CroMo-Mixup: Augmenting Cross-Model Representations for Continual Self-Supervised Learning

Erum Mushtaq[1], Duygu Nur Yaldiz[1], Yavuz Faruk Bakman[1], Jie Ding[2], Chenyang Tao[3]*, Dimitrios Dimitriadis[3]*, and Salman Avestimehr[1]

[1] University of Southern California
{emushtaq, yaldiz, ybakman, avestime}@usc.edu
[2] University of Minnesota, dingj@umn.edu
[3] Amazon AI, {chenyt, dbdim}@amazon.com

Abstract. Continual self-supervised learning (CSSL) learns a series of tasks sequentially on the unlabeled data. Two main challenges of continual learning are catastrophic forgetting and task confusion. While CSSL problem has been studied to address the catastrophic forgetting challenge, little work has been done to address the task confusion aspect. In this work, we show through extensive experiments that self-supervised learning (SSL) can make CSSL more susceptible to the task confusion problem, particularly in less diverse settings of class incremental learning because different classes belonging to different tasks are not trained concurrently. Motivated by this challenge, we present a novel cross-model feature Mixup (CroMo-Mixup) framework that addresses this issue through two key components: 1) Cross-Task data Mixup, which mixes samples across tasks to enhance negative sample diversity; and 2) Cross-Model feature Mixup, which learns similarities between embeddings obtained from current and old models of the mixed sample and the original images, facilitating cross-task class contrast learning and old knowledge retrieval. We evaluate the effectiveness of CroMo-Mixup to improve both Task-ID prediction and average linear accuracy across all tasks on three datasets, CIFAR10, CIFAR100, and tinyImageNet under different class-incremental learning settings. We validate the compatibility of CroMo-Mixup on four state-of-the-art SSL objectives. Code is available at https://github.com/ErumMushtaq/CroMo-Mixup.

Keywords: Cross-Model feature Mixup · Self-supervised Continual Learning · Cross-Task data Mixup

1 Introduction

Self-supervised learning (SSL) has advanced significantly in recent years, demonstrating performance on par with supervised learning on diverse computer vision

* This work does not relate to their position at Amazon.

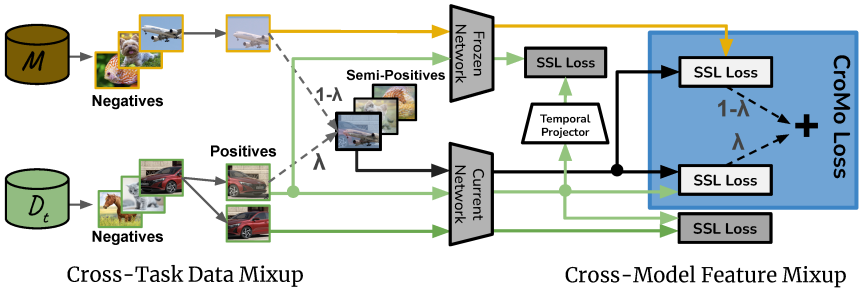


Fig. 1: Illustration of our proposed CroMo-Mixup framework. At the input, cross-task mixed samples are generated by a convex interpolation of the current and old task samples from the memory buffer. At the output, the model learns similarities between the embeddings of the cross-task mixed sample and the original samples that were mixed to create it. The embeddings of memory buffer samples come from the frozen network saved from the old task (t-1), whereas mixed samples and current task sample embeddings are attained from the network of the current task (t). In addition, model learns current task via task-specific SSL loss and distills old knowledge on the current task samples via a temporal projector-based distillation loss.

tasks, including image classification [36], segmentation [49], and object detection [14]. However, many existing SSL works assume the availability of large, unbiased datasets for model training, which may not always represent a realistic scenario. Data often becomes available progressively in many real-world applications such as self-driving cars [50] and conversational agents [28, 30]. Given the sequential nature of the data generation process of these real-world applications, it can be impractical to obtain human annotations on-the-fly. Therefore, the exploration of SSL for continual learning holds significant importance.

Continual self-supervised learning (CSSL) refers to the machine learning setting where the model learns tasks sequentially but without data labels. Under continuous shifts in data distributions, deep learning models suffer from catastrophic forgetting; loss of prior tasks knowledge while learning the new tasks. In general, CSSL has two well-explored setups, task-incremental learning (TIL) and class-incremental learning (CIL). In both TIL and CIL, each task has a distinct set of classes. However, for TIL, task id is available at the inference time, whereas, in CIL, evaluation is performed without knowing task-ids. Therefore, CIL is known to be a more challenging setup of continual learning [22].

CSSL has gained research community attention recently, and various knowledge distillation-based methods [6, 13] and exemplar-based algorithms [19, 34] have been proposed to mitigate catastrophic forgetting. However, a recent study [22] on continual supervised learning has shown that catastrophic forgetting makes supervised CIL prone to another challenge, named task confusion. In task confusion, as the model learns the new tasks, it may forget the prior tasks knowledge and therefore may fail to establish discriminative decision boundaries between the classes of different tasks. Though they study various supervised CIL

works to analyze task-confusion aspect of continual learning (CL), CSSL has not been studied from the perspective of task-confusion before. Motivated by this research gap, we study CSSL from the task-confusion aspect in CIL setups.

First, we hypothesize that the task confusion problem in contrastive SSL methods arises primarily from the inability to train classes belonging to different tasks concurrently. In supervised continual learning, cluster overlap can be a result of forgetting whereas SSL may suffer from this problem even when forgetting effects are eliminated. To demonstrate this, we conduct experiments, presented in Section 3, to study only the task-confusion problem in SSL. Our study shows that contrastive SSL baselines observe a significant drop (4% or more) in both linear accuracy and task-id prediction when classes are separated across tasks, even if tasks can be revisited frequently. This accuracy drop is compared to the offline setting when classes are randomly sampled from the whole training dataset. The performance drop especially in task-id prediction highlights the model confusion in predicting the task-ids correctly. However, within-task performance remains equally good across both experiment settings. Interestingly, we did not observe such accuracy drop in linear accuracy and task-id prediction for the similar experiment settings of supervised learning, which hints that without forgetting, supervised learning may not experience task confusion.

