2.28**	**48.20±1.38**
GSA	79.87±3.26	77.09±4.55	58.16±1.58	55.13±1.81	50.34±1.73	20.46±1.59	15.86±1.26	14.50±0.63	62.59±1.17
GSA + Ours	**91.69±1.11**	**90.98±1.33**	**73.73±1.03**	**72.68±0.98**	**70.36±1.07**	**80.36±1.22**	**74.77±1.66**	**70.71±1.19**	**73.71±1.12**
OnPro	84.23±2.00	85.60±1.56	41.34±1.63	42.59±1.65	42.92±1.00	20.84±1.47	16.73±1.27	15.82±1.04	39.60±0.86
OnPro + Ours	**90.39±1.59**	**90.18±1.58**	**46.30±1.10**	**47.13±1.01**	**47.27±1.81**	**25.87±1.91**	**21.40±1.52**	**19.75±1.22**	**52.55±2.18**

Table 2. Learning Accuracy (%, higher is better) on four benchmark datasets with difference memory buffer size M, with and without our proposed CCL scheme. The result of our method is given by the ensemble of two peer models. All values are averages of 10 runs.

mance available in Sec. 6. Generally, the ensemble method provides about 1% additional accuracy compared to independent inference. For all datasets, memory size M, and baseline methods, applying CCL-DC constantly improves the performance by a large margin. Notably, even for state-of-the-art methods like GSA and OnPro, we can still gain significant performance when incorporating CCL-DC.

More interestingly, for almost all settings with different memory buffer sizes M, the performance gain tends to be a constant on a relative basis. For example, CCL-DC can boost the performance of ER on Tiny-ImageNet from 10.82 to 16.56 when M=2k, which is a 53.0% performance gain on a relative basis. The performance gain is 53.4% and 52.7% when M=5k and M=10k respectively. This indicates that we can achieve a decent performance gain regardless of the memory buffer size, and it shows the scalability of our method to different resource conditions.

Plasticity and stability metric. As mentioned in Sec. 3, we evaluate the plasticity and stability of different continual learners with Learning Accuracy and Relative Forgetting, respectively. Table 2 shows the plasticity metric on four datasets. For all settings, CCL-DC constantly improves the model plasticity by a large margin. For model stability, as indicated by RF in Table 3, models trained with CCL-DC are comparable with the baselines under most cases. ER-ACE is an exception as its plasticity is unexpectedly low, especially on TinyImagenet. Also, the stability of ER-ACE is compromised when incorporating CCL-DC. We will explain the reason for this unexpected phenomenon in the supplementary material.

5.3. Ablation Studies

Effect of multiview learning. As mentioned in Sec. 4, CCL-DC benefits from multiview learning with data aug-

