Self-Supervised Training Enhances Online Continual Learning

Jhair Gallardo[1]
gg4099@rit.edu
Tyler L. Hayes[1]
tlh6792@rit.edu
Christopher Kanan[1,2,3]
kanan@rit.edu

[1] Rochester Institute of Technology
New York, USA
[2] Paige
New York, USA
[3] Cornell Tech
New York, USA

Abstract

In continual learning, a system must incrementally learn from a non-stationary data stream without catastrophic forgetting. Recently, multiple methods have been devised for incrementally learning classes on large-scale image classification tasks, such as ImageNet. State-of-the-art continual learning methods use an initial supervised pre-training phase, in which the first 10% - 50% of the classes in a dataset are used to learn representations in an offline manner before continual learning of new classes begins. We hypothesize that self-supervised pre-training could yield features that generalize better than supervised learning, especially when the number of samples used for pre-training is small. We test this hypothesis using the self-supervised MoCo-V2, Barlow Twins, and SwAV algorithms. On ImageNet, we find that these methods outperform supervised pre-training considerably for online continual learning, and the gains are larger when fewer samples are available. Our findings are consistent across three online continual learning algorithms. Our best system achieves a 14.95% relative increase in top-1 accuracy on class incremental ImageNet over the prior state of the art for online continual learning.

1 Introduction

Conventional convolutional neural networks (CNNs) are trained offline and then evaluated. When new training data is acquired, the CNN is re-trained from scratch. This can be wasteful for both storage and compute. Ideally, the CNN would be updated on only new samples, which is known as continual learning. The longstanding challenge has been that catastrophic forgetting [48] occurs in conventional CNNs when updating on only new samples, especially when the data stream is non-stationary (e.g., when classes are learned incrementally) [6, 17, 36]. Recently, continual learning methods have scaled to incremental class learning on full-resolution images in the 1000 category ImageNet dataset with only a small gap between them and offline systems [10, 27, 29, 34, 55, 68, 72]. These methods use an initial *supervised* pre-training period on the first 10–50% of the classes before continual learning begins.

Performance of these systems depends on the amount of data used for supervised pre-training [27]. We hypothesized that self-supervised pre-training would be more effective,

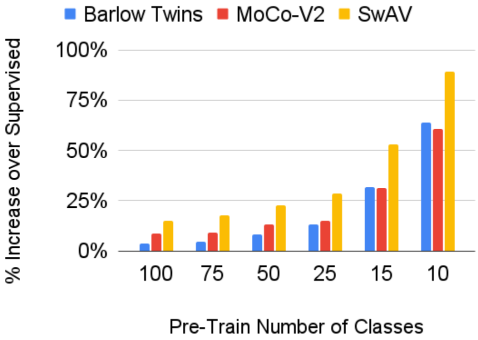

Figure 1: In this study, we found that self-supervised pre-training was more effective than supervised pre-training on ImageNet when less data was used. This figure shows the percentage increase in final top-1 accuracy on ImageNet of three self-supervised methods, MoCo-V2 [16], Barlow Twins [69], and SwAV [9], compared to supervised pre-training when paired with REMIND [29] as a function of the number of ImageNet classes used for pre-training.

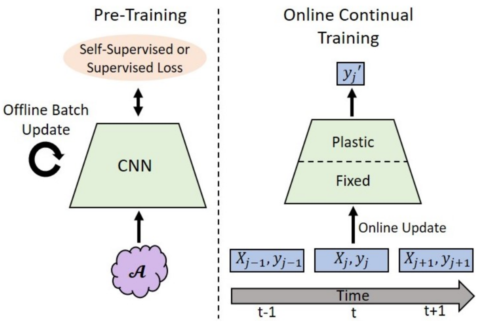

Figure 2: During pre-training (left), a CNN is trained offline on a (possibly labeled) dataset, $\mathcal{A}$. These initialized weights are copied into the continual learner's CNN, where a portion of the network is frozen and the rest is updated continually. In online learning (right), the learner receives a single labeled sample (X_j, y_j) at each time-step t. The continual learner first learns all samples from the pre-train dataset, $\mathcal{A}$, before learning samples from dataset $\mathcal{B}$.

especially when less data is used. Because supervised learning only requires features that discriminate among the classes in the pre-training set, it may not produce optimal representations for unseen categories. In contrast, self-supervised learning promotes the acquisition of category agnostic feature representations [9, 22]. Using these features could help close the generalization gap between CNNs trained offline and those trained in a continual manner. Here, we test this hypothesis using three self-supervised learning methods.

