Calibrating Higher-Order Statistics for Few-Shot Class-Incremental Learning with Pre-trained Vision Transformers

Dipam Goswami[1,2] Bartłomiej Twardowski[1,2,3] Joost van de Weijer[1,2]
[1]Department of Computer Science, Universitat Autònoma de Barcelona
[2]Computer Vision Center, Barcelona [3]IDEAS-NCBR
{dgoswami, btwardowski, joost}@cvc.uab.es

Abstract

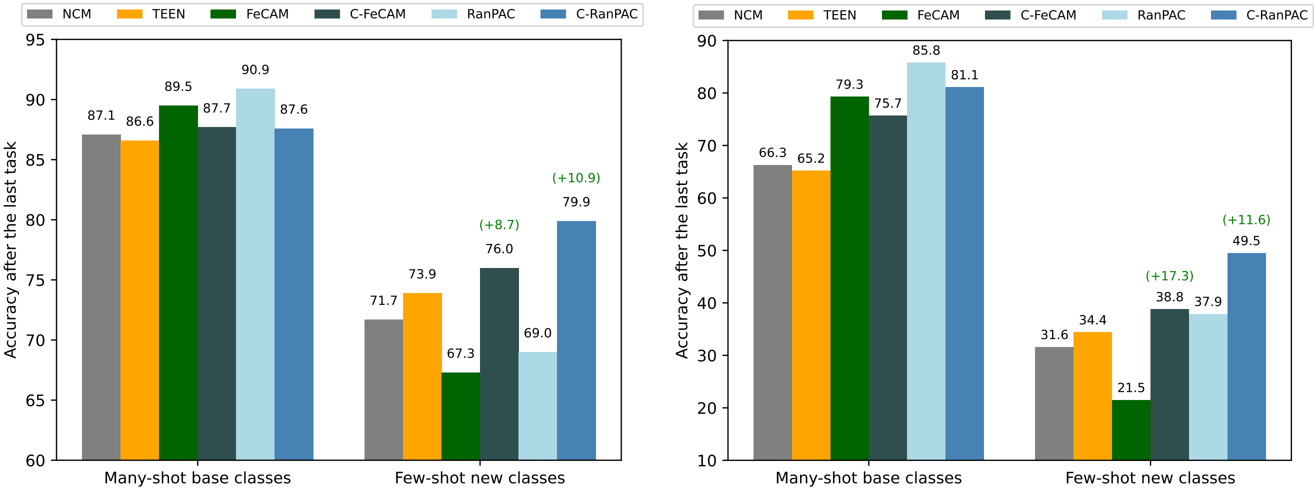
Few-shot class-incremental learning (FSCIL) aims to adapt the model to new classes from very few data (5 samples) without forgetting the previously learned classes. Recent works in many-shot CIL (MSCIL) (using all available training data) exploited pre-trained models to reduce forgetting and achieve better plasticity. In a similar fashion, we use ViT models pre-trained on large-scale datasets for few-shot settings, which face the critical issue of low plasticity. FSCIL methods start with a many-shot first task to learn a very good feature extractor and then move to the few-shot setting from the second task onwards. While the focus of most recent studies is on how to learn the many-shot first task so that the model generalizes to all future few-shot tasks, we explore in this work how to better model the few-shot data using pre-trained models, irrespective of how the first task is trained. Inspired by recent works in MSCIL, we explore how using higher-order feature statistics can influence the classification of few-shot classes. We identify the main challenge of obtaining a good covariance matrix from few-shot data and propose to calibrate the covariance matrix for new classes based on semantic similarity to the many-shot base classes. Using the calibrated feature statistics in combination with existing methods significantly improves few-shot continual classification on several FSCIL benchmarks. Code is available at https://github.com/dipamgoswami/FSCIL-Calibration.

1. Introduction

Continual Learning (CL) aims to learn from data in a continuous manner where the data distribution changes over time and the model is expected to not forget the old classes learned in previous tasks, a phenomenon known as catastrophic forgetting [19, 30, 37]. Class-incremental learning (CIL) refers to learning new classes incrementally over time with the goal of classifying all classes seen so far without any task information [6, 29, 46, 49]. While many recent studies focus on many-shot CIL (MSCIL) [11, 31, 34, 61] assuming the availability of sufficient training data for all classes, a more challenging setting is few-shot CIL (FSCIL), which considers very few training samples for each class [1, 33, 43, 45, 50, 54, 57, 58]. In this work, we address the FSCIL problem.

Existing FSCIL methods consider a many-shot first task where a good feature extractor is trained. After the first task, all subsequent tasks are few-shot. The standard practice is to freeze the backbone after the first task and then compute the class-wise prototypes [47] by averaging the feature embeddings and classifying using the nearest class mean (NCM) classifier [35]. While most FSCIL [7, 33, 40, 57, 58] methods are dependent on the base task learning and focus on how to effectively train the many-shot first task to learn more compact representations of base classes in order to generalize better in future few-shot steps, fewer methods [11, 50] propose how to better model the few-shot data in new tasks independent of the first task training. Recently, pre-trained ViTS have been extensively studied for many-shot CIL [11, 31, 55, 59, 60]. Unlike the conventional FSCIL settings using ResNet architectures [13] (without pre-training on large-scale datasets), we study how the ViTs [9] pre-trained on large-scale datasets like ImageNet21k [36] can be exploited for FSCIL.