Dataset	CIFAR10		CIFAR100			Tiny-ImageNet			ImageNet-100
Memory Size M	500	1000	1000	2000	5000	2000	5000	10000	5000
ER	31.63±3.81	20.63±8.32	55.71±2.24	39.11±3.87	23.05±3.69	85.00±1.30	71.62±2.18	62.43±3.83	39.26±3.21
ER + Ours	**26.74±3.99**	**17.58±2.71**	**54.34±2.22**	**37.67±2.16**	**21.98±2.59**	**81.13±1.93**	**64.79±1.32**	**53.18±0.99**	**37.78±2.18**
DER++	23.60±3.64	17.71±2.18	55.65±4.36	41.27±4.93	31.72±3.95	87.79±2.35	73.28±3.88	72.51±5.53	42.97±5.89
DER++ + Ours	**22.62±3.03**	**16.43±3.36**	**53.45±1.40**	**39.39±2.71**	**23.71±3.39**	**87.16±1.60**	**73.15±2.15**	**64.48±3.08**	**35.32±2.80**
ER-ACE	**12.25±3.84**	**9.92±2.83**	**25.88±4.10**	**17.68±1.90**	**10.62±2.08**	57.41±2.38	44.48±1.96	37.83±3.12	**23.92±2.05**
ER-ACE + Ours	20.62±2.26	14.32±2.58	46.78±1.91	34.19±2.40	19.01±0.94	**56.56±4.16**	**42.20±3.94**	**31.13±3.52**	34.43±3.60
OCM	13.05±4.37	11.00±3.11	31.16±2.69	17.90±3.73	6.85±2.25	56.66±2.53	40.59±1.55	30.80±2.29	**4.55±1.60**
OCM + Ours	**10.75±2.52**	**8.45±2.63**	**29.65±4.00**	**17.02±3.01**	**6.16±1.35**	**51.58±2.81**	**35.58±2.54**	**27.24±1.60**	15.33±2.28
GSA	25.02±2.83	**16.56±4.02**	53.42±3.12	**37.29±2.60**	**20.50±4.33**	**66.87±3.31**	**53.42±3.84**	**43.44±3.81**	**35.44±2.42**
GSA + Ours	**24.96±3.27**	16.59±2.09	**52.29±2.06**	38.76±2.41	21.36±2.36	80.08±1.97	63.85±1.78	49.73±2.10	40.46±2.54
OnPro	**16.47±4.23**	12.93±3.02	35.03±4.45	24.26±2.31	12.04±2.11	64.69±3.36	50.47±4.20	42.81±4.63	**14.44±2.08**
OnPro + Ours	17.54±4.15	**12.90±2.77**	**27.64±3.29**	**17.78±1.39**	**8.41±2.62**	**56.03±2.96**	**38.70±1.88**	**29.24±1.33**	15.72±3.29

Table 3. Relative Forgetting (%, lower is better) on four benchmark datasets with difference memory buffer size M, with and without our proposed CCL scheme. The result of our method is given by the ensemble of two peer models. All values are averages of 10 runs.

Method	Acc. ↑	LA ↑
ER	31.89±1.45	51.53±1.66
ER + Multivew	38.18±1.46	64.02±1.12
ER + Ours (CCL only)	41.05±1.21	68.76±0.79
ER + Ours	44.45±1.04	70.86±0.72
ER-ACE	34.21±1.53	39.95±2.00
ER-ACE + Multivew	38.61±1.48	47.45±1.88
ER-ACE + Ours (CCL only)	40.90±1.08	50.91±1.63
ER-ACE + Ours	45.14±1.00	68.39±1.32

Table 4. Ablation studies on CIFAR-100 (M=2k). We report the ensemble performance for methods incorporating CCL.

Method	Distillation scheme	Acc. ↑	LA ↑
ER	Easy to hard	40.95±0.97	60.03±0.98
ER	Same difficulty	43.64±1.09	69.49±0.78
ER	Hard to easy (Ours)	44.45±1.04	70.86±0.72
ER-ACE	Easy to hard	38.46±1.51	39.00±1.03
ER-ACE	Same difficulty	43.81±1.28	55.37±1.54
ER-ACE	Hard to easy (Ours)	45.14±1.00	68.39±1.32

Table 5. Comparison of different distillation schemes in DC on CIFAR-100 (M=2k).

mentation in DC. For fair comparison, we explore how multiview learning will impact the performance of the baselines. We apply the classification loss part of CCL-DC to the baselines. Table 4 demonstrates that multiview learning can improve both AA and LA of baselines. However, those performance gains are still inferior to CCL-DC.

Effect of CCL. We evaluate how CCL alone can improve the baselines. In the experiments, we remove multiview learning and DC, and we train the continual learner pair with the loss illustrated in Eq. 9. Table 4 shows the performance gain for ER and ER-ACE. We can see that CCL alone can provide significant gains in both final accuracy and plasticity. Also, when combining CCL with DC, the performance can be further improved.

Distillation scheme of DC. We also evaluate the effectiveness of DC's strategy of distilling from harder samples to easier samples in collaborative learning manner. As shown in Table 5, we compared it with other distillation strategies. The result shows that the distillation scheme of DC constantly outperforms other schemes. Extra experiments explaining the working mechanism of DC can be found in the supplementary material.

6. Discussions

In this section, we analyze some properties of CCL-DC.

