Few-Shot Class Incremental Learning Leveraging Self-Supervised Features

Touqeer Ahmad Akshay Raj Dhamija Steve Cruz
Ryan Rabinowitz Chunchun Li Mohsen Jafarzadeh Terrance E. Boult
Vision and Security Technology Lab, University of Colorado Colorado Springs, Colorado Springs, CO, USA
{touqeer, adhamija, scruz, rrabinow, cli, mjafarzadeh, tboult}@vast.uccs.edu

Abstract

Few-Shot Class Incremental Learning (FSCIL) is a recently introduced Class Incremental Learning (CIL) setting that operates under more constrained assumptions: only very few samples per class are available in each incremental session, and the number of samples/classes is known ahead of time. Due to limited data for class incremental learning, FSCIL suffers more from over-fitting and catastrophic forgetting than general CIL. In this paper we study leveraging the advances due to self-supervised learning to remedy over-fitting and catastrophic forgetting and significantly advance the state-of-the-art FSCIL. We explore training a lightweight feature fusion plus classifier on a concatenation of features emerging from supervised and self-supervised models. The supervised model is trained on data from a base session, where a relatively larger amount of data is available in FSCIL. Whereas a self-supervised model is learned using an abundance of unlabeled data. We demonstrate a classifier trained on the fusion of such features outperforms classifiers trained independently on either of these representations. We experiment with several existing self-supervised models and provide results for three popular benchmarks for FSCIL including Caltech-UCSD Birds-200-2011 (CUB200), miniImageNet, and CIFAR100 where we advance the state-of-the-art for each benchmark. Code is available at: https://github.com/TouqeerAhmad/FeSSSS

1. Introduction

As deep learning models emerge from their infancy and are being deployed in the real world, more of their limitations have been identified, e.g., these models classify only a fixed set of classes and are generally trained with large amounts of data. This naturally leads to several interesting and practical problems, *e.g.*, task incremental learning [19, 46], class incremental learning [13, 29, 42, 47], continual learning [35], few-shot learning [51, 54, 58, 63], open-set recognition [6, 49], and open-world learning [7, 17].

Inspired by few-shot learning, Tao et al. [51] recently addressed class incremental learning in an even more challenging and practical setting, i.e., *Few-Shot Class Incremental Learning* (FSCIL) where only K shots/samples per class are available and K is very small (5 samples per class) than general class incremental learning. As we highlight in Fig 1, due to more constrained assumptions, FSCIL suffers more over-fitting on the "few" classes. It also suffers catastrophic forgetting of old classes. As a comparison, in general CIL, even the number of exemplars retained per class are way more (generally more than 20) than the total samples per class (typically 5) in the incremental sessions of FSCIL.

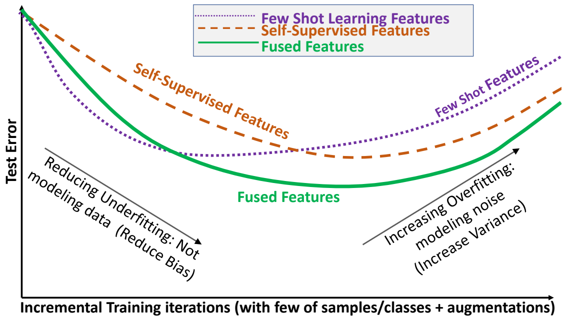

Figure 1. Bias and variance have an inherent trade-off and are normally considered with respect to model complexity. However, feature spaces also have an important role in the trade-off and the resulting test error. Few-shot learning (dotted curve) using supervised data, reduces bias quickly. However, it only has a small sample of data, hence higher variance, and hence over-fits that data earlier. Self-supervised feature spaces (dashed line) are trained with lots of data but have larger bias since they are only weakly related to actual classes of interest. Either large bias or large variance limits overall accuracy. Appropriate fusing of these two feature spaces (solid line) can produce a model with much better test error. These logical curves are for a single iteration; in *Few-Shot Class-Incremental Learning* the problem is compounded as extending the model for each new set of classes exasperates the problem of over-fitting. We show that while self-supervised features alone may not improve performance, combining a lightweight multi-layer network to fuse standard few-shot features with self-supervised features significantly improves the state-of-the-art in few-shot class incremental learning.

Given limited data availability in a FSCIL setting, it is natural to explore if other features can remedy the inherent catastrophic forgetting and over-fitting, which is the focus of this study.

In a traditional learning paradigm, it was always assumed that labeled data belonging to all classes is readily available with a large number of samples per class. However, as deep learning migrates from academics to industry, it became evident that labeling is laborious, costly, time-consuming, and prone to human errors/biases. Additionally, not all classes of interest are known beforehand, they may become available periodically and a learner may not have a large number of labeled samples per class. To circumvent the labeling issue there has been a surge in approaches targeting training of deep networks without labels/supervision, i.e., *self-supervised learning* [10, 11, 14, 21–23, 28, 32, 43–45, 60, 61]. Self-supervised learning has been demonstrated to compete or even outperform pure supervised models on downstream tasks of image classification and object detection. Recently, self-supervised features have been deployed even for other novel downstream tasks [9, 17, 20, 24]. On the other hand, to address the periodic nature of data and re-purpose a single model for various tasks, fields of continual learning, task incremental learning, and class incremental learning have been developed where a significant progress has been achieved within a relatively shorter time-span. While continual learning and task incremental learning have their own applications, class incremental learning is more practical. Class incremental learning requires a model to learn new classes while maintaining the knowledge of old classes that it has previously learned. To address catastrophic forgetting many approaches have been investigated, e.g., retaining exemplars, knowledge distillation, temperature scaling – to name a few.

We investigate fusion of several self-supervised models for FSCIL and use established benchmarks to demonstrate that a lightweight classification module trained on combined supervised and self-supervised features results in outperforming the state-of-the-art methods. Specifically, we leverage the self-supervised models trained on ImageNet-2012 [50] or OpenImages-v6 [34], and a supervised model trained on data belonging to base session where a relatively larger number of labeled images are available. Both supervised and self-supervised models are kept frozen after the initial training on base session and disjoint unlabeled data set respectively, and serve as first-level feature extractors for the subsequent incremental sessions. A lightweight two layer classification head is employed as the learnable module that adapts over time with data for each incremental session. During training, the classification module is trained on the concatenation of independently normalized features emerging from images belonging to the respective incremental sessions. Since images are not directly fed to the classification module, several feature vectors per image are generated with conventional data augmentation techniques. To further mitigate catastrophic forgetting of old classes, we develop a Gaussian Generator. The Gaussian's centroid vector per class can be thought of as a single exemplar per class in a conventional CIL setting and aligns well with limited resource constraint of FSCIL. We generalize this by generating synthetic data by additionally maintaining a scalar/vector for variance and assuming classes to be multivariate Gaussian from which we can sample. An ablation study is conducted to demonstrate that such synthetically generated data further improves the performance.