With the availability of pre-trained weights, a more realistic approach to solving FSCIL would be to focus on how to better use the few-shot data, irrespective of how the base task is adapted. We propose to adapt the pre-trained ViT model to the base classes in the first task with an adaptor [4], similar to [59]. After adapting the model in the first task, we study different ways of classifying the test features in the few-shot steps by exploring recent concepts from the many-shot CIL domain which works on a frozen feature space [11, 31]. Recently, FeCAM [11] explored the heterogeneous nature of feature distributions in CL settings and proposed to use the anisotropic Mahalanobis distance in the feature space for classification. Another recent work



(a) CUB-200 (100 classes in base task, 100 classes in 10 new tasks)

(b) Stanford Cars (98 classes in base task, 98 classes in 7 new tasks)

Figure 1. Performance of different prototype-based classification methods on FSCIL settings with ViT-B/16 pre-trained on ImageNet-21k. All the methods - NCM, TEEN [50], FeCAM [11] and RanPAC [31] are biased towards the base task classes. While TEEN improves the performance on the few-shot classes by prototype calibration, methods using second-order feature statistics - FeCAM and RanPAC performs much poorly on the few-shot classes compared to the many-shot base task classes. This drop in performance for new classes can be attributed to the poor estimates of second-order statistics from few-shot data. We propose to calibrate the covariance matrix of few-shot classes by using the strong covariance estimates of base classes. We observe that on using our proposed calibration, C-FeCAM and C-RanPAC improve performance significantly on the new classes, leading to an overall better accuracy.

- RanPAC [31] proposed random projections to a very high-dimensional feature space and classified features using a shared Gram matrix instead of the covariance matrix. We use both FeCAM and RanPAC after adapting the pre-trained ViT to the base task and observe poor performance in the few-shot classes (see Fig. 1), suggesting that few-shot class statistics are poorly calibrated.

A recent work in FSCIL, TEEN [50] proposed to calibrate the prototypes for the new classes based on semantic similarity to the base classes, which resulted in better NCM classification of the new classes. In a similar fashion, we hypothesize that the covariances of new classes can be calibrated based on the covariances of many-shot base classes. This has some evidence in previous works [34, 38, 52]. We propose to calibrate the covariance matrices of new classes based on the similarity to the base classes. Finally, we use the calibrated statistics with FeCAM and RanPAC and achieve significant improvement over the baseline methods.

We perform experiments on CIFAR100 [23] and fine-grained classification datasets like CUB-200 [48], Stanford Cars [22], and FGVC-Aircraft [28], which are very good use cases of few-shot data. We study the commonly used settings with a big first task and also introduce challenging settings with a small first task (equally splitting the dataset into tasks). We demonstrate that calibrated higher-order statistics enable better classification of few-shot classes, which is reflected in the harmonic mean accuracy. The main contributions can be summarized as:

1. We explore how knowledge from pre-trained ViTs can be transferred to new few-shot classes in FSCIL settings instead of conventional approaches to training ResNets from scratch.
2. We identify that higher-order statistics-based classification approaches perform poorly in classifying few-shot classes due to poor estimates of statistics from very few samples.
3. We propose feature covariance calibration for few-shot classes, exploiting strong covariance estimates of many-shot base classes, thus enabling better classification of few-shot classes using recent state-of-the-art MSCIL methods like FeCAM and RanPAC.

2. Related Work

Class-Incremental Learning. Class-Incremental Learning [6, 29, 46, 49] learns new classes in incremental tasks and aims to preserve the knowledge of old classes without access to task-ID at inference. While many methods [3, 8, 10, 15, 27, 35] store samples/exemplars from old classes in memory, recent methods [11, 31, 34, 61] propose exemplar-free solutions to CIL. The main challenge of exemplar-free CIL is to prevent forgetting of old classes since it is difficult to distinguish classes from different tasks [41]. CIL methods typically focus on either regularization approaches [10, 21, 24, 26]. Several methods [5, 11, 12, 16, 31, 32] show that using prototypes and higher-order statistics for classification approaches is very efficient. In this work, we explore how these state-of-the-art MSCIL methods perform in FSCIL settings.

Few-Shot Class-Incremental Learning. A common practice in existing FSCIL methods [7, 14, 33, 40, 45, 57, 58] is to use a simple NCM classifier based on the class-wise prototypes on the frozen model (where the model is only trained on the first task). FACT [57] reserves space in the embedding space during the base task training for future classes by allocating virtual prototypes. To enable better model generalization in future few-shot tasks, during the base task training SAVC [40] proposes using semantic-aware fantasy classes, ALICE [33] uses angular penalty loss, LIMIT [58] uses fake incremental tasks and EHS [7] uses expanding the hyperspherical space. Some methods update the network in the few-shot tasks by learning a neural gas network [43], learning soft masks [18], by weight space rotation process [20], by self-supervised stochastic classifiers [17] or by training a two-branch network with class-aware bilateral distillation [56]. Different from these methods, inspired by neural collapse, NC-FSCIL [53] fixes the prototype positions in a simplex equiangular tight frame and trains a model to map the features to their corresponding prototypes. While all these works use resnet architectures, pre-trained ViTs have been recently used [44] for FSCIL.

Recent methods [2, 50] propose to calibrate new-class prototypes using semantic information of base classes. Akyürek et al. [2] propose a semantic subspace regularization-based objective to calibrate new class prototypes. Wang et al. [50] observe that while the base class prototypes are well-calibrated due to abundant training samples in the many-shot first task, the new class prototypes are biased, resulting in high false-positive classifications of new classes to their most similar base classes. Based on this observation, they propose to simply calibrate the new class prototypes by simply fusing them with the base class prototypes, which are weighted based on their similarity. Motivated by these prototype calibration methods, we propose to extend calibration to include higher-order statistics of few-shot classes.

3. Method

The main objective of FSCIL methods is to improve the classification of few-shot classes while not forgetting the base classes. The most commonly studied classification method for FSCIL is to perform NCM and assign the test sample to the closest prototype mean in the feature space using euclidean distance. We observe that while calibrated prototypes using TEEN [50] improve performance marginally compared to naive NCM, the higher-order statistics (covariance matrix or Gram matrix) for new classes are still poorly estimated from a few samples and thus affect classification with methods like FeCAM and RanPAC. We show in Fig. 1 that directly applying these methods to FSCIL settings results in poor performance on few-shot classes. The main contribution of our work is to show how to get better estimates of the feature distributions of the few-shot classes by calibrating their covariances based on the base class covariances.