Improving plasticity. One of the important advantages of CCL-DC is that it can improve the plasticity of continual learners. This can be evident by plasticity metrics like LA. Moreover, we have observed that the plasticity of CCL-DC facilitates the model to converge faster and descend to a deeper loss. Figure 3 illustrates the classification loss (cross-entropy) curve of the model. To obtain the loss curve, we take a snapshot of the model every 10 iterations and compute the cross-entropy over all the training samples on the *current* task. We plot the curve on the logarithm scale so that it is easy to observe that CCL-DC helps the model descend deeper at the end of each task.

Improving feature discrimination. Another advantage of CCL-DC is its ability to enhance the feature discrimination of continual learners. Fig. 4 illustrates the t-SNE visualization [46] of the memory data's embedding space at the end of the training. We can see that the feature representation of the method with CCL-DC is more discriminative compared with the baseline.

Moreover, we can evaluate the feature discrimination using the clustering methods. Following [31], we remove the

CVPR
#7545

Method	NCM Acc. ↑	Logit Acc. ↑
ER	36.56$_{\pm 0.60}$	31.89$_{\pm 1.45}$
ER + Ours	44.76$_{\pm 0.55}$	44.45$_{\pm 1.04}$
ER-ACE	34.91$_{\pm 1.02}$	34.21$_{\pm 1.53}$
ER-ACE + Ours	45.62$_{\pm 1.04}$	45.14$_{\pm 1.00}$
OnPro	34.32$_{\pm 0.95}$	33.33$_{\pm 0.93}$
OnPro + Ours	42.82$_{\pm 0.67}$	41.89$_{\pm 0.82}$

Table 6. Final average accuracy on CIFAR-100 (M=2k), with and without NCM classifier.

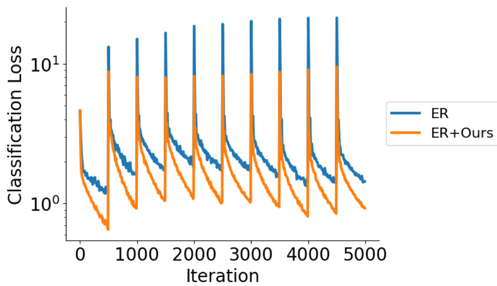
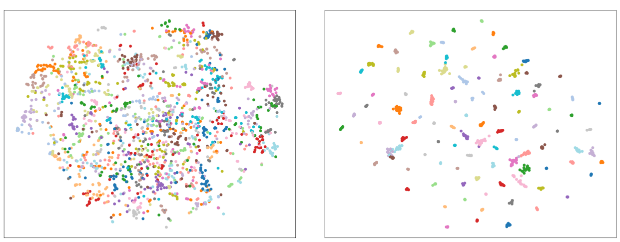


Figure 3. Classification loss curve of ER on CIFAR-100 (M=2k). The curve is calculated on all training samples of the *current* task. Since there are 10 tasks in total, the curve has 10 peaks.



(a) ER (b) ER + Ours

Figure 4. T-SNE visualization of memory data at the end of training on CIFAR-100 (M=2k).

final FC classifier and use Nearest-Class-Mean (NCM) [31] classifier with intermediate representations. Table 6 demonstrates that CCL-DC can greatly enhance the NCM accuracy, which evidences the capability of CCL-DC in improving feature discrimination.

Alleviating shortcut learning. Shortcut learning [18] is another commonly observed issue that hinders the generalization capability of continual learners [48]. In Fig. 5, we use GradCAM++ [8] on the training set of ImageNet-100 (M=5k) at the end of the training of ER and GSA. Although both ER and GSA make correct predictions, we observed that they focus on irrelevant objects, which indicates a tendency toward shortcut learning. Also, we can see that by integrating CCL-DC, the shortcut learning can be greatly alleviated.