Our Contributions

- First to investigate self-supervised learning for the challenging downstream task of FSCIL.
- An effective learning paradigm (named FeSSSS) fusing supervised and self-supervised features demonstrating enhanced performance rather than relying on either one of them independently. This approach can further benefit from future advances in self-supervised learning and/or supervised FSCIL.
- Demonstrate that our novel Gaussian Generator reduces catastrophic forgetting in FSCIL.
- New state-of-the-art on established benchmarks of FSCIL statistically significantly outperforming existing methods.
- Ablation study demonstrating that both fusion and our Gaussian Generator statistically significantly matter.

2. Related Work

2.1. Incremental Learning

While there are multiple subtypes of incremental learning, herein we focus on class incremental learning.

Nearest Mean Classifier (NMC) [40,41] represents each class using a prototype vector that is the mean of all the examples seen for that class. Approaches such as DeeSIL [2] and DeepSLDA [26] attempt to classify feature representations using independent classifiers such as SVMs.

Most recent class incremental learning tries to address *catastrophic forgetting* and *concept drift* by partially/fully retraining the network. [36] attempted to address catastrophic forgetting by introducing knowledge distillation in the loss function. Another common approach to circumvent catastrophic forgetting is by maintaining *exemplars*, *i.e.*, some samples from old classes are retained in the replay buffer. The network for the next incremental phase is then trained on both the new classes and the exemplars [27,48,56]. Choosing these exemplars is also an active research area, *e.g.*, methods like *herding* [55] and *mnemonics* [37]. PODNet [19] studied *rigid-plasticity trade-off* where the network learns to balance between remembering the old classes (*rigidity*) and learning new ones (*plasticity*). Unlike other methods [31, 37, 56] which typically employ iCaRL protocol [48] where five or more classes are introduced per incremental-task, in PODNet [19] proposed and other methods [30, 48, 56] are additionally evaluated on even learning one class per task. [27] tried to address the incremental learning in a more challenging online setting. Each online incremental learning phase was followed by an offline retraining phase where all the data available up to that point was used to retrain the network. They also maintained an exemplar set and employed herding [55]. Recently, there have been more attempts towards incremental learning [46, 62], and detailed surveys on the topic also exist [38, 39]. In [5], a comprehensive evaluation of recently proposed incremental learning approaches [2–4, 8, 25, 26, 30, 36, 42, 48, 56] was conducted.

Few-Shot Class Incremental Learning (FSCIL) Few-shot learning itself is a very active area of research with hundreds of papers [54]. We focus here on related work on FSCIL, which has different challenges than few-shot learning, since the representations must adapt over time and is a harder problem than classic class incremental learning because of the limited data per class.

FSCIL was first introduced in [51] where authors proposed a TOPIC framework that used a *neural gas* (NG) network to learn feature space topologies formed by different classes for knowledge representation. To mitigate the catastrophic forgetting of old classes, they stabilized the topology of NG while adapting it to enhance the discriminative power of learned features for few-shot new classes. They adapted a number of general CIL approaches [13, 29, 47] for FSCIL and demonstrated TOPIC outperformed all of them on benchmark data sets. [63] proposed a prototype-based FSCIL approach where they introduced a *Self-Promoted Prototype Refinement* (SPPR) scheme to update the existing prototypes by utilizing a relation matrix between representation of the new class samples and the old class prototypes. They employed a random episode selection strategy to enhance the extensibility of feature representation to circumvent severe catastrophic forgetting inherent to FSCIL. In ERL [18], authors focused on stability-plasticity dilemma and proposed *exemplar relation distillation incremental learning* framework to balance the tasks of old-knowledge preservation and new-knowledge adaptation. In [59], authors devised a decoupled learning strategy for representations and classifiers where only the classifiers are updated in each incremental session to avoid knowledge forgetting. To propagate context information between classifiers learned on individual incremental sessions, they employed a graph model and proposed *Continually Evolved Classifier* (CEC). A pseudo incremental learning paradigm is further designed to enable the learning of CEC. In addition to comparing against TOPIC, they also adapted a couple of few-shot methods [52, 58] for FSCIL by inducing their decoupled learning strategy. To the best of our knowledge, CEC is the best performing FSCIL approach prior to this writing.

In the latest FSCIL work [16], authors proposed *Semi-Supervised Few-Shot Class Incremental Learning* (SSFSCIL) approach where they relied on knowledge distillation and unlabeled data to boost the performance of existing CIL approaches [29, 47] adapted for FSCIL. For each incremental session, in addition to *N*-way *K*-shot data, many samples from *N* classes without labels are additionally used for which the predictions are refined over time by means of knowledge distillation. More unlabeled samples are sequentially added to the labeled set along with their predicted pseudo-labels and representation is revised with way more data than conventional *K*-shots per class. The proposed approach is demonstrated to improve the baseline iCaRL [47] and NCM [29] methods, however, as evidenced in Tab. 1, SSFSCIL is outperformed by pure supervised approaches, e.g., CEC, ERL/ERL++, by a large margin.

2.2. Self-Supervised Learning

Self-supervised learning is an active research area where many approaches have emerged in recent years to learn better feature representations without any supervision and labeling. Self-supervised learning has been accomplished by solving a *pretext task* [21–23, 32, 43–45, 60, 61] using *contrastive loss* [14, 28] or by *clustering* [10, 11] the underlying deep features. Generally, models learned in a self-supervised manner are evaluated on the downstream task of object recognition, using ImageNet-2012 [50], by training a classification head. However, there have been recent studies where self-supervised learning has also been explored for other downstream tasks such as incremental/open-world learning [9, 17], continual learning [20], and novel class discovery and recognition [24]. Inspired by these recent advances, we explore the suitability of self-supervised learning for challenging FSCIL problem.