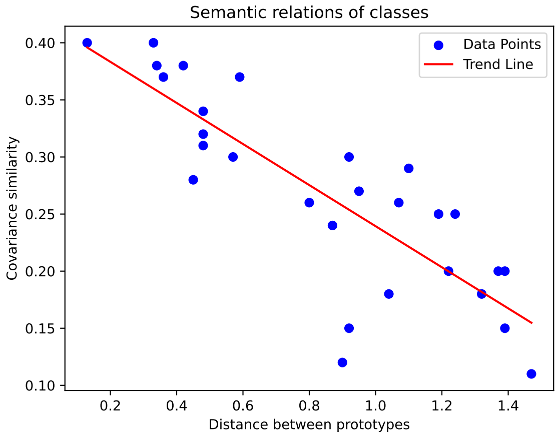


Figure 2. Illustration to demonstrate how the similarity of the covariance matrices of classes vary based on the distance between the class prototypes. We train the model on 28 base classes on the Stanford Cars dataset, and plot the covariance similarity with respect to the prototype distance of a new class (from task 1) with the base classes. We observe that the classes with similar prototypes (lesser distance between the prototypes) have higher covariance similarities.

We explore how these various methods work with few-shot data when using ViT models pre-trained on large-scale datasets like ImageNet-21k. While pre-trained models have been found to provide very good classification accuracy without any training on the downstream datasets [11, 16], it is important to adapt the model to the downstream datasets [31, 59]. Conventionally, FSCIL methods are evaluated with ResNet models (without large-scale pretraining). However, it is more intuitive to use pre-trained knowledge in FSCIL settings due to the availability of very few training samples. In this work, we adapt the pre-trained ViT model on the dataset in the many-shot base task by optimizing the adaptor, similar to [4] and then use the prototype-based methods for classification in the new tasks without any further adaptation.

3.1. Motivation

Prototype calibration in TEEN [50] aims to exploit the semantic relationships between the base classes on which the model is adapted and the new classes, which have very few samples. The relationships between the means and variances of different classes have been explored in few-shot learning [38, 52]. In few-shot learning [51], class distributions have been calibrated in different forms by simply averaging from few most similar base classes while discarding the current class covariance matrices [52] or by iteratively learning the distribution calibration [25]. Following the recent success of using higher-order feature statistics for classification [11, 31], we explore how the covariance matrix of each new class can be calibrated based on their semantic relations to all base classes.

We analyze in Fig. 2 how the covariance matrices vary for classes with varying prototype similarities. We observe that the covariances are similar for classes with higher cosine similarity between their prototypes. Thus, we propose to exploit these semantic relationships to calibrate the covariance matrices of new few-shot classes using the strong covariance estimates of many-shot base classes, weighted based on their prototype similarities.

3.2. Statistics Calibration

The similarity between a pair of base class prototype μ_b and a new class prototype μ_n can be used to compute the weights for averaging all the old class statistics with new ones. We obtain the cosine similarity $S_{b,n}$ between μ_b and μ_n as follows:

$$S_{b,n} = \frac{\mu_b \cdot \mu_n}{\|\mu_b\| \cdot \|\mu_n\|} \cdot \tau \tag{1}$$

where τ controls the sharpness of the weight's distribution. Following [50], we consider $\tau = 16$ in our experiments.

The weightage of a new class prototype μ_n corresponding to a base class prototype μ_b can be obtained by performing softmax over all the base class prototypes as follows:

$$w_{b,n} = \frac{e^{S_{b,n}}}{\sum_{i=1}^{B} e^{S_{i,n}}} \tag{2}$$

such that $\sum_{b=1}^{B} w_{b,n} = 1$ for a new class n, where B is the number of base classes.

Prototype Calibration. Similar to [50], the biased prototype means of new classes μ_n can be calibrated as follows:

$$\hat{\mu}_n = \alpha\, \mu_n + (1-\alpha) \sum_{b=1}^{B} w_{b,n}\mu_b \tag{3}$$

where α controls the degree of calibration.

Covariance Calibration. Similar to Eq. (3), we propose to use the softmaxed similarity weights from Eq. (2) to calibrate the new class covariances Σ_n by averaging with the base class covariances Σ_b as follows:

$$\hat{\Sigma}_n = \beta\,(\Sigma_n + \sum_{b=1}^{B} w_{b,n}\Sigma_b) \tag{4}$$

where β controls the scaling of the covariance matrix.

3.3. Calibration with existing methods

3.3.1 FeCAM

Goswami et al. [11] propose to use a Mahalanobis distance-based classifier to classify feature embeddings at test time. They demonstrate that the anisotropic Mahalanobis distance is more effective than the commonly used Euclidean distance in CL settings for classification in the embedding space. To compute the Mahalanobis distance, FeCAM uses the covariance matrix of the feature embeddings for all classes. In the few-shot setting, the covariance matrix has to be obtained using only 5 samples per class resulting in very poor estimates. To get better estimates of the few-shot class distributions, we propose to use FeCAM with calibrated statistics.

Calibrated FeCAM. In order to complement FeCAM for FSCIL, we use calibrated prototypes and covariances for each new class. Similar to [11, 31], we perform covariance shrinkage to obtain an invertible full-rank covariance matrix. To account for the shift in feature distributions between base classes on which the backbone is trained and new classes that are not used for training, we follow [11] and obtain the correlation matrix from the shrunk covariance matrix by performing correlation normalization.