Given the above-mentioned observations, a straightforward solution can be storing some samples and using them as a replay. However, the challenge is that those limited old task samples might not be enough to create sufficient contrast between classes of old and new tasks as we observed for the ER baseline [37] in Table 2, and some other baselines in Table 1. Therefore, to integrate memory buffer samples effectively for contrastive learning, we propose Cross-Model feature Mixup (CroMo-Mixup) framework that exploits a small memory buffer and last task's model. As shown in Fig. 1, our proposed formulation consists of two components, Cross-Task data Mixup and Cross-Model feature Mixup. Cross-Task data Mixup generates cross-task class mixed samples via mixup data augmentation [51] to enhance negative sample diversity. Cross-Model feature Mixup formulation learns the similarities between the embeddings of the cross-task mixed samples and the original samples that were mixed to create it. Instead of following the traditional SSL approach of contrasting positives and negatives only, it learns similarities between the original samples and their stochastic mixtures with another negative which can be more challenging. Note that in the proposed formulation, we obtain the embeddings from cross-models, that is, the old task data embedding from the old model and new task data embedding from the new model. This formulation essentially enhances the remembrance of old knowledge via cross-model knowledge retrieval, and learns better class boundaries by learning on diverse and challenging cross-task mixed samples.

Our key contributions in this work are as follows,

- First, we show the inherent challenges of self-supervised learning that could impact CSSL. With extensive experiments on four SSL baselines, we show the susceptibility of CSSL to the task confusion problem even under relatively simpler setups where forgetting effects are mitigated.

- ◇ We propose a novel cross-model feature mixup framework for CSSL. It creates stochastic mixtures of cross-task data samples that enhance the negative sample diversity. To learn better cross-task class contrast on these samples, we exploit cross-model feature mixup that learns similarities between the cross-model embeddings of the cross-task mixed and original samples.
- ◇ We implement CroMo-Mixup framework with four SSL baselines, CorInfomax [36], Barlow-Twins [48], SimCLCR [8], and BYOL [14] to show its compatibility with SSL. For all these baselines, CroMo-Mixup consistently outperforms the state-of-the-art CSSL work, CaSSLe, on three datasets, CIFAR10, CIFAR100, and tinyImageNet. For the best-performing SSL baseline, CroMo-Mixup outperforms CaSSLe by 4.2%, 5.8%, 1.5%, 5.2% accuracy improvement on CIFAR100-Split5, CIFAR100-Split10, CIFAR10-Split2 and tinyImageNet-Split10, respectively.

2 Preliminaries

2.1 Self-Supervised Learning

Self-supervised learning aims to learn data representations without the need for explicit external labels. Given a dataset $\mathcal{D} = \{x_1, x_2, ..., x_N\}$, where x_i represents the i-th data sample in the dataset, and N is the total number of samples, the objective of SSL is to learn an embedding function $f : \mathcal{X} \rightarrow \mathcal{H}$. The embedding function f maps an input space $\mathcal{X}$ to a feature space $\mathcal{H}$ such that samples coming from the same class in the input space $\mathcal{X}$ are linearly separable in the embedding space $\mathcal{H}$ from other classes. The most common state-of-the-art SSL methods adopt contrastive learning for this purpose and have shown comparable results to supervised learning [8]. These methods use an additional projector network g which is mostly an MLP [8,36,48]. Initially, two different views of input x_i are obtained by applying multiple augmentations $\mathcal{A}$ such as cropping, rotation, color distortion, and noise injection [8,36,48]. The augmented views are regarded as positive pairs for each other. The first view $x_i^1 = \mathcal{A}_1(x_i)$ is fed to the encoder and the projector to yield its representations $z_i^1 = g(f(x_i^1))$; and the second view $x_i^2 = \mathcal{A}_2(x_i)$ is forwarded to the copy of f and g or the target network (e.g same architecture with f and g but parametrized with exponential moving average of parameters of f and g) to yield z_i^2. Finally, an SSL loss $\mathcal{L}_{SSL}$ is applied between the final features of two views:

$$\arg\min_{\theta} \mathbb{E}[\mathcal{L}_{SSL}(\mathbf{z}^1, \mathbf{z}^2)], \tag{1}$$

where $\mathbf{z}^k = [z_1^k, z_2^k, ..., z_N^k,]$ and θ represents the parameters of f and g functions together. Some popular SSL loss functions are InfoNCE [8,17], MSE [14], Cross-Correlation [48], Infomax [36]. The key objective of these algorithms is to learn distortion-invariant visual representations, i.e., output similar embeddings for the positive pairs, and dissimilar embeddings for the negative samples. We provide further details of our baseline SSL methods in the Appendix D.3.

2.2 Problem Definition and Evaluation Setup

Continual Self-Supervised Learning We consider Continual Self-Supervised Learning (CSSL) problem, where the main aim is to make neural network continually learn from new data over time without forgetting previously acquired knowledge. Formally, let us consider a sequence of tasks $\mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_T$ that an SSL model encounters over time, where each task $\mathcal{T}_t$ is associated with a distinct data $\mathcal{D}_t = \{(x_{t,i}, y_{t,i})\}_{i=1}^{N_t}$ having N_t samples and only the corresponding data $\mathcal{D}_t$ is available during task $\mathcal{T}_t$. The goal of CSSL is to optimize the model's performance across all tasks:

$$\arg\min_{\theta} \sum_{t=1}^{T} \mathbb{E}[\mathcal{L}_{SSL}(\mathbf{z}_t^1, \mathbf{z}_t^2)], \tag{2}$$

where $\mathbf{z}_t^k = [z_{t,1}^k, z_{t,2}^k, ..., z_{t,N_t}^k]$. According to the data distribution across tasks, CSSL can be broadly classified under two setups, Task Incremental Learning (TIL) and and Class Incremental Learning (CIL). In both TIL and CIL, each task has a distinct set of classes. Formally, let Y_t be the set of classes in task t, then it is satisfied that $(Y_t \cap Y_{t'}) = \emptyset$ for all $t \neq t'$. In both cases, new classes occur over time while the data of the previous classes becomes unavailable. However, for TIL, task id is available at the inference time, whereas, in CIL, evaluation is performed without knowing task-ids. We focus on CIL setup as it is often regarded a more challenging setup [22] due to the unavailability of task-id at the inference time, where the model is expected to differentiate between classes belonging to different tasks. We refer to Class Incremental Self-supervised Learning as CSSL in the rest of this paper.

Evaluation of Class Incremental Self-Supervised Learning Following the setup used in previous CSSL works [13], the performance of an SSL model is measured with linear classification at the end of all tasks while the parameters of encoder network f is frozen. The linear classifier is trained using the set of encoded vectors $\{h_i = f(\mathcal{A}_{lin}(x_i))\}_{i=1}^{N}$ as inputs, where $\mathcal{A}_{lin}$ is the test data augmentations used in this process (Typically, the training data augmentations $\mathcal{A}_1$ and $\mathcal{A}_2$ are chosen to be harsher than the test data augmentations $\mathcal{A}_{lin}$). After the linear classifier training, the accuracy of the classification on the test dataset is considered as SSL performance. To analyze the behaviour of CSSL algorithms better, besides reporting the linear accuracy, we follow [22] to define and evaluate two sub-problems of CIL in a probabilistic framework. We provide the definition of the two sub-problems, Within-Task Prediction (WP) and Task-ID Prediction (TP), as well as linear accuracy below.