3. Problem Statement

Few-shot class incremental learning operates in various incremental sessions where few labeled samples per class become available to the learner in each session. The objective of the underlying model is to learn new concepts while retaining the knowledge of old ones. Following [51, 59], let $\{\mathcal{D}^0_{train}, \mathcal{D}^1_{train}, \cdots, \mathcal{D}^n_{train}\}$ be the training sets for n incremental sessions and class labels for i-th session, i.e., $\mathcal{D}^i_{train}$ is denoted by $\mathcal{C}^i$. The classes added in different sessions do not have any overlap, i.e., $\forall i, j$ where $i \neq j, \mathcal{C}^i \cap \mathcal{C}^j = \emptyset$. After each incremental session i, the model is evaluated on test data belonging to the current session and classes seen in all previous sessions, i.e., $\mathcal{C}^0 \cup \mathcal{C}^1 \cdots \cup \mathcal{C}^i$. In FSCIL, it is conventional to have way more training data in the base session ($\mathcal{D}^0_{train}$) than in the

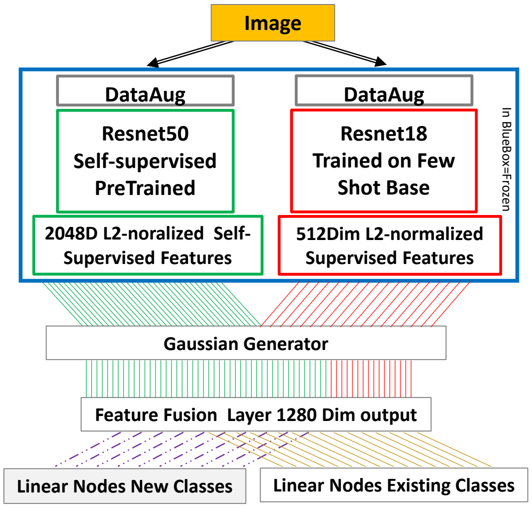
Image
DataAug
DataAug
Resnet50
Self-supervised
PreTrained
Resnet18
Trained on Few
Shot Base
In BlueBox=Frozen
2048D L2-noralized Self-Supervised Features
512Dim L2-normalized Supervised Features
Gaussian Generator
Feature Fusion Layer 1280 Dim output
Linear Nodes New Classes
Linear Nodes Existing Classes

Figure 2. The overall architecture of our ***Fe**w-**S**hot **S**elf-**S**upervised **S**ystem* (FeSSSS). The initial training uses the few-shot base session and disjoint unlabeled data after which the core feature extractors (in the blue box) are frozen. The ResNet-18 for few-shot supervised training produces 512 dimensional vectors. The self-supervised ResNet-50 produces 2048 dimensional output. These ResNets are frozen and not updated after initial training. The L^2 normalized features are concatenated and fed into the Gaussian Generator that learns models for new classes while passing along their features. During incremental training the Gaussian Generator samples from its model and provides features for old classes. The features are fed into the lightweight Feature Fusion Layer, which learns to map from the 2560 raw feature dimension into a fused 1280 dimensional output feature. These in turn feed through a fully-connected layer, the linear nodes that provide the final classifications. When new classes need to be added, new nodes are created and connected to the feature fusion layer and these new connections are initialized with random weights (dashed lines). With new class data, the linear classifiers and the feature fusion layer update, but the nodes for existing classes and part of feature fusion layer are warm started keeping their prior data as respective centroids. To address catastrophic forgetting, the network is then trained with the new data plus the centroids (from frozen features) of the existing classes.

incremental sessions where N-way K-shot setting is employed, i.e., each incremental session has N classes and only K samples per class are available. For our approach we assume there exist another unlabeled data set $\mathcal{D}^{u}_{train}$ that is disjoint with data for any of the sessions in FSCIL. The self-supervised model is learned on this disjoint data set.

4. Method

The overall architecture of ***Fe**w-**S**hot **S**elf-**S**upervised **S**ystem* (FeSSSS) is summarized in Fig. 2. The core of the method are the major elements: the feature extraction, the feature fusion, the linear classifier, and our Gaussian Generator to address catastrophic forgetting. Below, we describe each component of our pipeline, code is available on our GitHub page.

4.1. Supervised & Self-Supervised Feature Extractors

A typical deep learning model can be thought of as a composition of a feature extractor $\hat{x} = f(x; \theta)$ followed by a classification head $c(\hat{x}; \phi)$; where θ, and ϕ are learnable parameters, and x (an image), $\hat{x}$ (feature vector) are the inputs for respective modules. During training, these parameters are learned using data in a supervised or self-supervised manner depending upon the setting. In our hybrid framework FeSSSS, we train one deep model ($f_s(x; \theta_s)$, $c_s(\hat{x}; \phi_s)$) using data from base session $\mathcal{D}^{0}_{train}$ in a supervised manner, and another network ($f_{ss}(x; \theta_{ss})$, $c_{ss}(\hat{x}; \phi_{ss})$) is trained on $\mathcal{D}^{u}_{train}$ in a self-supervised manner. We do not assume any fixed task for the self-supervised model and it can be learned in any conventional manner, i.e., using a *pretext task*, employing *contrastive loss*, or through *clustering*. Once the two models are trained fully on their respective data sets, their classification heads are discarded and outputs from feature extractors $f_s(x; \theta_s)$, $f_{ss}(x; \theta_{ss})$ are normalized to have unit L^2 norm yielding $(\bar{x}_s, \bar{x}_{ss})$ which are then used for input to the lightweight classification module.

4.2. Lightweight Incremental Feature Fusion and Classification Module

Our model operating in the incremental setting is comprised of a lightweight network $l_c(\theta_c)$ that has two fully connected layers followed by a Softmax. It takes concatenated normalized feature vectors $\bar{x}_t = (\bar{x}_s | \bar{x}_{ss})$ as input and provides the probability vector for n classes that have been enrolled up to the current incremental session. The number of nodes in the intermediate feature fusion layer is set to half of the feature dimension of the concatenated vector $\bar{x}_t$, whereas, the number of output nodes are equal to the number of total classes enrolled so far and grow with each incremental session. This lightweight module is initially trained with the base class data, giving the initial training of feature fusion considerably more data than in each increment.

In each incremental session the lightweight model $l_c(\theta^{i}_{c})$ is initialized with weights from the previous session θ^{i-1}_{c} and N more nodes are added to the output layer. The weights for these new connections are randomly initialized. After training each incremental session, the model is evaluated on test samples belonging to all classes that have been enrolled so far. Importantly, the weights between the input normalized concatenated features are retained so that the system continues to better learn feature fusion over time. The new nodes, while randomly initialized, can exploit those fused features. If the system only used a simple linear classifier (linear layer), even retaining weights for known classes would not allow learning to fuse since the new classes would have no access to that information. We show in the ablation

Table 1. Comparison of FeSSSS with the state-of-the-art on CUB200 data set, DeepCluster-v2 [11] trained on ImageNet-2012 has been used as self-supervised feature extractor. ‡ indicates results reported in [59], * identifies the few-shot approaches adapted by [59] for FSCIL, and † shows the results for approaches taken from their respective papers. Our proposed approach based on the concatenation of supervised and self-supervised features outperforms the latest FSCIL methods. Our relative performance gain with respect to each approach in terms of average incremental accuracy is noted in the last column. Using a two-sided t-test with each iteration as the data, our approach is statistically significantly better than the state-of-the-art with $p < .0001$