We compute the Mahalanobis distance between the prototypes and the test features by using the correlation matrix of each class as follows:

$$\mathcal{D}_M(f(x), \hat{\mu}_y) = (f(x) - \hat{\mu}_y)^T (N(\hat{\Sigma}_y + \gamma I))^{-1} (f(x) - \hat{\mu}_y) \tag{5}$$

where $\hat{\mu}_y$ refers to the calibrated prototypes, $\hat{\Sigma}_y$ refers to the calibrated covariances obtained using Eq. (4), $f(x)$ refers to the features obtained using the test samples, and N denotes the correlation normalization. For the base classes, we apply $\hat{\Sigma}_y = \Sigma_y$.

The features are then classified based on the Mahalanobis distance as follows:

$$y^* = \underset{y=1,\dots,Y}{\operatorname{argmin}}\, \mathcal{D}_M(f(x), \hat{\mu}_y) \tag{6}$$

3.3.2 RanPAC

McDonnell et al. [31] randomly project the features using non-linear activations to a very high dimensional space and perform classification in high dimensions where the linear separability of features are better. They advocated using the Gram matrix of features for prototype-based classification due to reduced off-diagonal correlations among the class prototypes, which enables better separability of classes. RanPAC proposed to compute the probability scores as follows:

$$S_y = \phi(f(x)W)(G + \lambda I)^{-1} c_y. \tag{7}$$

		Small First Task (10 tasks)							Big First Task (11 tasks)						
				A_{HM}			A_{last}	A_{inc}			A_{HM}			A_{last}	A_{inc}
	Task	1	3	5	7	9			2	4	6	8	10		
CUB200	NCM	88.48	80.95	77.75	72.19	66.23	70.25	79.68	75.9	84.21	82.16	78.35	71.33	79.43	84.24
	TEEN [50]	89.77	81.33	78.57	71.76	67.25	70.52	80.04	78.01	85.34	82.89	79.44	73.06	80.24	84.83
	FeCAM [11]	91.14	80.33	76.59	70.55	65.24	70.25	79.93	71.05	82.0	77.65	74.36	64.42	79.43	84.03
	C-FeCAM	92.45	82.94	79.29	74.01	69.45	72.68	81.87	80.56	86.59	83.36	81.59	74.35	81.86	86.12
	RanPAC [31]	91.87	83.54	79.23	75.11	70.23	74.65	82.91	70.94	81.65	80.76	80.12	67.72	79.96	84.87
	C-RanPAC	92.09	84.16	81.47	76.72	**72.77**	**76.22**	**83.73**	84.63	86.47	85.30	85.07	**78.23**	**83.72**	**87.43**
FGVC-Aircraft	NCM	27.21	20.85	17.1	13.07	15.01	13.98	21.37	24.31	5.6	16.88	18.86	25.75	27.45	33.93
	TEEN [50]	29.31	21.65	18.35	14.04	15.02	14.16	21.82	26.02	5.6	18.71	19.39	27.67	27.63	33.95
	FeCAM [11]	25.85	23.09	21.33	14.29	16.38	15.42	23.54	19.08	1.2	10.32	12.59	22.58	29.49	37.87
	C-FeCAM	32.42	26.39	23.63	15.71	17.55	16.38	24.9	30.19	15.81	20.78	24.96	29.42	31.17	39.13
	RanPAC [31]	33.37	27.78	25.88	17.21	21.89	21.12	32.0	26.15	6.81	21.32	14.84	35.94	38.22	48.53
	C-RanPAC	41.61	32.54	28.79	19.97	**24.33**	**21.87**	**33.63**	43.26	20.61	29.77	25.52	**43.89**	**40.32**	**50.43**

Table 1. Evaluation of methods in FSCIL settings on CUB200 and FGVC-Aircraft datasets with small and big first task settings. Best results in **bold** and second-best results are underlined.

where W refers to the random projection weights which are kept frozen after the first task, $\phi(.)$ denotes element-wise non-linear activation function, G refers to the summation of gram matrices of all classes, λ refers to the shrinkage parameter and c_y refers to the summation of features of class y. The Gram matrix G and the class prototypes (without averaging) c_y are updated in every task.

Calibrated RanPAC. In the FSCIL scenario, the estimates of G and c_y are not good and biased for the new few-shot classes, which results in high scores for the base classes and poor classification for the new classes. We propose to use the calibrated means and covariances (from Eq. (4)) to assume gaussian distributions of new classes and then sample features from these calibrated distributions as follows:

$$\hat{f}(x) \sim \mathcal{N}(\hat{\mu}_y, \hat{\Sigma}_y) \tag{8}$$

We compute G and c_y from these sampled features as follows:

$$G = \sum_{t=1}^{T} \sum_{i=1}^{N_t} h_{t,n} \cdot h_{t,n}, \quad C = \sum_{t=1}^{T} \sum_{i=1}^{N_t} h_{t,n} \cdot y_{t,n} \tag{9}$$

where $h_{t,n} = \phi(\hat{f}(x)W)$ and $(\cdot)$ refers to the outer products. Thus, we use the calibrated feature distributions of new few-shot classes and perform classification in the randomly projected high dimensional space.

4. Experiments

Datasets. We perform experiments on several publicly available datasets. CIFAR100 [23] contains 50,000 training images and 10000 images for testing, divided among 100 classes. CUB-200 [48] contains 5994 training images, 5794 testing images and covers 200 classes of birds. Stanford Cars [22] consists of 196 classes of car models with 8144 images for both training and testing. FGVC-Aircraft [28] has 10,200 images covering 102 classes of aircraft variants, out of which we use randomly selected 100 classes.