Let $Y_{t,j}$ be the j^{th} class of t^{th} task, Y_t be the set of all classes at t^{th} task and $X_{t,j}$ be the set of all images belong to $Y_{t,j}$. Linear layer ϕ try to map $f(x \in X_{t,j})$ to the $Y_{t,j}$. Following this notation, the metrics are defined below:

- **Linear Accuracy (LA)** is the probability of an image that is correctly classified into its class, i.e, both the task-ID and class within-the-task are correctly classified. Mathematically, LA $= P(\phi(x \in X_{t,j}) = Y_{t,j})$.

– **Task-ID Prediction (TP)** is predicting the task ID. The probability of the linear layer correctly maps an image into one of the classes that belong to the same task of that image. Mathematically, TP $= P(\phi(x \in X_{t,j}) \in Y_t)$
– **Within-Task Prediction (WP)** is predicting the class of an image given the task-id. It is the probability of doing correct classification given that task-id is correctly predicted. Mathematically, WP $= P(\phi(x \in X_{t,j}) = Y_{t,j} | \phi(x \in X_{t,j}) \in Y_t)$.

As it is obviously shown in [22], LA = WP $\times$ TP.

3 Challenges of Class Incremental Self-Supervised Learning

In this section, we explain the challenges of CSSL and their significance in CSSL problem formulation.

3.1 Catastrophic Forgetting

Catastrophic forgetting is the most addressed issue in continual self-supervised learning literature. It represents the significant loss of performance on previous tasks upon learning new ones. Due to forgetting of the previous tasks, the model's within-task prediction performance on the previous tasks decreases substantially,

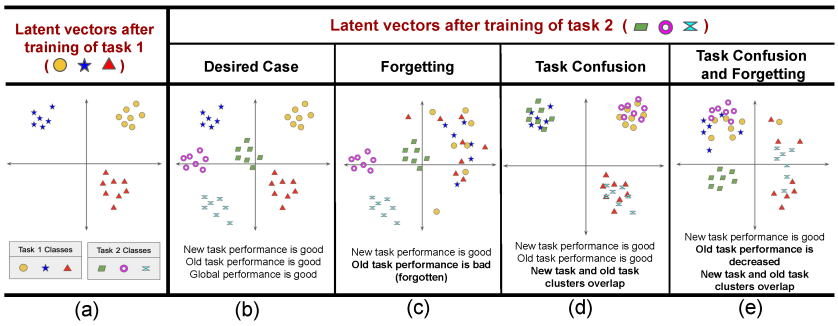


Fig. 2: Demonstration of Catastrophic Forgetting and Task Confusion challenges in a two-task based Continual Learning setup where each task contains three classes. Figure (a) illustrates the linear separability of latent vectors of task 1 classes at the end of task 1 training. Figures (b)-(e) represent the four cases after training on task 2. Case (b) shows the desired case where all classes of both tasks are linearly separable. Figure (c) illustrates the forgetting effect where task 2 classes are linearly separable but task 1 classes are not. Figure (d) shows the task confusion problem, where the model fails to draw distinctive decision boundaries between different task classes and may have overlapping clusters. Figure (e) shows the effects of task confusion and forgetting together, which is the problem in CSSL settings we want to solve.

whereas it remains at a desirable level on the current task. Forgetting also degrades the task-id prediction performance as shown in Figure 2. The earlier CSSL works exploited memory replay to address catastrophic forgetting [19]. The state-of-the-art CSSL works have proposed self-supervised learning loss adaptation for knowledge distillation [6,13] and fine-tuning [44] to mitigate this problem. However, the current literature on CSSL does not pay due attention to task confusion challenge that can hinder learning distinctive representations in the absence of labels as explained below.

3.2 Task Confusion

Task confusion represents the model failure to establish distinctive decision boundaries between different classes belonging to different tasks [20,22]. Task confusion is crucial in CIL because the task id is not present at the inference time and its absence could result in the learner's failure to accurately predict the task id, leading to mis-classification of test images. In supervised continual learning, task confusion arises as a result of forgetting which leads to the overlap of inter-task class embeddings clusters as shown in Figure 2 (subfigure (e)). However, in this work, we show that contrastive learning-based SSL methods can be susceptible to inter-task class separation problem even when forgetting effects are eliminated such as shown in Figure 2 (subfigure (d)). The model remembers old samples and WP is good, however, the model struggles to identify task-id correctly due to clusters overlap. This is because class-incremental setup naturally leads to a lesser diversity of negative samples as old task data cannot be visited with new task data to draw a contrast in the absence of labels. To illustrate it further, we describe our hypothesis and experiment results below.

Our hypothesis is as follows,

> *The task confusion problem in Contrastive SSL methods arises primarily from the inability to train the model with different classes belonging to different tasks concurrently.*

To study the hypothesis, we explore both self-supervised and supervised learning in a fairly simple but representative class-incremental setup. Our experiment setup is as follows: we follow a traditional CIL setup with a sequence of tasks $\mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_T$ that are mutually exclusive in classes. We consider the CIFAR100 dataset and split 100 classes across 10 tasks by assigning 10 classes per task. Further, we assume that tasks change after each iteration (one gradient descent step), i.e., mini-batches are sampled from different tasks at each iteration as shown in Figure 3. To study the task confusion problem explicitly and remove the forgetting effect, we assume that tasks can be revisited, i.e., task 2 follows task 1, task 3 follows task 2, and so on. The repeatability of tasks ensures that the SSL learner does not forget the previous knowledge while mutual exclusivity of classes is also maintained across mini-batches naturally leading to the representative CIL setting of lesser diversity across mini-batches. For simplicity, we refer to this experimental setup, 10x10 class-incremental learning across mini-batches, 10x10 CIL-minibatch because the data is divided into 10 tasks where each task contains 10 classes. To show how the performance of the methods changes in

CIL-minibatch setting, we also do a training on the regular setting where we sample uniformly random from the whole training data. We call this regular training setting as 100x1 CIL-minibatches because there is 1 task containing all 100 classes. Lastly, we focus on the training accuracy of the methods because we only care about the methods' capability of creating linearly separable features on the data they are trained on. We present our key results in Fig. 4. Our main observations from the results of these experiments are:

– **1**: The linear accuracy of all four representative self-supervised learning models drops significantly in 10x10 CIL mini-batch experiments as compared to the 100x1 Joint-SSL case. The accuracy problem stems from the fact that certain classes are not trained in the same mini-batch together. The lower linear accuracy is reflected majorly in lower TP whereas WP remains good overall. This hints that when trained in a lesser negative diversity setup such as CIL, self-supervised learning suffers from the task confusion problem as reflected in lower Task-ID prediction performance.