Method	Acc. in each session (%) ↑											Avg. ↑	our relative improvement
	0	1	2	3	4	5	6	7	8	9	10		
Ft-CNN‡	68.68	43.7	25.05	17.72	18.08	16.95	15.1	10.6	8.93	8.93	8.47	22.02	**40.83**
NCM‡ [29]	68.68	57.12	44.21	28.78	26.71	25.66	24.62	21.52	20.12	20.06	19.87	32.49	**+30.36**
EEIL‡ [13]	68.68	53.63	47.91	44.2	36.3	27.46	25.93	24.7	23.95	24.13	22.11	36.27	**+26.58**
iCaRL‡ [47]	68.68	52.65	48.61	44.16	36.62	29.52	27.83	26.26	24.01	23.89	21.16	36.67	**+26.18**
TOPIC‡ [51]	68.68	62.49	54.81	49.99	45.25	41.4	38.35	35.36	32.22	28.31	26.28	43.92	**+18.93**
LEC-Net† [57]	70.86	58.15	54.83	49.34	45.85	40.55	39.70	34.59	36.58	33.56	31.96	45.08	**+17.77**
SS-iCaRL† [16]	69.89	61.24	55.81	50.99	48.18	46.91	43.99	39.78	37.50	34.54	31.33	47.28	**+15.57**
SS-NCM† [16]	69.89	61.91	55.51	51.71	49.68	46.11	42.19	39.03	37.96	34.05	32.65	47.33	**+15.52**
SPPR† [63]	68.68	61.85	57.43	52.68	50.19	46.88	44.65	43.07	40.17	39.63	37.33	49.32	**+13.53**
SS-NCM-CNN† [16]	69.89	64.87	59.82	55.14	52.48	49.60	47.87	45.10	40.47	38.10	35.25	50.78	**+12.07**
Decoupled-DeepEMD‡ [58]*	75.35	70.69	66.68	62.34	59.76	56.54	54.61	52.52	50.73	49.20	47.60	58.73	**+4.12**
Decoupled-Cosine‡ [52]*	75.52	70.95	66.46	61.20	60.86	56.88	55.40	53.49	51.94	50.93	49.31	59.36	**+3.49**
ERL† [18]	73.52	70.12	65.12	62.01	58.56	57.99	56.77	56.52	55.01	53.68	50.01	59.93	**+2.92**
ERL++† [18]	73.52	71.09	66.13	63.25	59.49	59.89	58.64	57.72	56.15	54.75	52.28	61.18	**+1.67**
CEC‡ [59]	75.85	71.94	68.50	63.5	62.43	58.27	57.73	55.81	54.83	53.52	52.28	61.33	**+1.52**
FeSSSS (Ours)	**79.60**	**73.46**	**70.32**	**66.38**	**63.97**	**59.63**	**58.19**	**57.56**	**55.01**	**54.31**	**52.98**	**62.85**	

study that using just a linear classifier reduces performance. Training algorithm for lightweight classifier is described in Algorithm 2 in the supplemental.

4.3. Catastrophic Forgetting Mitigation via Gaussian Generator

To mitigate the effect of catastrophic forgetting of old classes, we use a Gaussian Generator model for each class. This starts by calculating the mean vector of each class and retain the centroids between incremental sessions. The mean could be thought of as maintaining a single exemplar per class in a conventional CIL setting. The problem with centroids is that they do not capture anything about the shape of the class, and some classes are much more compact than others leading to a type of catastrophic forgetting as we forget their shapes.

With only a few samples per class developing shape models is a challenge. Inspired from generative models, we developed generating synthetic data for each of the old classes using a Gaussian model that uses the maintained centroid vector and sample variance. We explored with either a scalar (spherical) or vector (axial) model of variance, with a slightly better performance with the scalar – the number of samples (K-shot) is far less than the vector dimensionality likely leading to overfitting. During incremental training, the Gaussian Generator is either learning models for the new classes or generating samples for the known classes by randomly sampling from its Gaussian model. Either way, it provides features to the downstream training.

5. Experiments and Results

We conduct our experiments on three established benchmark data sets for FSCIL, i.e., Caltech-UCSD Birds-200-2011 (CUB200) [53], miniImageNet [50], and CIFAR100 [33]. Below we list the details about data sets and experimental settings and subsequently provide results comparing our approach against state-of-the-art FSCIL methods.

5.1. Data Sets

Caltech-UCSD Birds-200-2011 CUB200 [53] is a fine-grained image classification data set originally comprised of 5994 training and 5794 test images belonging to 200 classes of birds. In [51], authors established a FSCIL protocol where 100 classes were used for the base session and the remaining 100 were equally distributed in 10 incremental sessions. For training, the base session was comprised of 30 samples per class, whereas each incremental session is a 10-way 5-shot setting. It should be noted that all test examples belonging to the enrolled classes at any given incremental session were used for evaluation. In a conventional setting, images were resized to 256 maintaining aspect ratio and 224×224 random crops with random horizontal flip were used during training while central crops of the same size were used for evaluation.

miniImageNet miniImageNet is a small subset of ImageNet-2012 [50] comprised of 100 classes, each having 600 images; 500 training and 100 test images. Tao et al. [51] split the 100 classes into 60 base and 40 incremental classes. The 40 incremental classes were further divided into 8 incremental sessions where each class contained only 5 samples; synthesizing 5-way 5-shot setting. The image size of miniImageNet was 84 × 84.

CIFAR100 CIFAR100 [33] is a popular small scale classification data set comprised of 100 classes. Following [51], 100 classes were divided into 60 base and 40 incremental

Table 2. Comparison of FeSSSS with the state-of-the-art on miniImageNet data set, Moco-v2 [15] trained on OpenImages-v6 has been used as self-supervised feature extractor. ‡ indicates results copied from CEC [59], * identifies the few-shot approaches adapted by [59] for FSCIL, and † shows the results for approaches taken from their respective papers. Further $^\diamond$ identifies that the results have been approximated from graphs since tabular results are unavailable from respective papers. Using a two-sided t-test with each iteration as the data, our approach is statistically significantly better than the state-of-the-art CEC with $p < .0001$