This paper makes the following contributions: **(1)** We compare the discriminative power of supervised features to self-supervised features learned with MoCo-V2 [16], Barlow Twins [69], and SwAV [9] as a function of the amount of training data used during pre-training. In offline linear evaluation experiments, we find that self-supervised features generalize better to ImageNet categories omitted from pre-training, with the gap being larger when less data is used. **(2)** We further study the ability of supervised and self-supervised features on datasets that were not used for pre-training. **(3)** We study the effectiveness of self-supervised features in three systems for online continual learning of additional categories in ImageNet. Across these algorithms, we found self-supervised features worked better when fewer categories were used for pre-training compared to supervised pre-training (see Fig. 1). **(4)** We set a new state of the art for online continual learning of ImageNet, where data is ordered by category, with a relative increase of 14.95% over prior work [29].

2 Problem Formulation

We study the common continual learning paradigm in which pre-training precedes continual learning [4, 10, 27, 29, 34, 35, 36, 55, 68, 72]. Formally, given a pre-training dataset $\{(X_i, y_i)\} \in \mathcal{A}$, with M images X_i and their corresponding labels $y_i \in \mathcal{Y}$, a set of parameters

θ are learned for a CNN using $\mathcal{A}$ in an offline manner, i.e., the learner can shuffle and loop over data. Here, we compare learning θ with supervised versus self-supervised approaches.

Continual learning begins after pre-training by first initializing the CNN parameters ψ to be continually updated to $\psi \leftarrow \theta$. The continual learning dataset $\{(X_j, y_j)\} \in \mathcal{B}$, with images X_j and their corresponding labels $y_j \in \mathcal{K}$ is then incrementally presented to the learner, which cannot loop over $\mathcal{B}$. We study the most general form of the problem, online continual learning, where examples are presented one-by-one to the learner and cannot be revisited without using auxiliary storage, which is kept at a fixed capacity. In incremental class learning, examples are presented in order of their category, and categories during pre-training and continual learning are disjoint, i.e, $\mathcal{Y} \cap \mathcal{K} = \emptyset$. Our training protocol is depicted in Fig. 2.

We focus on online continual learning because these systems are more general and can learn from data presented in any order [27, 28, 29], while most incremental batch learning implementations are bespoke to incremental class learning and require significant algorithmic changes for other orderings. In incremental batch learning, a system queues up examples in memory until it acquires a batch, which is typically about 100000 examples (100 classes) in papers using ImageNet [10, 55, 68]. Systems loop over the batch and then purge it from memory[1], and the next batch is acquired. The online formulation is more general, i.e., the batch size is one sample, it closely matches real-world applications, systems train $5-130\times$ faster [27, 29], and recent work has shown it works nearly as well as batch learning [29].

3 Related Work

3.1 Continual Learning

Deep neural networks suffer from catastrophic forgetting [48] when they are incrementally updated on non-stationary data streams. Catastrophic forgetting occurs when past representations are overwritten with new ones, causing a drop in performance on past data. There are three approaches to mitigating catastrophic forgetting [54]: 1) increasing model capacity to include new representations [2, 46, 47, 56, 58, 59, 61, 66, 67], 2) regularizing parameter updates such that parameters do not deviate too much from their past values [3, 12, 13, 18, 37, 41, 45, 57, 62, 70], and 3) replay (or rehearsal) models that cache previous data in an auxiliary memory buffer and mix it with new data to fine-tune the network [5, 10, 21, 29, 30, 34, 35, 55, 65, 68]. While network expansion and regularization methods have been popular, they do not easily scale to large datasets (see [6] for a review). Moreover, many of these methods require additional information such as task labels and task boundaries, or they perform poorly [27, 29, 36]. Our experiments use a single shared classifier during online learning, where task labels are unknown to models during training and evaluation. This setting is more realistic since task information isn't always available.

Recently, methods that use replay have demonstrated success for large-scale continual learning of the ImageNet dataset [5, 10, 21, 29, 34, 55, 65, 68]. All of these models follow a similar training paradigm where they are first pre-trained on a subset of 100 [10, 29, 55, 68] or 500 [21, 34, 65] ImageNet classes in an offline setting, where they can shuffle and loop over the data in batches. After pre-training, these methods are updated on the remaining ImageNet classes. Although the algorithms differ fundamentally, they all perform pre-training using a supervised cross-entropy loss and a fixed pre-train number of classes. Only [27] has investigated the impact of the size of the supervised pre-training set on a continual learner's

[1]With the exception of those samples cached in an auxiliary storage buffer, if one is used.

performance. However, we think this is an important aspect that should be investigated further as smaller pre-training sizes require less compute and less data.