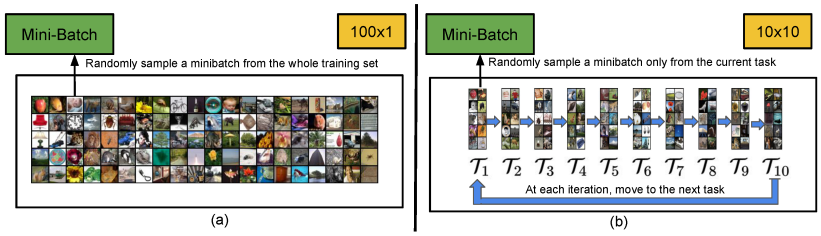
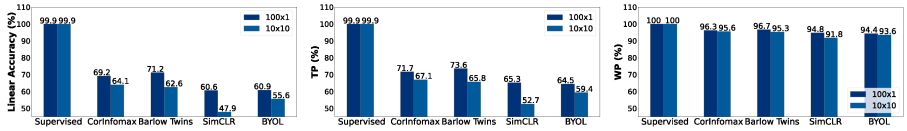


Fig. 3: Depiction of 100x1 and 10x10 CIL-minibatch task confusion experiment setup on the CIFAR100 dataset. Figure (a) represents the 100x1 case where a regular uniform sampling is performed from all the samples containing all 100 classes. Figure (b) shows the 10x10 setting where there are 10 tasks and each task contains only 10 classes. Classes are mutually exclusive across tasks. For SSL training, a mini-batch is sampled only from a single task at a time. After each iteration, mini-batch sampler moves to the next task so that tasks can be revisited throughout the training.



(a) LA: 10x10 versus 100x1 **(b)** TP: 10x10 versus 100x1 **(c)** WP: 10x10 versus 100x1

Fig. 4: Training LA, WP, and TP performance of contrastive SSL methods, CorInfomax [36], Barlow-Twins [48], SimCLR [8], and BYOL [14], and supervised learning on the CIFAR100 Dataset for 100x1 and 10x10 CIL-minibatch settings. Figure (a) demonstrates that the 10x10 setting leads to a significant accuracy drop across all SSL baselines as compared to the 100x1 setting. Figure (b) presents that the lower linear accuracy is reflected in lower task-id prediction performance, demonstrating the task-confusion problem. Figure (c) shows that the WP performance remains relatively good.

– **2**: In contrast to self-supervised learning, supervised learning exhibits no change in accuracy and maintains linear separability of classes in the embedding space for both 100x1 joint-SL and 10x10 test cases. This shows that this challenge is unique to contrastive learning-based SSL methods and may not affect supervised learning due to the presence of explicit class labels.

Overall, our experimental results in this section confirm our hypothesis, underlining the importance of addressing task confusion problem in CSSL. A data incremental setting-based ablation study that further strengthens our hypothesis can be found in Appendix Section B.

4 Proposed Method

In CSSL, the main objective is to learn visual representations that remain informative about the old task data distributions while learning the new task such that linear separability of all the classes from all the data distributions is maximized at the end of the CL Phase. Existing works have proposed supervised learning solution adaptations to CSSL to address catastrophic forgetting such as Distillation [6,13], and memory-replay [34]. However, the CSSL representation continuity has not been studied from task confusion perspective before. As we have shown in Section 3, task confusion is a major challenge for SSL. In literature, it is well-known that SSL often requires large unsupervised datasets, and larger mini-batches to ensure sufficient negative sample diversity from all the classes to learn the linear separability of different classes in the embedding space [8,14,45]. However, under the CSSL problem setup, only a small amount of data can be saved (as an exemplar), and this might not be sufficient to produce the desired negative sample diversity even when used as a replay (as shown in the results Section, e.g., 6, ER [37], DER [3], EWC [24], LUMP [34]). Given these challenges, we propose an exemplar-based approach that focuses on enhancing the negative sample diversity under the limited memory buffer constraints.

Our proposed framework consists of two components: 1) Cross-Task Data Mixup and 2) Cross-Model Feature Mixup which are described in detail below.

Cross-Task Data Mixup We generate cross-task mixed samples by exploiting the well-known mixup data augmentation [51] for self-supervised learning. Specifically, for x_{t_i} sampled from current task data distribution $\mathcal{D}_t$, we randomly sample $x_{\mathcal{M}_j}$ from the memory buffer $\mathcal{M}$ that contains samples from the previous task data distributions, and generate inter-task class mixed data sample $x_{mix_{ij}}$, a convex interpolation of x_{t_i} and $x_{\mathcal{M}_j}$ as shown below,

$$x_{mix_{ij}} = \lambda x_{t_i} + (1 - \lambda) x_{\mathcal{M}_j} \tag{3}$$

where $\lambda \in \text{Beta}(\alpha, \alpha)$, and $\alpha \in (0, \infty)$. Note that we use mixup for both views of data samples, $(x^1_{t_i}, x^2_{t_i})$ and $(x^1_{\mathcal{M}_j}, x^2_{\mathcal{M}_j})$ which subsequently results in two inter-task mixed data samples $x^1_{mix_{ij}}$ and $x^2_{mix_{ij}}$.

Cross-Model Feature Mixup (CroMo-Mixup) For learning on the cross-task mixed data samples, our proposed formulation is as follows,

$$\mathcal{L}_{CroMo}(z_{mix_{ij}}, z_{t_i}, \bar{z}_{\mathcal{M}_j}) = \lambda \cdot \mathcal{L}_{SSL}(z_{mix_{ij}}, z_{t_i}) + (1-\lambda) \cdot \mathcal{L}_{SSL}(z_{mix_{ij}}, \bar{z}_{\mathcal{M}_j}) \quad (4)$$

where $\mathcal{L}_{SSL}$ is the SSL loss for the considered SSL baseline. $z_{mix_{ij}}$, z_{t_i} are the feature embeddings obtained from the current model for the $x_{mix_{ij}}$ and x_{t_i} data samples, and $\bar{z}_{\mathcal{M}_j}$ feature embedding is obtained from the frozen old task model for the $x_{\mathcal{M}_j}$ data-point.

The proposed learning objective has three key features. For a given x_{t_i}, $x_{\mathcal{M}_j}$, and $x_{mix_{ij}}$ data samples, it treats the embeddings of the rest of the cross-task mixed samples z_{mix}, the current task embeddings z_t, and the old task embeddings $\bar{z}_{\mathcal{M}}$ in the mini-batch as negatives for each other which enriches the negative sample diversity of overall learning, as compared to the traditional SSL learning where we only have the old and new task's data embeddings as negatives for each other. Second, it encourages the learner to learn the similarities between the cross-task mixed sample $z_{mix_{ij}}$ and the corresponding current task sample z_{t_i} as well as the old task sample $\bar{z}_{\mathcal{M}_j}$. This soft distance learning helps in improving the task-id prediction performance because learner learns to identify the similarity of an image from a more challenging image, augmented as well as mixed with a cross-task negative sample, than only an augmented version of itself. Further, the proposed formulation exploits the old task data embeddings from the old model $\bar{z}_{\mathcal{M}}$, which promotes retrieval and preservation of the old knowledge while learning new knowledge.