Independent network performance. Although the ensemble method gives extra performance at inference time, by averaging the logit output of two networks in CCL-DC,

Method	Ind. Acc. ↑	Ens. Acc. ↑
ER + Ours	43.58$_{\pm 1.05}$	44.45$_{\pm 1.04}$
DER++ + Ours	42.79$_{\pm 1.38}$	43.52$_{\pm 1.35}$
ER-ACE + Ours	44.15$_{\pm 1.05}$	45.14$_{\pm 1.00}$
OCM + Ours	42.39$_{\pm 1.36}$	43.34$_{\pm 1.51}$
GSA + Ours	43.84$_{\pm 1.34}$	44.74$_{\pm 1.32}$
OnPro + Ours	41.18$_{\pm 0.83}$	41.89$_{\pm 0.82}$

Table 7. Comparison of the final average accuracy achieved through independent inference and the use of the ensemble method on CIFAR-100 (M=2k). Independent accuracy (Ind. Acc.) is calculated by averaging the accuracy of two networks in CCL-DC. All values are averaged over 10 runs.

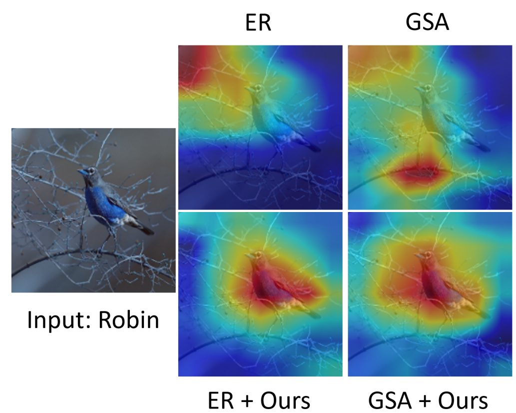


Figure 5. GradCAM++ visualization on the training set of ImageNet-100 (M=5k). Shortcut learning exists in the baseline methods despite making correct predictions.

it also doubles the computation. In some cases, computational efficiency becomes more crucial during inference. Continual learners trained with CCL-DC are also able to do inference independently, albeit with a slight performance drop compared with ensemble inference. Table 7 illustrates the accuracy achieved through independent inference. It is evident that the performance loss in independent inference, when compared to ensemble inference, is minimal (approximately 1%).

7. Conclusion

In this paper, we highlight the significance of plasticity in online CL, which has been neglected in prior research. We also establish the quantitative link between plasticity, stability, and final accuracy. The quantitative relationship sheds light on the future direction of online CL research. Based on this, we introduce collaborative learning into online CL and propose CCL-DC, a strategy that can be seamlessly integrated into existing online CL methods. Extensive experiments show the effectiveness of CCL-DC in boosting plasticity and subsequently improving the final performance.

References

[1] Rahaf Aljundi, Punarjay Chakravarty, and Tinne Tuytelaars. Expert gate: Lifelong learning with a network of experts. In *CVPR*, 2017. 2

[2] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *ECCV*, 2018. 2

[3] Rahaf Aljundi, Eugene Belilovsky, Tinne Tuytelaars, Laurent Charlin, Massimo Caccia, Min Lin, and Lucas Page-Caccia. Online continual learning with maximal interfered retrieval. In *NeurIPS*. 2019. 1, 5

[4] Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. In *NeurIPS*, 2019. 1

[5] Rohan Anil, Gabriel Pereyra, Alexandre Passos, Robert Ormandi, George E Dahl, and Geoffrey E Hinton. Large scale distributed neural network training through online distillation. *arXiv preprint arXiv:1804.03235*, 2018. 1, 2, 4

[6] Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. In *NeurIPS*, 2020. 1, 2, 5, 6

[7] 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 preprint arXiv:2104.05025*, 2021. 1, 2, 5, 6

[8] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *WACV*, 2018. 8

[9] Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *ECCV*, 2018. 2, 3

[10] 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 preprint arXiv:1902.10486*, 2019. 2

[11] Zhiyuan Chen and Bing Liu. Lifelong machine learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 12(3):1–207, 2018. 1

[12] Yubei Chen Chun-Hsiao Yeh. IN100pytorch: Pytorch implementation: Training resnets on imagenet-100. https://github.com/danielchyeh/ImageNet-100-Pytorch, 2022.