Settings. We use all available samples for each class in the many-shot base task and only 5 samples from each class in the few-shot tasks. We follow both the commonly used setting of having a big first task (50% of the classes in the first task) as well as the challenging small start setting (equally split the dataset in all tasks). In the big start setting, we use 50 classes in the base task for the FGVC-Aircraft dataset, 100 classes for CUB-200 and 98 classes for Stanford Cars. For the small start setting, we use 10 classes in the base task and all subsequent tasks for CIFAR-100 and FGVC-Aircraft datasets, 20 classes for CUB-200 and 28 classes for Stanford Cars in all tasks.

Implementation Details. Similar to [59], we use the ViT-B/16 model from the timm library which is pre-trained on ImageNet-21K and then finetuned on ImageNet-1k. We follow [4, 59] and use a ViT adaptor for adapting the pretrained ViT to the dataset in the first task for 40 epochs. We adapt the model in the base task and use the same model weights for different methods to ensure fair comparison. We follow the code-base from [42]. Different from [59], we do not merge the embeddings obtained using the old model with the current model embeddings and use only the current ones. For the prototype calibration in TEEN, C-FeCAM, and C-RanPAC, we use $\alpha = 0.9$ for CUB-200, Stanford Cars, and FGVC-Aircraft and $\alpha = 0.75$ for CIFAR-100. For the covariance calibration, we use $\beta = 1$ for C-FeCAM and $\beta = 0.5$ for C-RanPAC for all datasets. For C-RanPAC, we sample 800 points in the feature space for each class us-

	Small First Task (7 tasks)						Big First Task (8 tasks)					
	A_{HM}				A_{last}	A_{inc}	A_{HM}				A_{last}	A_{inc}
Task	1	2	4	6			1	3	5	7		
NCM	44.15	35.66	24.22	23.59	25.71	39.52	47.72	41.17	41.64	41.48	48.91	57.74
TEEN [50]	45.66	37.57	25.49	24.57	26.46	40.20	51.48	45.15	44.14	43.47	49.83	58.49
FeCAM [11]	41.71	32.63	24.22	23.4	27.07	41.61	39.12	33.69	34.37	30.95	50.39	62.43
C-FeCAM	54.18	43.61	31.87	28.86	30.71	45.65	57.91	52.15	50.03	49.16	57.21	66.77
RanPAC [31]	58.58	48.14	39.13	35.65	38.3	54.15	53.22	54.62	48.13	47.61	61.85	72.12
C-RanPAC	62.63	51.97	43.64	**38.18**	**40.73**	**56.05**	65.98	62.24	58.78	**57.82**	**65.3**	**74.32**

Table 2. Evaluation of methods in FSCIL settings on Stanford Cars dataset with small and big first task settings. Best results in **bold** and second-best results are underlined.

			A_{HM}			A_{last}	A_{inc}
Task	1	3	5	7	9		
NCM	92.26	88.42	81.92	76.79	80.18	79.46	86.57
TEEN [50]	93.45	89.44	81.71	75.83	80.95	79.39	86.95
FeCAM [11]	92.19	87.99	79.37	74.26	78.94	78.42	86.04
C-FeCAM	94.16	89.63	81.23	76.11	81.31	79.73	87.24
RanPAC [31]	91.79	85.25	80.24	75.37	79.85	78.96	85.73
C-RanPAC	92.51	88.89	82.88	77.49	**81.84**	**81.32**	**87.55**

Table 3. Evaluation of methods in FSCIL settings on CIFAR100 dataset (small-start setting with 10 tasks). Best results in **bold** and second-best results are underlined.

ing Eq. (8) to compute the matrix G and class-wise prototypes C in Eq. (9).

For FeCAM, we use $\gamma = 100$ following the original implementation from [11]. We do not use the tukey's transformation from [11] since the pre-trained ViTs do not have a final ReLU activation layer and can have negative values in the feature representations. For RanPAC, we follow the original implementation [31] and randomly project the features to 10,000 dimensional space. Following [31], we optimize the shrinkage parameter by computing G and c_y on 80% of the training set for some value of λ and then selecting the value which leads to minimum mean squared error between the predictions and the labels on the other 20% of the training set. However, since the training set is very small due to the few-shot nature of new tasks, the optimization of λ in new tasks leads to instability and performance collapse. In order to avoid this collapse, we optimize the λ value in the first task and then fix it for all tasks.

Note that FeCAM stores one (768×768) covariance matrix per class and RanPAC stores one big shared Gram matrix (10000×10000) across tasks in addition to the class prototypes. For the proposed calibration, we need to additionally store the covariance matrices of only the base classes of (768×768) size, to use them for calibration in future tasks.

Evaluation. We evaluate the performance of NCM [16, 35, 39], TEEN [50], FeCAM [11] and RanPAC [31] with the proposed statistics calibration. We refer to the calibrated versions as C-FeCAM and C-RanPAC. We evaluate in terms of the average accuracy after the last task (A_{last}), the average of the incremental accuracy from all tasks (A_{inc}) and the harmonic mean accuracy (A_{HM}) after each task. The harmonic mean accuracy better reflects the stability-plasticity trade-off. It is computed as

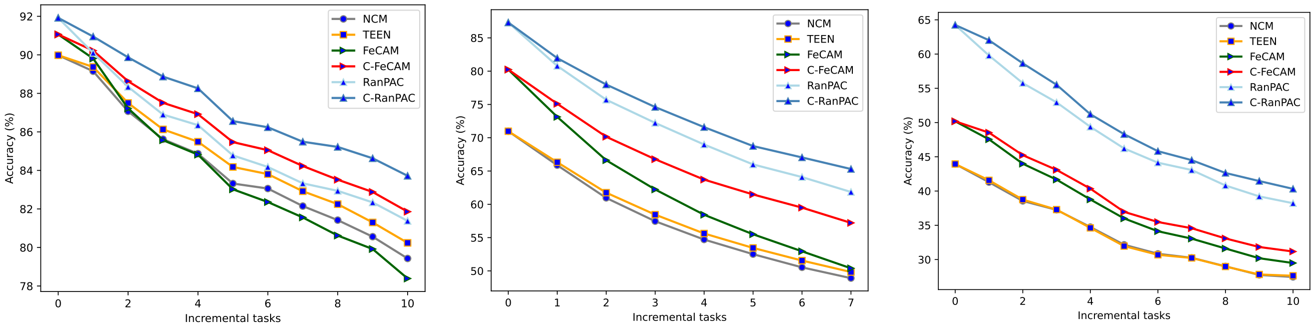
$$A_{HM} = \frac{2A_{old}A_{new}}{A_{old} + A_{new}} \tag{10}$$