In addition to the learning objective 4, a general task-specific loss is also employed $\mathcal{L}_{SSL}$ to learn the current task on the current task data distribution $\mathcal{D}_t$. We also exploit distillation on the current task data distribution $\mathcal{D}_t$ benefiting from the old task model. Hence, the total objective becomes,

$$\begin{aligned} \mathcal{L}_{total} = \mathcal{L}_{SSL}(z^1_{t_i}, z^2_{t_i}) + \zeta(\mathcal{L}_{SSL}(\bar{z}^1_{t_i}, h(z^1_{t_i})) + \mathcal{L}_{SSL}(\bar{z}^2_{t_i}, h(z^2_{t_i}))) + \\ \mathcal{L}_{CroMo}(z^1_{mix_{ij}}, z^1_{t_i}, \bar{z}^1_{\mathcal{M}_j}) + \mathcal{L}_{CroMo}(z^2_{mix_{ij}}, z^2_{t_i}, \bar{z}^2_{\mathcal{M}_j}) \end{aligned} \quad (5)$$

where ζ is a hyper-parameter for the distillation objective. Further, $\mathcal{L}_{SSL}(z^1_{t_i}, z^2_{t_i})$ denotes a task specifc loss. $h(.)$ represents an MLP Predictor that is employed on $z^2_{t_i}$ and $z^1_{t_i}$ to perform distillation as proposed in [13]. The embeddings $\bar{z}^1_{t_i}$ and $\bar{z}^2_{t_i}$ are obtained from the old model for the inputs $x^1_{t_i}$ and $x^2_{t_i}$.

It is worth mentioning that mixup [21,23,27] has already been explored in contrastive SSL works as a data augmentation scheme, however, here we exploit it to formulate a CSSL problem and perform learning in a cross-task and cross-model continual learning setting. Further, one baseline, LUMP [34], has used the idea of mixup for CSSL, however, our proposed objective formulation is different from theirs. Primarily, the formulation proposed in LUMP exploits cross-task mixed samples such that it learns general-purpose features across tasks to address forgetting. Whereas our objective function learns to identify task-specific features at a granular level to address both task confusion and forgetting problems in CSSL. LUMP minimizes the SSL loss over the mixed

samples and their augmented versions, whereas CroMo-Mixup learns to find the feature similarity between the original and cross-task mixed samples in the same proportion in which they were mixed. This implicit feature learning ensures that the model can identify different task samples at a granular level. Further, CroMo-Mixup outperforms this baseline in CSSL settings as shown in Table 1.

5 Related Works

Continual Self-Supervised Learning. The research community has recently shown keen interest in CSSL problem [6, 13, 19, 34] due to its applicability in real-world scenarios. In this line of research, [34] is among the first works that demonstrated the representation continuity of SSL in task-incremental learning settings. It proposed a cross-task data mixup approach and showed that its method outperforms various supervised learning baselines in TIL settings. Another work [10], has explored CSSL for TIL settings where task confusion is not a concern. Other works [19,31] have investigated the significance of simple memory replay to address catastrophic forgetting in CSSL. However, CaSSLe [13] made significant progress. They proposed self-supervised learning loss function adaptation via a temporal projector to perform distillation. Sy-Con [6], another recent work, proposed a loss formulation that exploits current and old model embeddings of negative samples to enhance distillation regularization performance. Nonetheless, CaSSLe remains state-of-the-art on most self-supervised learning baselines. Due to space constraints, we present the literature review of SSL and Continual Learning topics in the Appendix A.

6 Experiments

6.1 Experiment Settings

Datasets We perform experiments on three datasets: CIFAR10 [25], a 10-class dataset with 60,000 32x32 color images; CIFAR100 [25], a 100-class dataset with 60,000 32x32 color images; and TinyImageNet [26], a 200-class dataset with 100000 64x64 color images. For CIFAR10, we explore a 2 task setting where 5 classes are present per task. Following [13], we experiment with a 5-task class-incremental setting for CIFAR100. Further, we also include a more challenging case with 10 tasks for CIFAR100 dataset. For tinyImageNet, we exploit a 10-task setting where 20 classes are present per task. We provide further details for each dataset setup in Appendix Section D.1.

Implementation Details We use ResNet-18 [18] as an encoder network for CIFAR10 and CIFAR100 experiments, while we employ ResNet-50 [18] for TinyImageNet. We include BYOL [14], SimCLR [8], CorInfoMax [36], and Barlow-Twins [48] as representative SSL baselines in our work. We follow these works to set up the hyper-parameters such as optimizer, learning rate, and schedulers. For

Continual Learning experiments, we use 500 epochs/task for CIFAR10-Split2, 750 epochs/task for CIFAR100-Split5, 600 epochs for the first task, and 350 epochs for the rest of the task for CIFAR100-Split10. Likewise, for tinyImageNet, we use 500 epochs for the first task and 350 epochs for the rest of the tasks. Further details of hyper-parameters tuning can be found in Appendix D.1.

Evaluation Metrics Following the CSSL baseline [13], we evaluate the model at the end of CL training. To evaluate the model, we freeze the encoder and train a linear classifier layer on the training dataset of each specific dataset. We report the average linear accuracy on the test set of each specific dataset, which is calculated as $= \frac{\text{Total \# of Correct Classification of both Class and Task-ID Prediction Tasks}}{\text{Total \# of Test samples}}$. To analyze the model performance against task confusion, we also report TP, which is calculated as $(\frac{\text{Total \# of Correct Classification of Task-ID Prediction Task}}{\text{Total \# of test samples}})$. For WP, we report, $(\frac{\text{Total \# of Correct Classification of both Class and Task-ID Prediction Tasks}}{\text{Total \# of Correct Classification of Task-ID Prediction Task}})$.

6.2 Results

Average Linear Accuracy First, we evaluate CroMo-Mixup with Barlow-Twins, SimCLR, and BYOL on the CIFAR100-Split5. We compare the average linear accuracy performance of CroMo-Mixup with CSSL baselines CaSSLe [13], Sy-Con [6], and LUMP [34], and some replay-based methods from supervised continual learning that can be adapted to CSSL such as EWC [24], ER [37], DER [3] on CIFAR100-Split5 dataset in Table 1. We also include CaSSLe+ baseline that exploits both knowledge distillation and memory buffer to make a fair comparison with the state-of-the-art baseline, CaSSLe. Overall, CroMo-Mixup outperforms all these baselines by achieving higher average linear accuracy across all three SSL baselines. Further, though we observe 1-2% accuracy gain in CaSSLe+ as compared to CaSSLe, our proposed method outperforms this SOTA baseline, CaSSLe+, by 4.2%, 3.2%, and 3.2% accuracy improvement on Barlow-Twins, SimCLR, and BYOL, respectively. In Table 1, we reproduce all baseline results except † marked which we take from CaSSLe paper [13].