Method	Acc. in each session (%) ↑									Avg. ↑	our relative improvement
	0	1	2	3	4	5	6	7	8		
Ft-CNN‡	61.31	27.22	16.37	6.08	2.54	1.56	1.93	2.6	1.4	13.44	**+54.79**
NCM‡ [29]	61.31	47.8	39.31	31.91	25.68	21.35	18.67	17.24	14.17	30.82	**+37.41**
iCaRL‡ [47]	61.31	46.32	42.94	37.63	30.49	24	20.89	18.8	17.21	33.28	**+34.95**
EEIL‡ [13]	61.31	46.58	44	37.29	33.14	27.12	24.1	21.57	19.58	34.96	**+33.27**
LEC-Net† [57]	61.31	35.37	36.66	38.59	33.90	35.89	36.12	32.97	30.55	37.92	**+30.31**
TOPIC‡ [51]	61.31	50.09	45.17	41.16	37.48	35.52	32.19	29.46	24.42	39.64	**+28.59**
ERL$^{\dagger\diamond}$ [18]	61.67	56.19	54.70	51.19	47.61	45.23	44.0	40.95	39.8	49.03	**+19.20**
ERL++$^{\dagger\diamond}$ [18]	61.67	57.61	54.76	51.67	48.57	46.42	44.04	42.85	40.71	49.81	**+18.42**
SS-NCM-CNN$^{\dagger\diamond}$ [16]	62.88	60.66	57.55	52.66	50.44	48.44	45.11	41.55	40.88	51.13	**+17.10**
Decoupled-DeepEMD‡ [58]*	69.77	64.59	60.21	56.63	53.16	50.13	47.49	45.42	43.41	54.53	**+13.70**
Decoupled-Cosine‡ [52]*	70.37	65.45	61.41	58.00	54.81	51.89	49.10	47.27	45.63	55.99	**+12.24**
CEC‡ [59]	72.00	66.83	62.97	59.43	56.70	53.73	51.19	49.24	47.63	57.74	**+10.49**
SPPR$^{\dagger\diamond}$ [63]	80.0	74.0	68.66	64.33	61.0	57.33	54.66	51.66	49.0	62.29	**+5.94**
FeSSSS (Ours)	**81.5**	**77.04**	**72.92**	**69.56**	**67.27**	**64.34**	**62.07**	**60.55**	**58.87**	**68.23**	

ones; with incremental classes further divided equally in 8 incremental sessions. Like miniImageNet, CIFAR100 was also used in a 5-way 5-shot setting. The data set was comprised of 60000 32 × 32 images with 500 training and 100 test images per class. Due to space constraints, we present results for CIFAR100 only in the supplemental material.

5.2. Experimental Settings

Following existing approaches on FSCIL, for supervised training on base session, we use ResNet-18 for CUB200 and miniImageNet data sets, and ResNet-20 for CIFAR100. Although new models from scratch can be trained on base session data ($\mathcal{D}^0_{train}$), we leverage the advances made by the state-of-the-art approach of CEC [59] and use their trained models. For self-supervised learning, we investigate various models trained either on ImageNet-2012 [50] or OpenImages-v6 [34]. Specifically, we use ResNet-50 models trained in various self-supervised manners including Moco-v2 [15], DeepCluster-v2 [11], SwAV [12], and SeLa-v2 [1]. For main results in Tabs. 1, 2, and 5 (in supplemental) we use DeepCluster-v2 [11] based self-supervised features learned on ImageNet-2012 for CUB200, and CIFAR100 experiments and Moco-v2 [15] based self-supervised features learned on OpenImages-v6 for miniImageNet experiments. This mismatch is imposed to enforce no overlap between the data sets used for supervised and self-supervised models. Results with other self-supervised approaches are demonstrated as part of the ablation study.

To enhance the training data for the classification module, we extract features from both supervised and self-supervised models using various augmented versions of each image in each incremental session. For each FSCIL data set we follow the same data augmentations as originally employed by CEC [59]. We present an ablation analyzing the impact of the number of augmentations.

The classification module is trained for 1000 epochs at a learning rate of 0.1 for the base session. For each incremental session, we use 500 epochs and a smaller learning rate of 0.001. A batch size of 256 is used for both base and incremental training. We further employ class balancing to emphasize the importance of old class centroids or the generated samples from the Gaussian Generator. For each incremental session, we choose the model saved with the last epoch, not the best performing one on the test set. This is due to the fact that there is no held-out validation set, because there are so few samples, and we did not want to tweak the test set that might result in marginal improvement.

For experiments on CUB200, images are resized to 256 maintaining aspect ratio and then 224 × 224 random crops or horizontally flipped random crops are used for training. For miniImageNet, we follow CEC [59] and resize images to 92 and then 84 × 84 random or horizontally flipped random crops are used. During evaluation, for each image only central crop-based features are concatenated and forward passed through the trained classification module.

5.3. Results – Comparison Against SOTA FSCIL

We document our main results on CUB200 and miniImageNet data sets in Tabs. 1 and 2 respectively, while results for CIFAR100 are made available in the supplemental (Tab. 5). We provide a comparison of FeSSSS against the latest state-of-the-art FSCIL approaches [16, 18, 51, 57, 59, 63] and outperform each one of them by a significant margin. To emphasize the relative performance gain, we report the average incremental accuracy in second-to-last column and percentage improvement due to our approach in the last column.

Focusing on CUB200 data set in Tab. 1 , our approach performs better than all types of FSCIL methods, *e.g.*, classic CIL methods adapted for FSCIL (iCaRL, NCM,

Table 3. Ablation conducted on CUB200 demonstrating both supervised and self-supervised features alone are inferior to CEC [59] and feature fusion layer (vs simple linear classifer) result in the best performance. Also the Gaussian Generator's synthetic data for old classes improves performance over just centroid. Either variance model provides statistically better performance ($p < .0001$).

Method	features	feature fusion layer	Gaussian generation	variance	Acc. in each session (%) ↑											Avg. ↑	improvement over CEC
					0	1	2	3	4	5	6	7	8	9	10		
CEC	-	-	-	-	75.85	71.94	68.50	63.5	62.43	58.27	57.73	55.81	54.83	53.52	52.28	61.33	-
FeSSSS (Ours)	concat	✗	✓	scalar	77.02	68.18	63.90	59.46	55.51	51.89	49.24	46.75	44.92	43.53	42.06	54.76	**-6.57**
	self-supervised	✓	✗	n.a.	73.21	65.09	62.24	58.26	54.76	51.82	49.78	47.60	44.54	44.46	43.56	54.12	**-7.21**
	supervised	✓	✗	n.a.	74.37	67.86	64.54	60.81	58.19	54.62	53.22	51.61	49.55	48.63	46.63	57.27	**-4.06**
	concat	✓	✗	n.a.	79.60	72.19	69.47	65.63	63.55	58.78	58.01	56.64	54.09	53.65	52.81	62.22	**+0.89**
	concat	✓	✓	vector	79.60	72.70	70.02	66.33	63.87	59.40	58.19	57.09	54.78	54.62	52.91	62.68	**+1.35**
	concat	✓	✓	scalar	79.60	73.46	70.32	66.38	63.97	59.63	58.19	57.56	55.01	54.31	52.98	62.85	**+1.52**