Where A_{old} refers to the accuracy of all classes seen before the current task and A_{new} refers to the accuracy of the current task classes. The A_{HM} is low if the current task performance is poor, irrespective of the performance of the old classes. Thus, it is a good metric for evaluation in FSCIL benchmarks and is used in previous works [33, 50]. We show the A_{HM} after multiple tasks which reflects on the improved performance of the few-shot classses.

4.1. Quantitative Evaluation

We evaluate how the different methods discussed above perform in different settings in Tabs. 1 to 3. For each setting, we report the A_{HM} after alternate tasks, the average accuracy after the last task and the average incremental accuracy. We observe that TEEN [50] improves the harmonic mean accuracy marginally compared to NCM across all settings. While FeCAM shows similar performance to TEEN, it has a lower harmonic mean accuracy (64.42 compared to 71.33 in NCM for big-start CUB-200) due to poor estimates of covariance matrix for new classes. On using the proposed calibration, C-FeCAM improves significantly on the last task A_{HM} (from 64.42 to 74.35 for big-start CUB-200). Similarly, RanPAC has a lower last task A_{HM} (67.72 compared to 71.33 in NCM for big-start CUB-200) and on calibration, we see that C-RanPAC improves over RanPAC by 10.51 percentage points (pp).

Having a big first task helps to adapt the model better to the dataset and thus the big first task settings show bet-



(a) CUB-200 (b) Stanford Cars (c) FGVC-Aircraft

Figure 3. Accuracy after each incremental task for big-start settings on CUB-200, Stanford Cars and FGVC-Aircraft. Our proposed statistics calibration improves the average accuracy consistently after all tasks.

	C-FeCAM			C-RanPAC		
β	A_{HM}	A_{last}	A_{inc}	A_{HM}	A_{last}	A_{inc}
0.5	27.4	30.05	44.57	**38.18**	**40.73**	**56.05**
1.0	**28.86**	**30.71**	**45.65**	37.21	40.0	55.45
1.5	28.81	30.26	45.55	35.55	38.09	54.04

Table 4. Ablation on the impact of β in Eq. (4) for Stanford Cars (small-start setting). Best results in **bold**.

	μ_n Eq. (3)	Σ_n Eq. (4)	A_{HM}	A_{last}	A_{inc}
FeCAM	×	×	23.4	27.07	41.61
C-FeCAM	✓	×	24.45	28.16	42.65
C-FeCAM	✓	✓	**28.86**	**30.71**	**45.65**
RanPAC	×	×	35.65	38.3	54.15
RanPAC*	×	×	36.13	39.63	55.45
C-RanPAC	✓	×	37.32	39.85	55.39
C-RanPAC	✓	✓	**38.18**	**40.73**	**56.05**

Table 5. Ablation on the impact of prototype calibration and covariance calibration for Stanford Cars (small-start setting). Best results in **bold**.

ter performance compared to the small first task. Also, for the proposed statistics calibration, having more base classes will help to better calibrate the new few-shot class statistics. We observe this trend for all settings with higher margin of improvement in big-start settings. Using the challenging FGVC-Aircraft dataset in Tab. 1, we notice that all the methods struggle more in the small-start settings but still both C-FeCAM and C-RanPAC improve the last task A_{HM} by 1.17 pp and by 2.44 pp respectively, over the uncalibrated variants. For the big-start setting on FGVC-Aircraft, we see a bigger improvement of 6.84 pp and 7.95 pp for C-FeCAM and C-RanPAC respectively.

In Tab. 2, we observe a similar trend for Stanford Cars, where C-FeCAM improves the last task A_{HM} significantly over FeCAM by 5.46 pp for small-start setting and by 18.21 pp for big-start setting. Similarly, C-RanPAC improves over RanPAC by 2.53 pp and 10.21 pp for small-start and big-start settings respectively. We observe the same trend for CIFAR-100 in Tab. 3, where C-FeCAM outperforms FeCAM by 2.37 pp and C-RanPAC outperforms RanPAC by 1.99 pp for last task A_{HM}. These quantitative evaluations suggest that simply calibrating the second-order feature statistics by exploiting the base class statistics can improve few-shot classification significantly without any extra training complexity.

4.2. Ablation Studies

We perform ablation studies to observe the impact of the covariance scaling factor β in Tab. 4 and also the impact of the prototype calibration and the covariance calibration separately in Tab. 5. We observe both prototype calibration and covariance calibration improves the harmonic mean accuracy. The covariance calibration has a significant impact on the improvement of performance using both FeCAM and RanPAC. We also compare with the variant of RanPAC (denoted by RanPAC*) where we sample points from the class distributions and compute the probability instead of using just the available embeddings from few-shot training data.

Furthermore, to analyze how the average accuracy changes after every task, we show the accuracy plots in Fig. 3. We observe that using higher-order feature statistics, FeCAM and RanPAC already achieves better accuracy in the base task. In the incremental tasks, the proposed statistics calibration further improves both the methods and achieves a significant improvement after the last task. While the calibration improves the average accuracy after each task, the main improvement with the calibration is reflected in the harmonic mean accuracy in Tabs. 1 to 3 due to significantly better classification of few-shot classes.