3.2 Self-Supervised Learning

Self-supervised learning techniques have recently gained popularity because they now rival supervised learning techniques [9] without requiring labeled data, which can be expensive and difficult to obtain. Specifically, self-supervised learning methods use pretext tasks to learn visual features, where the network provides its own supervision during training. Different pretext tasks have been proposed including the prediction of image colorization [73], image rotation [24], and several others [8, 19, 20, 51]. Recent works use contrastive learning [26, 52], where the model learns which datapoints are similar or different based on feature similarity, and they have even surpassed supervised networks on many downstream tasks [9, 14, 15, 16, 25, 32, 40]. Popular contrastive learning methods include MoCo [32], SimCLR [14], MoCo-V2 [15], SimCLR-V2 [15], and SwAV [9]. SwAV [9] follows a different contrastive approach, where it performs a cluster assignment prediction instead of comparing features of instances directly. Barlow Twins [69] is a recent method that does not use contrastive learning. It enforces strong correlations between vector representations of distorted versions of the same image, and minimizes the redundancy between components of these vectors. Due to their competitive performance, we compare MoCo-V2, SwAV, and Barlow Twins in our experiments and discuss each method in detail in Sec. 4.2.

Self-supervised features are commonly tested by performing a linear evaluation on top of a frozen feature extractor, or by fine-tuning the network on separate tasks. In both cases, the entire dataset is expected to be available for pre-training. [50] demonstrates that self-supervision surpasses supervision on several downstream tasks, even when less labeled data is available. Our work differs from [50] in three ways: 1) we test newer self-supervised methods, 2) our downstream task is online continual learning for image classification, which assumes that only a small portion of the data is available for pre-training, and 3) we do not use controlled synthetic datasets and instead focus on high-resolution natural image datasets.

4 Algorithms

In our study, we compare online continual learning systems that use either self-supervised or supervised pre-training as a function of the size of the pre-training dataset. For all experiments, we use ResNet-18 as the CNN. For continual learning with the full-resolution ImageNet dataset, ResNet-18 has been adopted as the universal standard CNN architecture by the community [4, 5, 10, 21, 29, 29, 34, 55, 68]. Next, we give details for the pre-training approaches and then we describe the online continual learning algorithms.

4.1 Pre-Training Approaches

We describe the four pre-training algorithms we study. The self-supervised methods were chosen based on their strong performance for offline linear evaluation in their papers and because they were practical to train due to their relatively low computational costs.

Supervised: Supervised pre-training for continual learning is our baseline as it is the standard approach used [10, 27, 29, 34, 55, 68], where the CNN is trained using cross-entropy with the labels on the pre-training data. We followed the same protocol as [29] for pre-training, including the use of random resized crop and horizontal flip augmentation from [31]. We explore additional augmentations and longer training times in Sec. 6.3.

MoCo-V2: The original self-supervised MoCo [32] architecture builds a dynamic dictionary such that keys in the dictionary relate to training images. It then trains an encoder network such that query images should be similar to their closest key in the dictionary and further from dissimilar keys using a contrastive loss. MoCo-V2 [16] makes two additional improvements over MoCo: it uses an additional projection layer and blur augmentation.

Barlow Twins: Barlow Twins [69] makes the cross-correlation matrix across the outputs of two identical networks fed with distorted versions of an image close to the identity matrix. The objective function enforces the vector representations of distorted versions of the same image to be similar using an *invariance term*, while minimizing the redundancy between the components of these vectors using a *redundancy reduction term*. We chose Barlow Twins due to its novel objective function and competitive performance to contrastive learning methods.

SwAV: The self-supervised SwAV [9] method learns to assign clusters to different augmentations or "views" of the same image. Unlike standard contrastive methods, SwAV does not require direct pair-wise feature comparisons for its swapped prediction contrastive loss.

Pre-train models were trained using 4 TitanX GPUs (2015 edition) with 128 GB of RAM. Parameter settings for all approaches are in supplemental materials (Sec. S1.)

4.2 Online Continual Learning Models

We evaluate three online continual learning algorithms that use pre-trained features. They were chosen because they have been shown to get strong or state-of-the-art results on incremental class learning for ImageNet. For experiments with class-incremental learning on ImageNet, all continual learning methods first visit the pre-training set with online updates before observing the set that was not used for pre-training. We provide additional details below and parameter settings are in supplemental materials.

SLDA [27]: Deep Streaming Linear Discriminant Analysis (SLDA) keeps the pre-trained CNN features fixed. It solely learns the output layer of the network. It was shown to be extremely effective compared to earlier methods that do update the CNN, despite not using any auxiliary memory. It learns extremely quickly. SLDA stores a set of running mean vectors for each class and a shared covariance matrix, both initialized during pre-training and only these parameters are updated during online training. A new input is classified by assigning it the label of the closest Gaussian in feature space.

Online Softmax with Replay: Offline softmax classifiers are often trained with self-supervised features, so we created an online softmax classifier for continual learning by using replay to mitigate forgetting. Like SLDA, it does not learn features in the CNN after pre-training. When a new example is to be learned, it randomly samples a buffer of CNN embeddings to mix in 50 additional samples. This set of 51 samples is then used to update the weights using gradient descent. When the buffer reaches maximum capacity (735K samples = 1.5 GB), we randomly discard an example from the most represented class.