EEIL), methods leveraging from additional data and semi-supervised training (SS-iCaRL, SS-NCM, SS-NCM-CNN), few-shot classification approaches adopted for FSCIL learning (Decoupled-Cosine, Decoupled-DeepEMD), and methods specifically designed for FSCIL (TOPIC, CEC, ERL, ERL++, SPPR, LEC-Net). It is interesting to note that general CIL methods adapted for FSCIL perform very poorly. While the performance of specifically designed solutions for FSCIL is better, some of them (LEC-Net, SPPR) barely improve over baseline, *i.e.*, TOPIC [51]. The semi-supervised approach by [16] that leverages additional unlabeled data is capable of improving the baseline iCaRL and NCM but not in the top performers. Compared to these specifically designed FSCIL approaches (SPPR, LEC-Net) or adapted CIL approaches (iCaRL, NCM, EEIL), few-shot methods (Decoupled-Cosine, Decoupled-DeepEMD) adapted for FSCIL perform much better. The closest approaches were ERL/ERL++ and CEC which were consistently outperformed by our method across all incremental sessions. We should note that the performance gain due to our method is higher for initial incremental sessions, which is understandable as catastrophic forgetting becomes worse over a period of time as more and more classes are enrolled.

When we focus on results for miniImageNet documented in Tab. 2 we face some interesting contradictions. For example CEC is no longer the second best-performing approach and ERL/ERL++ performed poorly. However, it is worth noting that CEC and few-shot methods (Decoupled-Cosine, Decoupled-DeepEMD) are still in the lead. The contradiction may be due to the bigger base session in miniImageNet than CUB200 and introduction of fewer classes per incremental session. More specifically, the number of training images in the base session of miniImageNet is 30000, these images belong to 60 classes, whereas for CUB200 there are only 3000 images in base session belonging to 100 classes. Additionally, in miniImageNet only five classes are introduced in each incremental session whereas in CUB200 ten classes are introduced, possibly triggering more over-fitting of new classes. Nonetheless, as evidenced in Tab. 2, our approach consistently outperforms all other approaches by a high margin as noted in the last column as percentage improvement in average class incremental accuracy.

5.4. Ablation on Importance of Components

In Tab. 3 we report the results of an ablation study to analyze the importance of our claimed contributions including supervised and self-supervised representations and the Gaussian Generator. The lightweight classification module trained independently on either of the two feature representations performs poorly compared to the CEC approach. However, when the feature representations are combined by mere concatenation, the classification head outperforms CEC consistently across all incremental sessions. As mentioned earlier, to mitigate catastrophic forgetting we maintain a centroid (mean) vector per class which were also used during training to refresh the classification module on the old classes. To further reduce catastrophic forgetting we explored synthetic data generation for which a variance vector or scalar per class needs to be maintained. Tab. 3 shows that by synthesizing data for old classes and including it in an incremental session training, we were able to squeeze a little more performance gain. It is worth noting that there is not a significant difference whether a variance vector per class or a single scalar is maintained. Since the underlying data in each session is very limited (five images per class), generating a full co-variance matrix is pointless and does not align with the limited resource setting of FSCIL. Tab. 3 also demonstrates the importance of feature fusion layer as by removing it, the performance is significantly dropped.

5.5. Supervised & Self-Supervised Data Overlap

To avoid overlap between self-supervised and supervised data sets, we have used self-supervised models trained on ImageNet-2012 and OpenImages-v6 data sets respectively for experiments on CUB200/CIFAR100 and miniImageNet. Since, miniImageNet is comprised of selected classes from the original ImageNet-2012, using self-supervised features based on ImageNet would be biased. A good separation for miniImagenet classes would have been learned already in a self-supervised manner. We conduct an ablation in Tab. 7 (in supplemental) where self-supervised models trained on ImageNet are used as a feature extractor demonstrating that using ImageNet-2012-based features results in much better performance than reported in Tab. 2. We should further emphasize that in Tab. 7, classification heads trained on self-supervised features alone outperform the concatenated

Table 4. Ablation conducted on CUB200 for number of augmentations per training image. For this ablation, we used self-supervised features learned through DeepCluster-v2 [11]. It is evident as the number of augmentations per image increase the performance increases.

Method	number of augmentation per image	features	feature fusion layer	Gaussian generation	variance	Acc. in each session (%) ↑											Avg. ↑	improvement over CEC
						0	1	2	3	4	5	6	7	8	9	10		
CEC	-	-	-	-	-	75.85	71.94	68.50	63.5	62.43	58.27	57.73	55.81	54.83	53.52	52.28	61.33	-
FeSSSS (Ours)	10	concat	✓	✗	n.a.	77.72	70.44	67.49	63.57	60.84	56.70	56.15	54.59	52.59	52.00	50.72	60.25	**-1.08**
	20	concat	✓	✗	n.a.	78.45	71.04	68.07	64.86	62.31	58.04	57.10	55.60	52.86	52.71	51.58	61.14	**-0.19**
	30	concat	✓	✗	n.a.	79.01	71.33	68.45	64.86	62.53	58.57	57.17	55.83	53.11	53.07	52.08	61.45	**+0.12**
	40	concat	✓	✗	n.a.	79.32	71.61	68.83	65.42	62.85	58.73	57.45	56.36	53.72	53.34	52.26	61.80	**+0.47**
	50	concat	✓	✗	n.a.	79.29	72.44	69.09	65.63	63.18	59.01	57.69	56.38	53.93	53.45	52.72	62.07	**+0.74**
	60	concat	✓	✗	n.a.	79.60	72.19	69.47	65.63	63.55	58.78	58.01	56.64	54.09	53.65	52.81	62.22	**+0.89**

features.

5.6. Number of Augmentations Per Image

The classification module in our FeSSSS approach is not fed images directly, rather image features are used as input. To increase the variation of training data, we generated various augmented versions of training images. Specifically, we used the random crop and horizontal flip augmentations conventionally employed for the training of deep models. The augmented versions of training images are passed through both supervised and self-supervised feature extractors to generate the respective representations. Tab. 4 shows the results of the ablation study where the number of augmentations per training image are varied. It is evident that as the number of augmentations being used increases, the performance improves. We have chosen 60 augmentations per training image as a good compromise as the performance gain due to increased number of augmentations slowly declines and the training time of the classification module in each incremental session consequently increases.