Next, we draw a more detailed comparison between the proposed formulation and state-of-the-art baseline, CaSSLe and its variant CaSSLe+. We perform

Table 1: Experimental results of CSSL baselines on CIFAR100-Split5

	Barlow-Twins	**SimCLR**	**BYOL**
Method	**Avg. Linear Acc(%)**	**Avg. Linear Acc(%)**	**Avg. Linear Acc(%)**
Offline	70.0	65.1	66.7
Fine-Tune	54.4	42.7	55.2
EWC†	56.7	53.6	56.4
ER	57.2	47.7	56.1
DER†	55.3	50.7	54.8
LUMP†	57.8	52.3	56.4
Sy-Con	60.4	58.9	57.3
CaSSLe	60.6	57.6	56.9
CaSSLe+	61.3	59.5	57.4
CroMo-Mixup	65.5(+**4.2**)	62.7(+**3.2**)	60.6(+**3.2**)

Table 2: Comparison of CroMo-Mixup with state-of-the-art CSSL method, CaSSLe on CIFAR10-Split2, CIFAR100-Split5, CIFAR100-Split10 and tinyImageNet-Split20 using average linear accuracy (LA), within-task prediction (WP) & Task-ID prediction (TP)

		Barlow-Twins			CorInfoMax			SimCLR			BYOL		
	Method	LA(%)	WP(%)	TP(%)	LA(%)	WP(%)	TP(%)	LA(%)	WP(%)	TP(%)	LA(%)	WP(%)	TP(%)
CIFAR10-Split2	Offline	91.65	-	-	92.18	-	-	90.35	-	-	89.60	-	-
	Fine-tune	82.67	90.12	91.73	81.71	91.25	89.55	80.97	90.54	89.43	84.16	94.43	89.12
	ER	85.61	90.36	94.62	87.67	95.81	91.50	81.52	90.95	89.63	86.71	94.97	91.30
	CaSSLe	87.64	91.25	95.87	87.62	96.02	91.17	86.88	95.21	91.25	87.00	95.80	90.81
	CaSSLe+	86.81	90.30	95.91	87.58	95.51	91.70	87.52	95.54	91.61	87.81	96.01	91.49
	CroMo-Mixup*	87.56	92.11	94.69	86.00	93.54	91.96	84.18	92.70	90.81	89.27	96.28	92.72
	CroMo-Mixup	88.22	91.78	95.75	88.51	95.97	92.23	88.49	95.93	92.24	88.88	96.36	92.19
		(+**0.6**)		(**-0.1**)	(+**0.8**)		(+**0.7**)	(+**1.0**)	–	(+**0.6**)	(+**1.5**)	–	(+**0.7**)
CIFAR100-Split5	Offline	70.03	-	-	70.76	-	-	65.11	-	-	66.73	-	-
	Fine-tune	54.40	85.19	63.86	56.68	86.32	65.66	42.65	78.17	54.56	55.19	86.23	64.00
	ER	57.23	88.26	65.57	59.94	88.62	67.64	47.77	81.87	58.35	56.05	85.60	65.48
	CaSSLe	60.64	87.29	66.35	60.82	88.85	68.45	57.54	87.89	65.47	56.86	86.05	66.08
	CaSSLe+	61.25	88.50	69.03	60.26	88.74	67.92	59.48	88.26	67.39	57.35	87.28	65.71
	CroMo-Mixup*	63.94	91.40	69.88	62.32	88.46	70.39	59.15	87.66	67.48	59.60	88.11	67.64
	CroMo-Mixup	65.48	90.72	72.11	65.06	90.62	71.78	62.72	89.50	70.08	60.60	88.64	68.37
		(+**4.2**)		(+**3.1**)	(+**4.2**)		(+**3.3**)	(+**3.2**)		(+**2.7**)	(+**3.3**)	-	(+**2.7**)
CIFAR100-Split10	Offline	70.03	-	-	70.76	-	-	65.11	-	-	66.73	-	-
	Fine-tune	51.12	92.01	55.56	50.66	91.58	55.32	39.02	86.61	45.04	49.63	92.23	53.81
	ER	52.81	92.59	56.98	56.99	93.98	60.64	44.83	89.98	49.88	52.32	92.70	56.40
	CaSSLe	56.59	93.93	60.25	56.35	93.95	59.98	53.60	93.39	57.38	52.77	93.22	56.61
	CaSSLe+	56.64	93.49	60.68	57.00	93.95	60.67	55.02	93.43	58.89	53.39	92.67	57.61
	CroMo-Mixup*	60.01	94.50	63.41	60.30	94.34	63.74	55.21	92.84	59.84	56.59	93.52	60.51
	CroMo-Mixup	62.48	95.10	65.70	61.66	94.91	64.97	58.84	94.66	62.18	56.97	93.35	61.03
		(+**5.8**)		(+**5.0**)	(+**4.7**)		(+**4.3**)	(+**3.8**)		(+**3.3**)	(+**3.6**)	-	(+**3.4**)
tinyImageNet-Split10	Offline	55.60	-	-	55.20	-	-	49.74	-	-	47.58	-	-
	Fine-tune	39.90	77.00	51.82	41.14	78.12	52.66	36.72	75.09	48.90	37.15	76.10	48.82
	ER	40.14	77.07	52.08	41.44	78.22	52.98	37.96	70.08	50.56	37.78	76.17	49.60
	CaSSLe	43.40	79.08	54.88	41.66	77.69	53.62	40.66	77.80	52.26	38.18	77.07	49.54
	CaSSLe+	42.64	78.90	54.04	43.86	79.46	55.20	41.74	78.58	53.12	40.24	78.72	51.12
	CroMo-Mixup*	45.70	80.54	56.74	46.74	81.32	57.48	41.02	78.43	52.30	43.19	79.45	54.36
	CroMo-Mixup	47.32	81.78	57.86	48.22	81.90	58.88	45.82	80.36	57.02	45.44	80.68	56.32
		(+**3.7**)		(+**3.8**)	(+**4.4**)		(+**3.6**)	(+**4.1**)		(+**3.9**)	(+**5.2**)		(+**4.2**)