[13] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *CVPR-W*, 2020. 5

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

[15] Cyprien de Masson D'Autume, Sebastian Ruder, Lingpeng Kong, and Dani Yogatama. Episodic memory in lifelong language learning. In *NeurIPS*, 2019. 2

[16] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[17] 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 preprint arXiv:1701.08734*, 2017. 2

[18] 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. 8

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

[20] Qiushan Guo, Xinjiang Wang, Yichao Wu, Zhipeng Yu, Ding Liang, Xiaolin Hu, and Ping Luo. Online knowledge distillation via collaborative learning. In *CVPR*, 2020. 2

[21] Yiduo Guo, Bing Liu, and Dongyan Zhao. Online continual learning through mutual information maximization. In *ICML*, 2022. 1, 2, 5, 6

[22] Yiduo Guo, Bing Liu, and Dongyan Zhao. Dealing with cross-task class discrimination in online continual learning. In *CVPR*, 2023. 1, 2, 5, 6

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5

[24] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *CVPR*, 2019. 5

[25] Yen-Chang Hsu, Yen-Cheng Liu, Anita Ramasamy, and Zsolt Kira. Re-evaluating continual learning scenarios: A categorization and case for strong baselines. *arXiv preprint arXiv:1810.12488*, 2018. 1

[26] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017. 2

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

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

[29] Sang-Woo Lee, Jin-Hwa Kim, Jaehyun Jun, Jung-Woo Ha, and Byoung-Tak Zhang. Overcoming catastrophic forgetting by incremental moment matching. In *NeurIPS*, 2017. 2

[30] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *NeurIPS*, 2017. 3

[31] Zheda Mai, Ruiwen Li, Hyunwoo Kim, and Scott Sanner. Supervised contrastive replay: Revisiting the nearest class mean classifier in online class-incremental continual learning. In *CVPR*, 2021. 7, 8

[32] 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*, 45(5):5513–5533, 2022. 1

[33] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of Learning and Motivation*, pages 109–165. 1989. 1

[34] Nicolas Michel, Giovanni Chierchia, Romain Negrel, and Jean-François Bercher. Learning representations on the unit sphere: Application to online continual learning. *arXiv preprint arXiv:2306.03364*, 2023. 2

[35] 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

[36] Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, and Geoffrey Hinton. Regularizing neural networks by penalizing confident output distributions. *arXiv preprint arXiv:1701.06548*, 2017. 5

[37] 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. *arXiv preprint arXiv:1810.11910*, 2018. 3

[38] David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. In *NeurIPS*, 2019. 1, 2, 5, 6

[39] Amir Rosenfeld and John K Tsotsos. Incremental learning through deep adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(3):651–663, 2018. 2

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

[41] Joan Serra, Didac Suris, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. In *ICML*, 2018. 2

[42] Guocong Song and Wei Chai. Collaborative learning for deep neural networks. In *NeurIPS*, 2018. 2

[43] Petru Soviany, Radu Tudor Ionescu, Paolo Rota, and Nicu Sebe. Curriculum learning: A survey. *International Journal of Computer Vision*, 130(6):1526–1565, 2022. 4

[44] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016. 5

[45] Gido M Van de Ven and Andreas S Tolias. Three scenarios for continual learning. *arXiv preprint arXiv:1904.07734*, 2019. 1

[46] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9 (11), 2008. 7

[47] Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application. *arXiv preprint arXiv:2302.00487*, 2023. 1, 3

[48] Yujie Wei, Jiaxin Ye, Zhizhong Huang, Junping Zhang, and Hongming Shan. Online prototype learning for online continual learning. In *ICCV*, 2023. 1, 2, 5, 6, 8

[49] Li Yuan, Francis EH Tay, Guilin Li, Tao Wang, and Jiashi Feng. Revisiting knowledge distillation via label smoothing regularization. In *CVPR*, 2020. 5

[50] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *ICML*, 2017. 2

[51] Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. Deep mutual learning. In *CVPR*, 2018. 2

[52] Xiatian Zhu, Shaogang Gong, et al. Knowledge distillation by on-the-fly native ensemble. In *NeurIPS*, 2018. 2