5.7. Various Self-Supervised Features

In Tabs. 6, 7, and 8 (in supplementary) we have conducted an ablation study on various self-supervised features including DeepCluster-v2 [11], SwAV [12], Moco-v2 [15], and SeLa-v2 [1] for CUB200, miniImageNet, and CIFAR100 data sets. In each case the self-supervised features are learned on the ImageNet-2012 data set [50] with a ResNet-50 model. We can note that not all self-supervised features perform equally well for FSCIL, *e.g.*, while DeepCluster-v2 and SwAV features combined with supervised features outperform CEC, Moco-v2 and SeLa-v2 have lower performance. Irrespective of the underlying self-supervised representation, synthetic data generated with the Gaussian Generator is always helpful with either using scalar or vector variance compared to only maintaining the centroid vectors.

6. Discussion

Few-shot class incremental learning is a challenging CIL setting which has inherently limited number of labeled data for each session. Inspired from advances of self-supervised learning and its demonstrated applicability to novel downstream tasks, we have investigated its use for FSCIL. Through our main results in Tabs. 1 and 2, we have demonstrated that our proposed approach FeSSSS is capable of outperforming specially designed pure-supervised approaches for FSCIL. It is interesting to note that general CIL approaches do not scale well for FSCIL setting, whereas, pure few-shot approaches adapted for FSCIL are capable to perform comparatively. Similarly, by using extra data in a semi-supervised fashion, [16] are able to squeeze some performance gain from classical methods, however they performed rather poorly compared to other new methods and much worse than our FeSSSS. We should emphasize that after initial learning on session zero, we kept the supervised representation fixed throughout the incremental steps. A natural future work would be to let the supervised representation adapt with data from incremental sessions, *i.e.*, same as original CEC, in addition to adapting our classification module.

7. Conclusion

We present FeSSSS – a learning framework for the challenging problem of *few-shot class incremental learning* where we explore the importance of self-supervised learning. We demonstrate that using either supervised or self-supervised features independently is sub-optimal and results in performance lower than CEC/SPPR, *i.e.*, the previous state-of-the-art algorithms on FSCIL. By employing feature fusion, FeSSSS is capable of outperforming all existing methods for FSCIL on three established benchmarks by a large margin. We further show that the Gaussian Generator further addresses catastrophic forgetting of old classes and helps to further boost the performance of the proposed approach. An ablation on various self-supervised techniques demonstrates how some approaches (DeepCluster-v2, SwAV) are better suited for the task of FSCIL than others (Moco-v2, SeLa-v2). To further avoid the overlap between supervised and self-supervised feature extractors, we used Moco-v2 trained on OpenImages for miniImageNet experiments.

Acknowledgement

This research is supported by the Defense Advanced Research Projects Agency (DARPA), specifically SAIL-ON program under contract number HR001120C0055.

References

[1] Yuki Markus Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. In *Int. Conf. Learn. Represent.*, 2020. 6, 8, 12, 14, 15

[2] Eden Belouadah and Adrian Popescu. Deesil: Deep-shallow incremental learning. In *Eur. Conf. Comput. Vis. Work.*, 2018. 2, 3

[3] Eden Belouadah and Adrian Popescu. Il2m: Class incremental learning with dual memory. In *Int. Conf. Comput. Vis.*, 2019. 3

[4] Eden Belouadah and Adrian Popescu. Scail: Classifier weights scaling for class incremental learning. In *IEEE Win. Conf. App. Comput. Vis.*, 2020. 3

[5] Eden Belouadah, Adrian Popescu, and Ioannis Kanellos. A comprehensive study of class incremental learning algorithms for visual tasks. *ArXiv*, abs/2011.01844, 2020. 3

[6] Abhijit Bendale and Terrance Boult. Towards open world recognition. In *Computer Vision and Pattern Recognition*, 2015. 1

[7] Abhijit Bendale and Terrance E. Boult. Towards Open Set Deep Networks. In *Computer Vision and Pattern Recognition*, 2016. 1

[8] Mateusz Buda, Atsuto Maki, and Maciej A. Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249 – 259, 2018. 3

[9] Kaidi Cao, Maria Brbic, and Jure Leskovec. Open-World Semi-Supervised Learning. In *arXiv:2102.03526v2*, 2021. 2, 3

[10] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep Clustering for Unsupervised Learning of Visual Features. In *European Conference on Computer Vision*, 2018. 2, 3

[11] Mathilde Caron, Piotr Bojanowski, Julien Mairal, and Armand Joulin. Unsupervised Pre-Training of Image Features on Non-Curated Data. In *International Conference on Computer Vision*, 2019. 2, 3, 5, 6, 8, 12, 14, 15

[12] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *Adv. Neural Inform. Process. Syst.*, 2020. 6, 8, 12, 14, 15

[13] Francisco M. Castro, Manuel J. Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-End Incremental Learning. In *European Conference on Computer Vision*, 2018. 1, 3, 5, 6, 14

[14] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A Simple Framework for Contrastive Learning of Visual Representations. In *International Conference on Machine Learning*, 2020. 2, 3

[15] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *ArXiv*, 2010.15277, 2020. 6, 8, 12, 14, 15

[16] Yawen Cui, Wuti Xiong, Mohammad Tavakolian, and Li Liu. Semi-Supervised Few-Shot Class-Incremental Learning. In *International Conference on Image Processing*, 2021. 3, 5, 6, 7, 8, 14

[17] Akshay Raj Dhamija, Touqeer Ahmad, Jonathan Schwan, Mohsen Jafarzadeh, Chunchun Li, and Terrance E. Boult. Self-Supervised Features Improve Open-World Learning. In *arXiv:2102.07848v2*, 2021. 1, 2, 3

[18] Songlin Dong, Xiaopeng Hong, Xiaoyu Tao, Xinyuan Chang, Xing Wei, and Yihong Gong. Few-Shot Class-Incremental Learning via Relation Knowledge Distillation. In *AAAI Conference on Artificial Intelligence*, 2021. 3, 5, 6, 14

[19] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *Eur. Conf. Comput. Vis.*, 2020. 1, 2, 3

[20] Jhair Gallardo, Tyler L. Hayes, and Christopher Kanan. Self-Supervised Training Enhances Online Continual Learning. In *arXiv:2103.14010v2*, 2021. 2, 3

[21] Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. SCAN: Learning to Classify Images without Labels. In *European Conference on Computer Vision*, 2020. 2, 3

[22] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised Representation Learning by Predicting Image Rotations. In *International Conference on Learning Representations*, 2018. 2, 3

[23] Priya Goyal, Dhruv Mahajan, Abhinav Gupta, and Ishan Misra. Scaling and Benchmarking Self-Supervised Visual Representation Learning. In *International Conference on Computer Vision*, 2019. 2, 3