5. Conclusion

In this work, we explore how prototype- and higher-order statistics-based classification methods work in FSCIL settings when using ViT models pre-trained on large scale datasets. We identify that although the higher-order statistics-based methods like FeCAM and RanPAC performs very well with many-shot data and in existing MSCIL benchmarks [11, 31], these methods struggles with few-shot data due to poor and biased estimates of distribution statistics (see Fig. 1). We propose to perform a simple yet effective statistics calibration by using the strong statistics estimates which are computed for the many-shot base classes. We demonstrate that using the proposed calibration outperforms all existing methods across multiple settings and datasets by a significant margin. We highlight the improvement in the classification of few-shot classes using the harmonic mean accuracy which is consistently better after all tasks with our proposed calibration method.

Acknowledgement. We acknowledge projects TED2021-132513B-I00 and PID2022-143257NB-I00, financed by MCIN/AEI/10.13039/501100011033 and FSE+ and the Generalitat de Catalunya CERCA Program. This work was partially funded by the European Union under the Horizon Europe Program (HORIZON-CL4-2022-HUMAN-02) under the project "ELIAS: European Lighthouse of AI for Sustainability", GA no. 101120237. Bartłomiej Twardowski acknowledges the grant RYC2021-032765-I.

References

[1] Touqeer Ahmad, Akshay Raj Dhamija, Steve Cruz, Ryan Rabinowitz, Chunchun Li, Mohsen Jafarzadeh, and Terrance E Boult. Few-shot class incremental learning leveraging self-supervised features. In *Conference on computer vision and pattern recognition*, 2022. 1

[2] Afra Feyza Akyürek, Ekin Akyürek, Derry Wijaya, and Jacob Andreas. Subspace regularizers for few-shot class incremental learning. In *International Conference on Learning Representations (ICLR)*, 2022. 3

[3] Eden Belouadah and Adrian Popescu. Il2m: Class incremental learning with dual memory. In *International Conference on Computer Vision (ICCV)*, 2019. 2

[4] Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. *Advances in Neural Information Processing Systems*, 2022. 1, 3, 5

[5] Matthias De Lange and Tinne Tuytelaars. Continual prototype evolution: Learning online from non-stationary data streams. In *International Conference on Computer Vision (ICCV)*, 2021. 2

[6] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 2021. 1, 2

[7] Yao Deng and Xiang Xiang. Expanding hyperspherical space for few-shot class-incremental learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024. 1, 3

[8] Prithviraj Dhar, Rajat Vikram Singh, Kuan-Chuan Peng, Ziyan Wu, and Rama Chellappa. Learning without memorizing. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021. 1

[10] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *European Conference on Computer Vision (ECCV)*, 2020. 2

[11] Dipam Goswami, Yuyang Liu, Bartłomiej Twardowski, and Joost van de Weijer. Fecam: Exploiting the heterogeneity of class distributions in exemplar-free continual learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 1, 2, 3, 4, 5, 6, 8

[12] Tyler L Hayes and Christopher Kanan. Lifelong machine learning with deep streaming linear discriminant analysis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 220–221, 2020. 2

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1

[14] Michael Hersche, Geethan Karunaratne, Giovanni Cherubini, Luca Benini, Abu Sebastian, and Abbas Rahimi. Constrained few-shot class-incremental learning. In *Conference on Computer Vision and Pattern Recognition*, 2022. 3

[15] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[16] Paul Janson, Wenxuan Zhang, Rahaf Aljundi, and Mohamed Elhoseiny. A simple baseline that questions the use of pretrained-models in continual learning. *arXiv preprint arXiv:2210.04428*, 2022. 2, 3, 6

[17] Jayateja Kalla and Soma Biswas. S3c: Self-supervised stochastic classifiers for few-shot class-incremental learning. In *European Conference on Computer Vision*, 2022. 3

[18] Haeyong Kang, Jaehong Yoon, Sultan Rizky Hikmawan Madjid, Sung Ju Hwang, and Chang D Yoo. On the soft-subnetwork for few-shot class incremental learning. In *The Eleventh International Conference on Learning Representations*, 2022. 3

[19] Ronald Kemker, Marc McClure, Angelina Abitino, Tyler Hayes, and Christopher Kanan. Measuring catastrophic forgetting in neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, 2018. 1

[20] Do-Yeon Kim, Dong-Jun Han, Jun Seo, and Jaekyun Moon. Warping the space: Weight space rotation for class-incremental few-shot learning. In *The Eleventh International Conference on Learning Representations*, 2022. 3

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

[22] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *International Conference on Computer Vision (ICCV-W) Workshops*, 2013. 2, 5

[23] Alex Krizhevsky. Learning multiple layers of features from tiny images. pages 32–33, 2009. 2, 5

[24] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 2017. 2

[25] Binghao Liu, Boyu Yang, Lingxi Xie, Ren Wang, Qi Tian, and Qixiang Ye. Learnable distribution calibration for few-shot class-incremental learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 4

[26] Xialei Liu, Marc Masana, Luis Herranz, Joost Van de Weijer, Antonio M Lopez, and Andrew D Bagdanov. Rotate your networks: Better weight consolidation and less catastrophic forgetting. In *24th International Conference on Pattern Recognition (ICPR)*. IEEE, 2018. 2

[27] Yuyang Liu, Yang Cong, Dipam Goswami, Xialei Liu, and Joost van de Weijer. Augmented box replay: Overcoming foreground shift for incremental object detection. In *International Conference on Computer Vision (ICCV)*, 2023. 2

[28] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 2, 5

[29] Marc Masana, Xialei Liu, Bartlomiej Twardowski, Mikel Menta, Andrew D Bagdanov, and Joost van de Weijer. Class-incremental learning: survey and performance evaluation. *Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 2022. 1, 2

