

Robust Feature Learning and Global Variance-Driven Classifier Alignment for Long-Tail Class Incremental Learning

Jayateja Kalla and Soma Biswas

Department of Electrical Engineering
Indian Institute of Science, Bangalore, India.

{jayatejak, somabiswas}@iisc.ac.in

Abstract

This paper introduces a two-stage framework designed to enhance long-tail class incremental learning, enabling the model to progressively learn new classes, while mitigating catastrophic forgetting in the context of long-tailed data distributions. Addressing the challenge posed by the under-representation of tail classes in long-tail class incremental learning, our approach achieves classifier alignment by leveraging global variance as an informative measure and class prototypes in the second stage. This process effectively captures class properties and eliminates the need for data balancing or additional layer tuning. Alongside traditional class incremental learning losses in the first stage, the proposed approach incorporates mixup classes to learn robust feature representations, ensuring smoother boundaries. The proposed framework can seamlessly integrate as a module with any class incremental learning method to effectively handle long-tail class incremental learning scenarios. Extensive experimentation on the CIFAR-100 and ImageNet-Subset datasets validates the approach's efficacy, showcasing its superiority over state-of-the-art techniques across various long-tail CIL settings. Code is available at <https://github.com/JAYATEJAK/GVAlign>.

1. Introduction

In the realm of computer vision, the rapid progress of convolutional neural networks (CNNs) trained on balanced datasets has led to remarkable advancements [22, 30, 31]. However, real-world scenarios frequently involve large-scale datasets characterized by imbalanced and long-tailed distributions [24, 26, 37, 44]. In long-tail distributions, the categories with a majority of samples are termed as *long* classes, while those with fewer samples are termed *tail* classes. This inherent data distribution imbalance poses a significant challenge when training models for computer vision tasks. Within this context, tail classes encounter substantial under-representation during the training process,

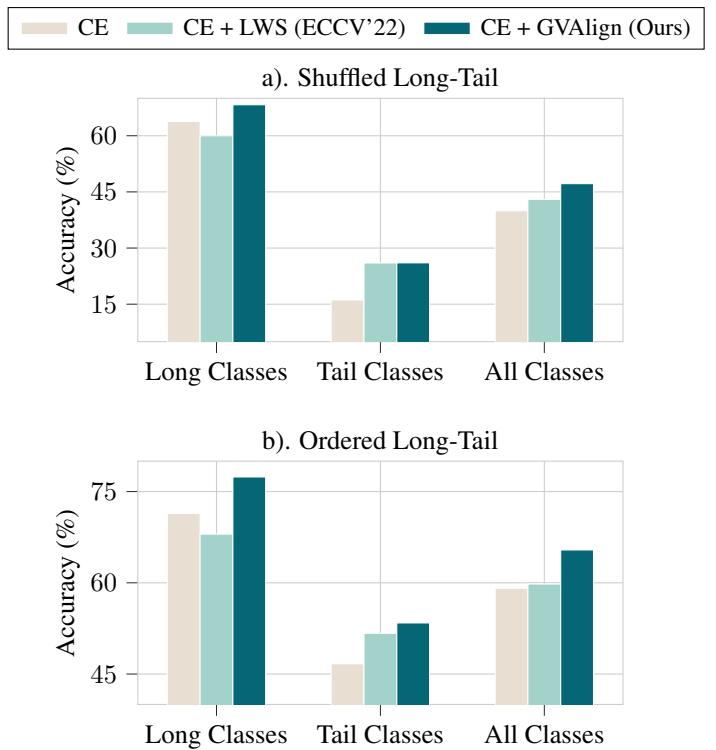


Figure 1. The performance of long, tail, and all classes is illustrated for two long-tail distributions as proposed in [25]. It is evident that training the Learnable Weight Scaling (LWS) layer with cross-entropy (CE) loss leads to a reduction in performance for the long classes, while simultaneously improving the performance of the tail classes. In contrast, our proposed approach, which leverages robust features and classifier alignment, exhibits an enhancement in the performance of both long and tail classes, thereby improving the overall all classes performance.

negatively impacting its recognition performance for these minority categories [44]. Furthermore, the model tends to exhibit bias towards long classes, due to extensive training data available for these majority categories.