[24] Kai Han, Sylvestre-Alvise Rebuffi, Sebastien Ehrhardt, Andrea Vedaldi, and Andrew Zisserman. Automatically Discovering and Learning New Visual Categories with Ranking Statistics. In *International Conference on Learning Representations*, 2020. 2, 3

[25] Tyler L. Hayes, Kushal Kafle, Robik Shrestha, Manoj Acharya, and Christopher Kanan. Remind your neural network to prevent catastrophic forgetting. In *Eur. Conf. Comput. Vis.*, 2020. 3

[26] Tyler L. Hayes and Christopher Kanan. Lifelong machine learning with deep streaming linear discriminant analysis. In *IEEE Conf. Comput. Vis. Pattern Recog. Worksh.*, 2020. 2, 3

[27] Jiangpeng He, Runyu Mao, Zeman Shao, and Fengqing Zhu. Incremental learning in online scenario. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020. 2, 3

[28] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum Contrast for Unsupervised Visual Representation Learning. In *Computer Vision and Pattern Recognition*, 2020. 2, 3

[29] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a Unified Classifier Incrementally via Rebalancing. In *Computer Vision and Pattern Recognition*, 2019. 1, 3, 5, 6, 14

[30] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019. 3

[31] Ahmet Iscen, Jeffrey Zhang, Svetlana Lazebnik, and Cordelia Schmid. Memory-efficient incremental learning through feature adaptation. In *Eur. Conf. Comput. Vis.*, 2020. 3

[32] Simon Jenni, Hailin Jin, and Paolo Favaro. Steering Self-Supervised Feature Learning Beyond Local Pixel Statistics. In *Computer Vision and Pattern Recognition*, 2020. 2, 3

[33] Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images. Technical report, University of Toronto, 2009. 5

[34] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, Tom Duerig, and Vittorio Ferrari. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *Int. J. Comput. Vis.*, 128:1956–1981, 2020. 2, 6

[35] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2021. 1

[36] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(12):2935–2947, 2018. 2, 3

[37] Yaoyao Liu, Yuting Su, An-An Liu, Bernt Schiele, and Qianru Sun. Mnemonics training: Multi-class incremental learning without forgetting. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020. 2, 3

[38] Yong Luo, Liancheng Yin, Wenchao Bai, and Keming Mao. An appraisal of incremental learning methods. *Entropy*, 22(11), 2020. 3

[39] Marc Masana, Xialei Liu, Bartlomiej Twardowski, Mikel Menta, Andrew D. Bagdanov, and Joost van de Weijer. Class-incremental learning: survey and performance evaluation. *ArXiv*, 2010.15277, 2020. 3

[40] Thomas Mensink, Jakob Verbeek, Florent Perronnin, and Gabriela Csurka. Metric learning for large scale image classification: Generalizing to new classes at near-zero cost. In *Eur. Conf. Comput. Vis.*, 2012. 2

[41] Thomas Mensink, Jakob Verbeek, Florent Perronnin, and Gabriela Csurka. Distance-based image classification: Generalizing to new classes at near-zero cost. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(11):2624–2637, 2013. 2

[42] Fei Mi, Lingjing Kong, Tao Lin, Kaicheng Yu, and Boi Faltings. Generalized class incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 240–241, 2020. 1, 3

[43] Ishan Misra and Laurens van der Maaten. Self-Supervised Learning of Pretext-Invariant Representations. In *Computer Vision and Pattern Recognition*, 2020. 2, 3

[44] Mehdi Noroozi and Paolo Favaro. Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles. In *European Conference on Computer Vision*, 2016. 2, 3

[45] Mehdi Noroozi, Hamed Pirsiavash, and Paolo Favaro. Representation Learning by Learning to Count. In *International Conference on Computer Vision*, 2017. 2, 3

[46] Jathushan Rajasegaran, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Mubarak Shah. itaml: An incremental task-agnostic meta-learning approach. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020. 1, 3

[47] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. iCaRL: Incremental Classifier and Representation Learning. In *Computer Vision and Pattern Recognition*, 2017. 1, 3, 5, 6, 14

[48] Sylverstre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. icarl: Incremental classifiers and representation learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017. 2, 3

[49] Ethan M Rudd, Lalit P Jain, Walter J Scheirer, and Terrance E Boult. The extreme value machine. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(3):762–768, 2017. 1

[50] 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. *Int. J. Comput. Vis.*, 115(3):211–252, 2015. 2, 3, 5, 6, 8, 12

[51] Xiaoyu Tao, Xiaopeng Hong, Xinyuan Chang, Songlin Dong, Xing Wei, and Yihong Gong. Few-Shot Class-Incremental Learning. In *Computer Vision and Pattern Recognition*, 2020. 1, 3, 5, 6, 7, 14

[52] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching Networks for One Shot Learning. In *Neural Information Processing Systems*, 2016. 3, 5, 6, 14

[53] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 5

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

[55] Max Welling. Herding dynamical weights to learn. In *Int. Conf. on Mach. Learning*, 2009. 2, 3

[56] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019. 2, 3

[57] Boyu Yang, Mingbao Lin, Binghao Liu, Mengying Fu, Chang Liu, Rongrong Ji, and Qixiang Ye. Learnable Expansion-and-Compression Network for Few-shot Class-Incremental Learning. In *arXiv:2104.02281*, 2021. 5, 6, 14

[58] Chi Zhang, Yujun Cai, Guosheng Lin, and Chunhua Shen. DeepEMD: Differentiable Earth Mover's Distance for Few-Shot Learning. In *Computer Vision and Pattern Recognition*, 2020. 1, 3, 5, 6, 14

[59] Chi Zhang, Nan Song, Guosheng Lin, Yun Zheng, Pan Pan, and Yinghui Xu. Few-Shot Incremental Learning with Continually Evolved Classifiers. In *Computer Vision and Pattern Recognition*, 2021. 3, 5, 6, 7, 14

[60] Richard Zhang, Phillip Isola, and Alexei A. Efros. Colorful Image Colorization. In *European Conference on Computer Vision*, 2016. 2, 3

[61] Richard Zhang, Phillip Isola, and Alexei A. Efros. Split-Brain Autoencoders: Unsupervised Learning by Cross-Channel Prediction. In *Computer Vision and Pattern Recognition*, 2017. 2, 3

[62] Bowen Zhao, Xi Xiao, Guojun Gan, Bin Zhang, and Shutao Xia. Maintaining discrimination and fairness in class incremental learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020. 3

[63] Kai Zhu, Yang Cao, Wei Zhai, Jie Cheng, and Zheng-Jun Zha. Self-Promoted Prototype Refinement for Few-Shot Class-Incremental Learning. In *Computer Vision and Pattern Recognition*, 2021. 1, 3, 5, 6, 12, 14