experiments across four CIL setups, three datasets, and four SSL baselines, represented in Table 2. We also include a memory replay based baseline ER to show that a limited memory buffer may not be enough to ensure sufficient negative sample diversity for optimal CSSL performance. For our proposed formulation, we include two setups, $\zeta = 0$ shown as CroMo-Mixup* in Table 2, that uses cross-model learning but does not exploit knowledge distillation, and $\zeta = 1$, shown as CroMo-Mixup that exploits both cross-model learning as well as knowledge distillation. On CIFAR10-Split2 dataset, we achieve the highest performance of 89.27% with CroMo-Mixup on the BYOL baseline that outperforms CaSSLe+ with 1.5% higher average linear accuracy. Next, for both CIFAR100-Split5 and CIFAR100-Split100 datasets, CroMo-Mixup achieves the highest average linear accuracy of 65.48% and 62.48%, respectively, on Barlow-Twins among all four SSL baselines. It outperforms CaSSLe+ with 4.2% and 5.8% higher average linear accuracy performance on CIFAR100-Split5 and Split10, respectively. It is worth noticing that even without distillation, CroMo-Mixup*($\zeta = 0$) case, CroMo-Mixup outperforms CaSSLe+, which exploits both knowledge distilla-

tion and memory buffer, with 2.7% and 3.4% higher linear accuracy performance. On tinyImageNet-Split10, CroMo-Mixup on BYOL achieves a higher accuracy of 45.44% outperforming CaSSLe+ with a 5.2% accuracy gain.

Task-ID Prediction To analyze performance against the task-confusion, we also compare the Task-ID prediction performance of CroMo-Mixup with CaSSLe and its variant CaSSLe+ across four CIL setups, three datasets, and four SSL baselines in Table 2. We observe that CroMo-Mixup achieves better performance in predicting task-ids as compared to CaSSLe and CaSSLe+ without compromising on the WP performance on nearly all SSL baselines and all considered dataset settings. The higher linear accuracy indicates the model performance against catastrophic forgetting, whereas better task-id prediction performance indicates the potential of CroMo-Mixup to maximize contrast between the classes of different tasks in the absence of class labels and limited buffer size. We also include further experiments such as an ablation study on different design components, out-of-distribution performance comparison, and buffer size versus accuracy analysis in Appendix C.

7 Conclusion

In this work, we study continual self-supervised learning (CSSL) from the task-confusion aspect of continual learning. First, we highlight its significance in CSSL problem which remained unexplored before in literature. Next, we propose a CroMo-Mixup formulation that exploits cross-task data mixup and cross-model feature mixup to enhance negative sample diversity and cross-task class contrast under limited memory buffer constraint of continual learning. Our proposed formulation outperforms the state-of-the-art CSSL works by achieving higher performance average linear accuracy and task-id prediction performance.

8 Limitations

Our proposed approach is based on a limited memory buffer. Therefore, it may not be applicable in scenarios where the user might want to delete all old samples due to privacy issues. In such cases, there is a need to design effective methods to address the task confusion challenge of the continual self-supervised learning problem. Further, we assume the tasks are clearly separated following the current literature [6, 13, 34]. However, in realistic scenarios, the transitions across tasks are mostly smoother. Therefore, this can be an important future work to explore.

Acknowledgements

This work is supported in part by a research gift from USC-Amazon Center on Secure and Trusted Machine Learning [4], ONR grant N00014-23-1-2191, and

[4] https://trustedai.usc.edu

ARO grant W911NF-22-1-0165. The work of Jie Ding was supported in part by the Army Research Office under the Early Career Program Award, Grant Number W911NF-23-10315.

References

1. Bakman, Y.F., Yaldiz, D.N., Ezzeldin, Y.H., Avestimehr, S.: Federated orthogonal training: Mitigating global catastrophic forgetting in continual federated learning. In: The Twelfth International Conference on Learning Representations (2024), https://openreview.net/forum?id=nAs4LdaP9Y
2. Bardes, A., Ponce, J., LeCun, Y.: VICReg: Variance-invariance-covariance regularization for self-supervised learning. In: International Conference on Learning Representations (2022), https://openreview.net/forum?id=xm6YD62D1Ub
3. Buzzega, P., Boschini, M., Porrello, A., Abati, D., Calderara, S.: Dark experience for general continual learning: a strong, simple baseline. Advances in neural information processing systems **33**, 15920–15930 (2020)
4. Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., Joulin, A.: Unsupervised learning of visual features by contrasting cluster assignments. Advances in neural information processing systems **33**, 9912–9924 (2020)
5. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 9650–9660 (2021)
6. Cha, S., Cho, K., Moon, T.: Augmenting negative representation for continual self-supervised learning. https://openreview.net/forum?id=7sASqAmGaO (2024)
7. Chaudhry, A., Ranzato, M., Rohrbach, M., Elhoseiny, M.: Efficient lifelong learning with a-GEM. In: International Conference on Learning Representations (2019), https://openreview.net/forum?id=Hkf2_sC5FX
8. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International conference on machine learning. pp. 1597–1607. PMLR (2020)
9. Chen, X., He, K.: Exploring simple siamese representation learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 15750–15758 (2021)
10. Cheng, H., Wen, H., Zhang, X., Qiu, H., Wang, L., Li, H.: Contrastive continuity on augmentation stability rehearsal for continual self-supervised learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5707–5717 (2023)
11. De Lange, M., Aljundi, R., Masana, M., Parisot, S., Jia, X., Leonardis, A., Slabaugh, G., Tuytelaars, T.: Continual learning: A comparative study on how to defy forgetting in classification tasks. arXiv preprint arXiv:1909.08383 **2**(6), 2 (2019)
12. Ermolov, A., Siarohin, A., Sangineto, E., Sebe, N.: Whitening for self-supervised representation learning. In: International Conference on Machine Learning. pp. 3015–3024. PMLR (2021)
13. Fini, E., Da Costa, V.G.T., Alameda-Pineda, X., Ricci, E., Alahari, K., Mairal, J.: Self-supervised models are continual learners. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9621–9630 (2022)
14. Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al.: Bootstrap your own