REMIND [29]: Unlike SLDA and Online Softmax with Replay, REMIND does not keep the CNN features fixed after pre-training. Instead, it keeps the lower layers of the CNN fixed, but allows the weights in the upper layers to change. To mitigate forgetting, REMIND replays mid-level CNN features that are compressed using Product Quantization (PQ) and stored in a buffer. After initializing the CNN features during pre-training, we use this same pre-training data to train the PQ model and store the associated compressed representations of this data in the memory buffer. We run REMIND with the settings from [29].

5 Experimental Setup

For continual learning, ImageNet ILSVRC-2012 is the standard benchmark for assessing the ability to scale [60]. It has 1.2 million images from 1000 classes, with about 1000 images per class used for training. Most existing papers use the first 100 classes (10%) for supervised pre-training [10, 29, 55, 68] or the first 500 (50%) classes [21, 34, 65]. An exception is [27], which studied performance as a function of the size of the supervised pre-training set and found that performance was highly dependent on the amount of data.

Following previous work [27, 29], we randomly select ImageNet classes for pre-training. We split ImageNet into pre-training sets for feature learning of various sizes: 10, 15, 25, 50, 75, 100 classes. In our experiments on ImageNet, after learning features on a pre-train set in an offline manner, the pre-train set and the remainder of the dataset are combined and then examples are fed one-by-one to the learner. Given our limited computational resources, we were not able to train for additional pre-training dataset sizes. In addition to doing continual learning on ImageNet itself, we also study how well the pre-trained features learned on ImageNet from various sizes of pre-train sets generalize to another dataset from an entirely different domain: scene classification. To do this we use the Places-365 dataset [74], which has 1.8 million images from 365 classes. We only use it for offline linear evaluation and continual learning, i.e., we do not perform pre-training on it. All of our continual learning experiments use the class incremental learning setting, where the data is ordered by class, but all images are shuffled within each class. We report top-1 accuracy on the validation sets.

6 Results

6.1 Offline Linear Evaluation Results

We compare the efficacy of self-supervised and supervised features as a function of the amount of pre-train data using the standard *offline* linear evaluation done for self-supervised learning [9, 14, 32, 69]. This lets us measure feature generalization when less data is used, without addressing catastrophic forgetting. Our offline linear evaluation uses a softmax classifier and results are in Table 1. Parameter settings are in supplemental materials.

We compared how effective pre-training approaches were when directly evaluated on the *same* classes used for pre-training. We had expected supervised features to outperform self-supervised features in this experiment, but surprisingly SwAV outperformed supervised features when 50 or fewer classes were used and it rivaled or exceeded performance when the number of classes was 75 or 100. MoCo-V2 and Barlow Twins only outperformed supervised features when using 15 or fewer classes.

We assessed the generalization of features to unseen categories by training the softmax classifier on all 1000 classes. SwAV outperformed all other features across all pre-train sizes. The gap in performance between self-supervised and supervised pre-training was larger when using fewer pre-train classes. SwAV outperformed supervised features by 3.24% and 20.67% when using 100 and 10 classes during pre-training, respectively. Thus, self-supervised features generalize better to unseen classes, especially for small pre-train sizes.

We examined the impact of the classes chosen for pre-training by randomly choosing 6 different sets of 10 pre-train classes from ImageNet. We trained an offline softmax classifier using these features on all 1000 ImageNet classes. The mean top-1 accuracy and standard deviation across runs was: 12.65%$\pm$1.26% for supervised, 22.78%$\pm$0.88% for MoCo-V2,

Table 1: Top-1 accuracies for offline linear evaluations with different features and varying numbers of pre-train classes. We show top-1 accuracy when evaluating on: 1) only pre-train ImageNet classes, 2) all 1000 ImageNet classes, and 3) all 365 Places classes.

Evaluation Set	Features	10	15	25	50	75	100
Pre-Train ImageNet	Supervised	71.20	82.67	86.64	83.16	**81.20**	80.32
	MoCo-V2	82.20	84.00	84.40	80.80	78.96	77.76
	Barlow Twins	85.20	85.33	83.60	78.60	75.73	74.52
	SwAV	**91.20**	**90.27**	**89.44**	**85.48**	81.04	**80.82**
Full ImageNet	Supervised	10.42	18.23	26.76	34.32	38.29	41.58
	MoCo-V2	21.76	27.50	30.61	36.45	40.15	43.19
	Barlow Twins	24.90	27.79	32.46	38.63	42.16	44.66
	SwAV	**31.09**	**33.62**	**35.81**	**39.54**	**43.43**	**44.82**
Full Places	Supervised	15.07	22.83	28.46	31.03	32.63	33.78
	MoCo-V2	24.07	29.45	31.01	32.58	34.54	35.93
	Barlow Twins	29.51	31.68	33.83	**36.94**	**38.59**	**39.37**
	SwAV	**32.71**	**33.88**	**35.10**	35.16	36.75	37.41

Table 2: Final top-1 accuracy values achieved by each continual learning method on ImageNet using various features and numbers of pre-train classes.