[30] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*. Elsevier, 1989. 1

[31] Mark McDonnell, Dong Gong, Amin Parvaneh, Ehsan Abbasnejad, and Anton van den Hengel. Ranpac: Random projections and pre-trained models for continual learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 1, 2, 3, 4, 5, 6, 8

[32] Aristeidis Panos, Yuriko Kobe, Daniel Olmeda Reino, Rahaf Aljundi, and Richard E. Turner. First session adaptation: A strong replay-free baseline for class-incremental learning. In *International Conference on Computer Vision (ICCV)*, 2023. 2

[33] Can Peng, Kun Zhao, Tianren Wang, Meng Li, and Brian C Lovell. Few-shot class-incremental learning from an open-set perspective. In *European Conference on Computer Vision (ECCV)*, 2022. 1, 3, 6

[34] Grégoire Petit, Adrian Popescu, Hugo Schindler, David Picard, and Bertrand Delezoide. Fetril: Feature translation for exemplar-free class-incremental learning. In *Winter Conference on Applications of Computer Vision (WACV)*, 2023. 1, 2

[35] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 2, 6

[36] Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik-Manor. Imagenet-21k pretraining for the masses. In *Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2021. 1

[37] Anthony Robins. Catastrophic forgetting, rehearsal and pseudorehearsal. *Connection Science*, 1995. 1

[38] Ruslan Salakhutdinov, Joshua Tenenbaum, and Antonio Torralba. One-shot learning with a hierarchical nonparametric bayesian model. In *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*. JMLR Workshop and Conference Proceedings, 2012. 2, 4

[39] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 6

[40] Zeyin Song, Yifan Zhao, Yujun Shi, Peixi Peng, Li Yuan, and Yonghong Tian. Learning with fantasy: Semantic-aware virtual contrastive constraint for few-shot class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 1, 3

[41] Albin Soutif-Cormerais, Marc Masana, Joost Van de Weijer, and Bartłømiej Twardowski. On the importance of cross-task features for class-incremental learning. *International Conference on Machine Learning (ICML) Workshops*, 2021. 2

[42] Hai-Long Sun, Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. Pilot: A pre-trained model-based continual learning toolbox. *arXiv preprint arXiv:2309.07117*, 2023. 5

[43] Xiaoyu Tao, Xiaopeng Hong, Xinyuan Chang, Songlin Dong, Xing Wei, and Yihong Gong. Few-shot class-incremental learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 3

[44] Songsong Tian, Lusi Li, Weijun Li, Hang Ran, Li Li, and Xin Ning. Pl-fscil: Harnessing the power of prompts for few-shot class-incremental learning. *arXiv preprint arXiv:2401.14807*, 2024. 3

[45] Songsong Tian, Lusi Li, Weijun Li, Hang Ran, Xin Ning, and Prayag Tiwari. A survey on few-shot class-incremental learning. *Neural Networks*, 2024. 1, 3

[46] Gido M Van de Ven and Andreas S Tolias. Three scenarios for continual learning. *arXiv preprint arXiv:1904.07734*, 2019. 1, 2

[47] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 2016. 1

[48] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 2, 5

[49] Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application. *arXiv preprint arXiv:2302.00487*, 2023. 1, 2

[50] Qi-Wei Wang, Da-Wei Zhou, Yi-Kai Zhang, De-Chuan Zhan, and Han-Jia Ye. Few-shot class-incremental learning via training-free prototype calibration. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 1, 2, 3, 4, 5, 6

[51] 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)*, 2020. 4

[52] Shuo Yang, Lu Liu, and Min Xu. Free lunch for few-shot learning: Distribution calibration. In *International Conference on Learning Representations (ICLR)*, 2021. 2, 4

[53] Yibo Yang, Haobo Yuan, Xiangtai Li, Zhouchen Lin, Philip Torr, and Dacheng Tao. Neural collapse inspired feature-classifier alignment for few-shot class-incremental learning. In *The Eleventh International Conference on Learning Representations*, 2022. 3

[54] Chi Zhang, Nan Song, Guosheng Lin, Yun Zheng, Pan Pan, and Yinghui Xu. Few-shot incremental learning with continually evolved classifiers. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1

[55] Gengwei Zhang, Liyuan Wang, Guoliang Kang, Ling Chen, and Yunchao Wei. Slca: Slow learner with classifier alignment for continual learning on a pre-trained model. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 1

[56] Linglan Zhao, Jing Lu, Yunlu Xu, Zhanzhan Cheng, Dashan Guo, Yi Niu, and Xiangzhong Fang. Few-shot class-incremental learning via class-aware bilateral distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 3

[57] Da-Wei Zhou, Fu-Yun Wang, Han-Jia Ye, Liang Ma, Shiliang Pu, and De-Chuan Zhan. Forward compatible few-shot class-incremental learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 3

[58] Da-Wei Zhou, Han-Jia Ye, Liang Ma, Di Xie, Shiliang Pu, and De-Chuan Zhan. Few-shot class-incremental learning by sampling multi-phase tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2022. 1, 3

[59] Da-Wei Zhou, Han-Jia Ye, De-Chuan Zhan, and Ziwei Liu. Revisiting class-incremental learning with pre-trained models: Generalizability and adaptivity are all you need. *arXiv preprint arXiv:2303.07338*, 2023. 1, 3, 5

[60] Da-Wei Zhou, Hai-Long Sun, Jingyi Ning, Han-Jia Ye, and De-Chuan Zhan. Continual learning with pre-trained models: A survey. *arXiv preprint arXiv:2401.16386*, 2024. 1

[61] Fei Zhu, Xu-Yao Zhang, Chuang Wang, Fei Yin, and Cheng-Lin Liu. Prototype augmentation and self-supervision for incremental learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 2