latent-a new approach to self-supervised learning. Advances in neural information processing systems **33**, 21271–21284 (2020)
15. Gui, J., Chen, T., Zhang, J., Cao, Q., Sun, Z., Luo, H., Tao, D.: A survey on self-supervised learning: Algorithms, applications, and future trends (2023)
16. Guo, Y., Liu, M., Yang, T., Rosing, T.: Improved schemes for episodic memory-based lifelong learning. In: Advances in Neural Information Processing Systems. vol. 33, pp. 1023–1035. Curran Associates, Inc. (2020), https://proceedings.neurips.cc/paper/2020/file/0b5e29aa1acf8bdc5d8935d7036fa4f5-Paper.pdf
17. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9729–9738 (2020)
18. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
19. Hu, D., Yan, S., Lu, Q., Hong, L., Hu, H., Zhang, Y., Li, Z., Wang, X., Feng, J.: How well does self-supervised pre-training perform with streaming data? arXiv preprint arXiv:2104.12081 (2021)
20. Huang, B., Chen, Z., Zhou, P., Chen, J., Wu, Z.: Resolving task confusion in dynamic expansion architectures for class incremental learning. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 908–916 (2023)
21. Kalantidis, Y., Sariyildiz, M.B., Pion, N., Weinzaepfel, P., Larlus, D.: Hard negative mixing for contrastive learning. Advances in Neural Information Processing Systems **33**, 21798–21809 (2020)
22. Kim, G., Xiao, C., Konishi, T., Ke, Z., Liu, B.: A theoretical study on solving continual learning. Advances in Neural Information Processing Systems **35**, 5065–5079 (2022)
23. Kim, S., Lee, G., Bae, S., Yun, S.Y.: Mixco: Mix-up contrastive learning for visual representation. arXiv preprint arXiv:2010.06300 (2020)
24. Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A.A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al.: Overcoming catastrophic forgetting in neural networks. Proceedings of the national academy of sciences **114**(13), 3521–3526 (2017)
25. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
26. Le, Y., Yang, X.: Tiny imagenet visual recognition challenge. CS 231N **7**(7), 3 (2015)
27. Lee, K., Zhu, Y., Sohn, K., Li, C.L., Shin, J., Lee, H.: i-mix: A domain-agnostic strategy for contrastive representation learning. arXiv preprint arXiv:2010.08887 (2020)
28. Li, Y., Zhao, L., Church, K., Elhoseiny, M.: Compositional language continual learning (2020)
29. Li, Z., Hoiem, D.: Learning without forgetting. IEEE transactions on pattern analysis and machine intelligence **40**(12), 2935–2947 (2017)
30. Lin, Y., Tan, L., Lin, H., Zheng, Z., Pi, R., Zhang, J., Diao, S., Wang, H., Zhao, H., Yao, Y., et al.: Speciality vs generality: An empirical study on catastrophic forgetting in fine-tuning foundation models. arXiv preprint arXiv:2309.06256 (2023)
31. Lin, Z., Wang, Y., Lin, H.: Continual contrastive learning for image classification. In: 2022 IEEE International Conference on Multimedia and Expo (ICME). pp. 1–6. IEEE (2022)

32. Liu, H., Liu, H.: Continual learning with recursive gradient optimization. In: International Conference on Learning Representations (2022), https://openreview.net/forum?id=7YDLgf9_zgm
33. Lopez-Paz, D., Ranzato, M.A.: Gradient episodic memory for continual learning. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017), https://proceedings.neurips.cc/paper/2017/file/f87522788a2be2d171666752f97ddebb-Paper.pdf
34. Madaan, D., Yoon, J., Li, Y., Liu, Y., Hwang, S.J.: Representational continuity for unsupervised continual learning. In: International Conference on Learning Representations (2022), https://openreview.net/forum?id=9Hrka5PA7LW
35. Mallya, A., Lazebnik, S.: Packnet: Adding multiple tasks to a single network by iterative pruning. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018. pp. 7765–7773. Computer Vision Foundation / IEEE Computer Society (2018). https://doi.org/10.1109/CVPR.2018.00810, http://openaccess.thecvf.com/content_cvpr_2018/html/Mallya_PackNet_Adding_Multiple_CVPR_2018_paper.html
36. Ozsoy, S., Hamdan, S., Arik, S., Yuret, D., Erdogan, A.: Self-supervised learning with an information maximization criterion. Advances in Neural Information Processing Systems **35**, 35240–35253 (2022)
37. Robins, A.: Catastrophic forgetting, rehearsal and pseudorehearsal. Connection Science **7**(2), 123–146 (1995)
38. Rusu, A.A., Rabinowitz, N.C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., Hadsell, R.: Progressive neural networks. CoRR **abs/1606.04671** (2016), http://arxiv.org/abs/1606.04671
39. Saha, G., Garg, I., Roy, K.: Gradient projection memory for continual learning. In: International Conference on Learning Representations (2021), https://openreview.net/forum?id=3AOj0RCNC2
40. Sarwar, S.S., Ankit, A., Roy, K.: Incremental learning in deep convolutional neural networks using partial network sharing. IEEE Access **8**, 4615–4628 (2020). https://doi.org/10.1109/ACCESS.2019.2963056
41. Serra, J., Suris, D., Miron, M., Karatzoglou, A.: Overcoming catastrophic forgetting with hard attention to the task. In: Dy, J., Krause, A. (eds.) Proceedings of the 35th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 80, pp. 4548–4557. PMLR (10–15 Jul 2018), https://proceedings.mlr.press/v80/serra18a.html
42. Shin, H., Lee, J.K., Kim, J., Kim, J.: Continual learning with deep generative replay. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017), https://proceedings.neurips.cc/paper_files/paper/2017/file/0efbe98067c6c73dba1250d2beaa81f9-Paper.pdf
43. Skean, O., Dhakal, A., Jacobs, N., Giraldo, L.G.S.: FroSSL: Frobenius norm minimization for self-supervised learning (2024), https://openreview.net/forum?id=1mOeklnLf4
44. Tang, C.I., Qendro, L., Spathis, D., Kawsar, F., Mascolo, C., Mathur, A.: Kaizen: Practical self-supervised continual learning with continual fine-tuning. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 2841–2850 (2024)

45. Wang, T., Isola, P.: Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In: International Conference on Machine Learning. pp. 9929–9939. PMLR (2020)
46. Yoon, J., Kim, S., Yang, E., Hwang, S.J.: Scalable and order-robust continual learning with additive parameter decomposition. In: International Conference on Learning Representations (2020), https://openreview.net/forum?id=r1gdj2EKPB
47. Yoon, J., Yang, E., Lee, J., Hwang, S.J.: Lifelong learning with dynamically expandable networks. In: International Conference on Learning Representations (2018), https://openreview.net/forum?id=Sk7KsfW0-
48. Zbontar, J., Jing, L., Misra, I., LeCun, Y., Deny, S.: Barlow twins: Self-supervised learning via redundancy reduction. In: International Conference on Machine Learning. pp. 12310–12320. PMLR (2021)
49. Zeng, Z., Xulei, Y., Qiyun, Y., Meng, Y., Le, Z.: Sese-net: Self-supervised deep learning for segmentation. Pattern Recognition Letters **128**, 23–29 (2019)
50. Zhang, H., Mueller, F.: Claire: Enabling continual learning for real-time autonomous driving with a dual-head architecture. In: 2022 IEEE 25th International Symposium On Real-Time Distributed Computing (ISORC). pp. 1–10. IEEE (2022)
51. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. In: International Conference on Learning Representations (2018), https://openreview.net/forum?id=r1Ddp1-Rb