Method	Features	10	15	25	50	75	100
Deep SLDA	Supervised	9.53	14.67	20.7	26.54	29.53	**31.99**
	MoCo-V2	19.37	20.66	21.64	24.43	26.26	28.31
	Barlow Twins	18.48	20.10	23.02	27.45	30.25	31.81
	SwAV	**22.33**	**24.01**	**25.22**	**28.5**	**30.89**	31.77
Online Softmax	Supervised	9.87	15.92	23.58	30.16	33.05	35.14
	MoCo-V2	**23.09**	**25.67**	25.66	28.79	31.14	33.81
	Barlow Twins	20.30	22.18	26.01	30.47	33.39	35.06
	SwAV	18.65	21.40	**27.42**	**35.79**	**39.48**	**41.31**
REMIND	Supervised	19.79	26.17	33.48	39.38	43.40	45.28
	MoCo-V2	31.81	34.39	38.43	44.50	47.36	49.25
	Barlow Twins	32.43	34.52	37.94	42.65	45.30	46.85
	SwAV	**37.43**	**40.09**	**43.05**	**48.31**	**51.01**	**52.05**

24.35%±0.50% for Barlow Twins, and 31.88%±1.40% for SwAV. With only 10 classes, all self-supervised features outperformed supervised features. Since all standard deviations were small, we did not study more seeds for larger pre-train sets.

We evaluated how well the ImageNet pre-trained features transferred to Places-365. Similar to our results on ImageNet, self-supervised features outperformed supervised ones when pre-trained on ImageNet and evaluated on the Places-365 dataset. SwAV features outperformed MoCo-V2 features for all pre-train sizes, where more benefit is seen using fewer pre-train classes, while Barlow Twins outperformed both SwAV and MoCo-V2 when 50 or more classes are used during pre-training. These results demonstrate the versatility of self-supervised features in generalizing to datasets beyond what they were trained on.

6.2 Online Continual Learning Results

Table 2 shows online continual learning results with ImageNet for each pre-training method and online learning algorithm. For Deep SLDA, SwAV and Barlow Twins outperform super-

Table 3: Final top-1 accuracies for Deep SLDA when pre-trained on ImageNet using various features and pre-train classes and continually updated and evaluated on Places-365.

FEATURES	10	15	25	50	75	100
Supervised	13.77	19.59	23.42	25.28	26.04	27.03
MoCo-V2	23.36	23.72	24.05	24.17	25.11	26.00
Barlow Twins	24.14	25.63	27.27	**29.21**	**30.27**	**30.90**
SwAV	**26.53**	**27.20**	**28.22**	28.48	29.38	29.58

vised features when the pre-train size is 75 or smaller, but this was only true for MoCo-V2 with a pre-train size of 25 or less. Even when supervised pre-training outperforms self-supervised features, SwAV and Barlow Twins still show competitive performance, with gaps no greater than 4%. For Online Softmax, SwAV performed better than supervised features for all pre-train sizes. Barlow Twins surpasses supervised features for 75 classes or fewer sizes and it is competitive for 100. MoCo-V2 outperforms supervised features for 25 classes or fewer, but it shows competitive performance for other sizes as well. REMIND shows the most benefit from self-supervised pre-training, where SwAV, Barlow Twins, and MoCo-V2 consistently outperform supervised features for all pre-train sizes tested. REMIND using SwAV pre-training on only 10 classes surpasses all results obtained by Deep SLDA. With 50 class pre-training, REMIND with SwAV outperforms supervised pre-training on 100 classes, showing that SwAV can surpass supervised features using half the pre-train data.

Fig. 1 shows the relative percentage increase of self-supervision over supervision for REMIND, with a maximum relative improvement of 89.14% for SwAV, 63.87% for Barlow Twins, and 60.74% for MoCo-V2 when only 10 classes are used for pre-training.

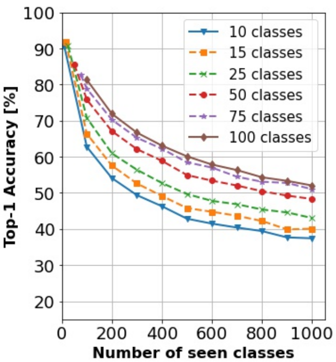

Figure 3: Learning curve for REMIND with SwAV features at various pre-train sizes.

Fig. 3 shows learning curves for REMIND using SwAV features for various pre-train sizes when multiples of 100 classes have been seen by the model. More learning curves are in supplemental materials. Adding more classes during pre-training consistently improves REMIND's performance. To compare with the original REMIND paper, we compute the average top-5 accuracy of REMIND using SwAV features pre-trained on 100 classes of ImageNet and evaluated on seen classes at increments of 100 classes. This variant of REMIND with SwAV achieves an average top-5 accuracy of 83.55%, which is a 4.87% absolute percentage improvement over the supervised results in [29]. This variant also achieves a 14.95% relative increase in final top-1 accuracy over supervised features. This sets a new state of the art result for online learning of ImageNet.

Domain Transfer with Deep SLDA: So far, we have studied the effectiveness of supervised and self-supervised pre-training to generalize to unseen classes from the same dataset. A natural next step is to study how well self-supervised pre-training of features on one dataset generalize to continual learning on another dataset, e.g., pre-train on ImageNet and continually learn on Places-365. This is similar to standard transfer learning and domain adaptation setups [43, 44, 53, 71], and could prove useful when performing continual learning on small

Table 4: Final top-1 accuracies on ImageNet for offline linear evaluations and online learning using REMIND using various pre-train sizes. We compare the performance Supervised$_{LT}$, Supervised$_{SA}$, Supervised, and SwAV features. For offline linear, we evaluate features on both the pre-train classes and all 1000 classes. REMIND evaluates on all 1000 classes.

METHOD	EVAL. SET	FEATURES	10	15	25	50	75	100
Offline Linear	*Pre-Train*	Supervised	71.20	82.67	86.64	83.16	81.20	80.32
		Supervised$_{LT}$	91.20	87.87	90.08	85.52	81.63	80.92
		Supervised$_{SA}$	**94.80**	**94.00**	**91.92**	**88.48**	**86.56**	**85.66**
		SwAV	91.20	90.27	89.44	85.48	81.04	80.82
Offline Linear	*Full*	Supervised	10.42	18.23	26.76	34.32	38.29	41.58
		Supervised$_{LT}$	24.85	27.34	30.89	34.11	36.37	39.80
		Supervised$_{SA}$	27.88	31.34	35.13	**39.57**	42.28	44.76
		SwAV	**31.09**	**33.62**	**35.81**	39.54	**43.43**	**44.82**
Online (REMIND)	*Full*	Supervised	19.79	26.17	33.48	39.38	43.40	45.28
		Supervised$_{LT}$	31.49	33.71	36.64	41.07	43.59	45.88
		Supervised$_{SA}$	34.59	37.35	42.23	46.20	49.32	50.82
		SwAV	**37.43**	**40.09**	**43.05**	**48.31**	**51.01**	**52.05**

datasets that require generalizable feature representations. To evaluate our hypothesis, we trained Deep SLDA on Places-365 using various pre-trained ImageNet features. We chose Deep SLDA since it is extremely fast to train. We start online learning with the first sample in Places-365 and learn one class at a time from mean vectors initialized as zeros and a covariance matrix initialized as ones, as in [27]. Results are in Table 3. SwAV features generalized the best for pre-train sizes less than or equal to 25 classes, while Barlow Twins features generalized better for 50 or more pre-train classes. Thus, self-supervised features improve online learning, even when the pre-train and continual datasets differ.

6.3 Impact of Augmentation and Pre-Training Time on Performance

Our main experiments followed [31] and used random resized crops and horizontal flip data augmentations for supervised learning. Further, following [29], we trained supervised models for 40 epochs. However, all self-supervised methods use additional augmentation techniques and longer training times, so we study supervised pre-training under similar conditions. To do this, we adopt SwAV augmentations (multi-crop, color jitter, gaussian blur, grayscale, and horizontal flips) and training time (400 epochs) for supervised pre-training. We modeled experiments after SwAV since it was the top-performing self-supervised method in Sec. 6.1 and Sec. 6.2. We define two supervised configurations: Supervised Long Training (Supervised$_{LT}$) trains with random resized crop and horizontal flip augmentations [31]; and Supervised Long Training with SwAV Augmentations (Supervised$_{SA}$) trains with SwAV augmentations [9]. Both methods train for 400 epochs with a learning rate of 0.01. All other parameters are the same as in S1.1. We compare the supervised features with SwAV features in the offline linear setting and online setting using REMIND. The results are in Table 4.

Across experiments, Supervised$_{LT}$ features either outperformed or were competitive with supervised features. Similarly, we found that using longer training times and SwAV data augmentations (Supervised$_{SA}$) yielded the best supervised feature performance. These results indicate that longer training and additional data augmentations can improve the quality of supervised features. For offline linear evaluations on the pre-training classes, Supervised$_{SA}$

features always performed better than SwAV features, with a maximum gap of 5.56%. However, when performing the offline linear evaluation on all 1000 classes, we found that SwAV outperformed or performed competitively with Supervised$_{SA}$. Similarly, SwAV features outperformed all variants of supervised features in the online setting with REMIND. While Supervised$_{LT}$ and Supervised$_{SA}$ features improved supervised performance, SwAV performed competitively or better than supervised features when evaluating on the full dataset.

7 Discussion and Conclusion

We replaced supervised pre-training with self-supervised techniques for online continual learning on ImageNet. We found that self-supervised methods require fewer pre-training classes to achieve high performance on image classification compared to supervised pre-training. This behavior is seen during offline linear evaluation and online learning and hypothesize that it is due to the generalizability of self-supervised features to unseen classes.

We followed others in using the ResNet-18 CNN architecture [27, 29, 55]; however, wider and deeper networks have been shown to improve performance, especially for self-supervised learning [14, 15, 25]. It would be interesting to explore these alternate architectures for online learners initialized with self-supervised features. Further, since online and batch learning share a similar pre-training procedure, we hypothesize that self-supervised features would benefit incremental batch learning models [55, 68]. Future work could also explore self-supervised models pre-trained on large datasets for online continual learning, since self-supervised learning shows even more benefits when trained with large amounts of data. Moreover, future studies could investigate semi-supervised pre-training. Semi-supervised learning would be advantageous to supervised learning because it requires fewer labels, but could still generalize similarly to self-supervised techniques. It would also be interesting to develop self-supervised or semi-supervised online updates for continual learning. These could be used to update the plastic layers of REMIND, which could improve performance and generalization further. Self-supervised features have been evaluated for several downstream tasks in offline settings [32] and it would be interesting to evaluate how well self-supervised features perform for other downstream continual learning tasks, e.g., continual object detection [1, 63], continual semantic segmentation [11, 49], or continual learning for robotics [23, 39]. Our work could also facilitate downstream open world learning [7] or automatic class discovery [64, 75], where an agent must identify samples outside of its training distribution as unknown and then learn them. We showed that self-supervised features generalize to unseen classes/datasets in online settings, so extending our work to open world learning would only require a component to identify unknown samples [33, 38, 42].

While much progress has been made to develop continual learners that scale, existing approaches require an initial supervised pre-training phase. We showed that self-supervised pre-training consistently outperformed supervised pre-training and set a new state-of-the-art for online learning by pairing REMIND with SwAV features. Self-supervised learning is beneficial as it reduces overfitting, promotes generalization, and does not require labels.

Acknowledgements. This work was supported in part by the DARPA/SRI Lifelong Learning Machines program [HR0011-18-C-0051], AFOSR grant [FA9550-18-1-0121], and NSF award #1909696. The views and conclusions contained herein are those of the authors and should not be interpreted as representing the official policies or endorsements of any sponsor. We thank fellow lab member Manoj Acharya for his comments and useful discussions.

References

[1] Manoj Acharya, Tyler L Hayes, and Christopher Kanan. Rodeo: Replay for online object detection. In *BMVC*, 2020.

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

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

[4] Eden Belouadah and Adrian Popescu. Deesil: Deep-shallow incremental learning. In *ECCV*, 2018.

[5] Eden Belouadah and Adrian Popescu. Il2m: Class incremental learning with dual memory. In *ICCV*, 2019.

[6] Eden Belouadah, Adrian Popescu, and Ioannis Kanellos. A comprehensive study of class incremental learning algorithms for visual tasks. *Neural Networks*, 2020.

[7] Abhijit Bendale and Terrance Boult. Towards open world recognition. In *CVPR*, 2015.

[8] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *ECCV*, 2018.

[9] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *NeurIPS*, 2020.

[10] Francisco M Castro, Manuel J Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In *ECCV*, 2018.

[11] Fabio Cermelli, Massimiliano Mancini, Samuel Rota Bulo, Elisa Ricci, and Barbara Caputo. Modeling the background for incremental learning in semantic segmentation. In *CVPR*, 2020.

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

[13] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with A-GEM. In *ICLR*, 2019.

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

[15] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. In *NeurIPS*, 2020.

[16] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

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

[18] Prithviraj Dhar, Rajat Vikram Singh, Kuan-Chuan Peng, Ziyan Wu, and Rama Chellappa. Learning without memorizing. In *CVPR*, 2019.

[19] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *CVPR*, 2015.

[20] Alexey Dosovitskiy, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with convolutional neural networks. In *NeurIPS*, 2014.

[21] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *ECCV*, 2020.

[22] Linus Ericsson, Henry Gouk, and Timothy M Hospedales. How well do self-supervised models transfer? In *CVPR*, 2021.

[23] Fan Feng, Rosa HM Chan, Xuesong Shi, Yimin Zhang, and Qi She. Challenges in task incremental learning for assistive robotics. *IEEE Access*, 8, 2019.

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

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

[26] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *CVPR*, 2006.

[27] Tyler L Hayes and Christopher Kanan. Lifelong machine learning with deep streaming linear discriminant analysis. In *CVPR-W*, 2020.

[28] Tyler L Hayes, Nathan D Cahill, and Christopher Kanan. Memory efficient experience replay for streaming learning. In *ICRA*, 2019.

[29] Tyler L Hayes, Kushal Kafle, Robik Shrestha, Manoj Acharya, and Christopher Kanan. Remind your neural network to prevent catastrophic forgetting. In *ECCV*, 2020.

[30] Tyler L Hayes, Giri P Krishnan, Maxim Bazhenov, Hava T Siegelmann, Terrence J Sejnowski, and Christopher Kanan. Replay in deep learning: Current approaches and missing biological elements. *arXiv preprint arXiv:2104.04132*, 2021.

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

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

[33] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *ICLR*, 2017.

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

[35] Ronald Kemker and Christopher Kanan. Fearnet: Brain-inspired model for incremental learning. In *ICLR*, 2018.

[36] Ronald Kemker, Marc McClure, Angelina Abitino, Tyler Hayes, and Christopher Kanan. Measuring catastrophic forgetting in neural networks. In *AAAI*, 2018.

[37] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *PNAS*, 114, 2017.

[38] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *NeurIPS*, 2018.

[39] Timothée Lesort, Vincenzo Lomonaco, Andrei Stoian, Davide Maltoni, David Filliat, and Natalia Díaz-Rodríguez. Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges. *Information Fusion*, 58, 2020.

[40] Junnan Li, Pan Zhou, Caiming Xiong, Richard Socher, and Steven CH Hoi. Prototypical contrastive learning of unsupervised representations. In *ICLR*, 2021.

[41] Zhizhong Li and Derek Hoiem. Learning without forgetting. *TPAMI*, 40, 2017.

[42] Shiyu Liang, Yixuan Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *ICLR*, 2018.

[43] Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and Philip S Yu. Transfer feature learning with joint distribution adaptation. In *ICCV*, 2013.

[44] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *ICML*, 2015.

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

[46] Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *CVPR*, 2018.

[47] Arun Mallya, Dillon Davis, and Svetlana Lazebnik. Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In *ECCV*, 2018.

[48] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24. 1989.

[49] Umberto Michieli and Pietro Zanuttigh. Incremental learning techniques for semantic segmentation. In *ICCV-W*, 2019.

[50] Alejandro Newell and Jia Deng. How useful is self-supervised pretraining for visual tasks? In *CVPR*, 2020.

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

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

[53] Sinno Jialin Pan, Ivor W Tsang, James T Kwok, and Qiang Yang. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2), 2010.

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

[55] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *CVPR*, 2017.

[56] Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Efficient parametrization of multi-domain deep neural networks. In *CVPR*, 2018.

[57] Hippolyt Ritter, Aleksandar Botev, and David Barber. Online structured laplace approximations for overcoming catastrophic forgetting. In *NeurIPS*, 2018.

[58] Amir Rosenfeld and John K Tsotsos. Incremental learning through deep adaptation. *TPAMI*, 42, 2018.

[59] Deboleena Roy, Priyadarshini Panda, and Kaushik Roy. Tree-cnn: a hierarchical deep convolutional neural network for incremental learning. *Neural Networks*, 2020.

[60] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *IJCV*, 115, 2015.

[61] 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.

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

[63] Konstantin Shmelkov, Cordelia Schmid, and Karteek Alahari. Incremental learning of object detectors without catastrophic forgetting. In *ICCV*, 2017.

[64] Josef Sivic, Bryan C Russell, Alexei A Efros, Andrew Zisserman, and William T Freeman. Discovering object categories in image collections. In *ICCV*, 2005.

[65] Xiaoyu Tao, Xinyuan Chang, Xiaopeng Hong, Xing Wei, and Yihong Gong. Topology-preserving class-incremental learning. In *ECCV*, 2020.

[66] Xiaoyu Tao, Xiaopeng Hong, Xinyuan Chang, Songlin Dong, Xing Wei, and Yihong Gong. Few-shot class-incremental learning. In *CVPR*, 2020.

[67] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Growing a brain: Fine-tuning by increasing model capacity. In *CVPR*, 2017.

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

[69] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. *arXiv preprint arXiv:2103.03230*, 2021.

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

[71] Jing Zhang, Wanqing Li, and Philip Ogunbona. Joint geometrical and statistical alignment for visual domain adaptation. In *CVPR*, 2017.

[72] Junting Zhang, Jie Zhang, Shalini Ghosh, Dawei Li, Serafettin Tasci, Larry Heck, Heming Zhang, and C-C Jay Kuo. Class-incremental learning via deep model consolidation. In *WACV*, 2020.

[73] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *ECCV*, 2016.

[74] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *TPAMI*, 2017.

[75] Jun-Yan Zhu, Jiajun Wu, Yan Xu, Eric Chang, and Zhuowen Tu. Unsupervised object class discovery via saliency-guided multiple class learning. *TPAMI*, 37, 2014.