Moreover, in real-time applications, not all class cate-

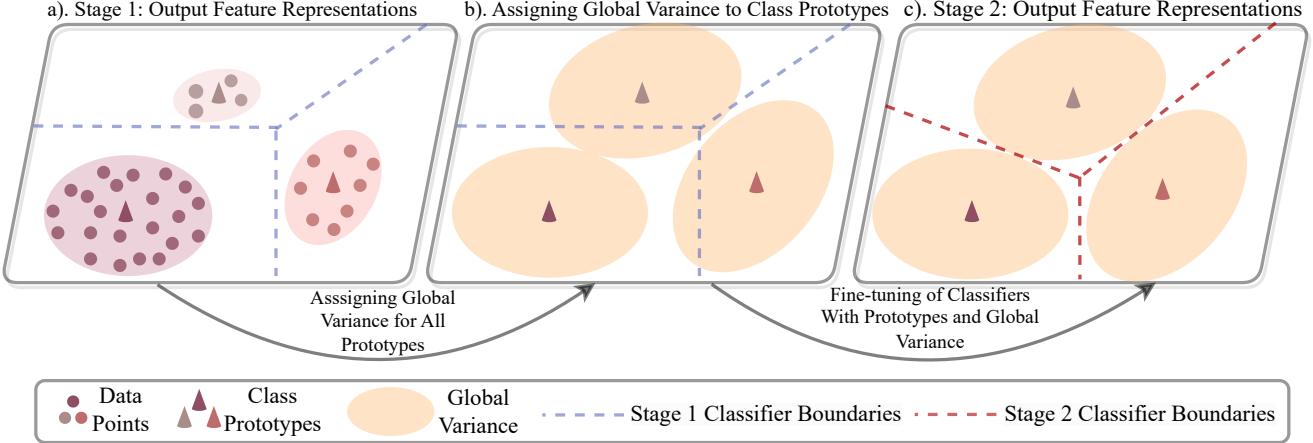


Figure 2. The leftmost figure ‘a’ illustrates the robust feature representations obtained in stage 1. Once robust representations are acquired, the global variance is assigned across all prototypes figure ‘b’. Sample feature representations drawn using prototypes and these global variance are then utilized to align the classifiers in stage 2. The rightmost figure ‘c’ depicts the aligned classifiers achieved after stage 2. The global variance signifies the data covariance of the base class with the highest number of samples.

gories are concurrently accessible; data becomes available in a continuous manner, and previous classes data might not be accessible due to privacy or storage limitations [23]. Expanding our model’s knowledge to encompass this continuously evolving data is of paramount importance. In the existing literature, the process of incrementally adding these classes to deep neural networks is referred to as class incremental learning (CIL) [33]. In this context, the addition of a set of new class information into the model is termed a *task*. At the end of each task in CIL, the model is evaluated on all the classes encountered so far. Typically, the initial task is trained using the cross-entropy (CE) loss and is often referred to as the base task, and gradually new classes are added at each incremental task.

Recently, Liu et al. [25] introduced long-tail distributions into the domain of class incremental learning (CIL) and coined the term “*long-tail class incremental learning*”. This approach involves the model’s endeavour to progressively learn new classes without succumbing to catastrophic forgetting of previously learned classes from the long-tailed data distributions at every task. Liu et al. [25] introduced a two-stage approach to address the challenges in long-tail CIL, where at each incremental task, the model learns through two stages. In the initial stage, they employed conventional incremental learning methods like UCIR [16] or PODNET [11]. Subsequently, in the second stage, they fixed the model parameters and trained an additional layer, the learnable weight scaling (LWS) layer, using a balanced dataset to address the issues in long-tail CIL.

To better understand the effectiveness of this two-stage LWS framework, we conducted experiments on two long-tail distributions as proposed by Liu et al. [25] on the CI-

FAR100 [21] dataset. Here, we take 50 classes from the CIFAR100 dataset and partitioned it into two categories: long classes (25 classes) and tail classes (25 classes), based on the number of samples available. Upon fine-tuning the model with the LWS layer using CE loss, we observed a reduction in performance on the long classes and a concurrent improvement in performance on the tail classes across both long-tail scenarios from Figure 1.

Inherent under-representation of tail classes within long-tail representations often results in misaligned or inadequately defined classifier boundaries. Adjusting these boundaries with balanced data samples can adversely affect the performance of long classes. To tackle this challenge in the context of class incremental learning (CIL), we propose a novel two-stage framework, termed **GVAlign** (**G**lobal **V**ariance-**D**riven **C**lassifier **A**lignment). In this framework, during the second stage, we propose aligning all classifiers based on global variance and class prototypes, thus eliminating the need for balanced data (which compels the model to repeatedly encounter the same data for tail classes) or additional layers. This global variance, as an informative measure, effectively captures class properties and it is intuitive to align the classifiers based on this information. Importantly, incorporating this approach not only preserves performance on long classes but also enhances performance on tail classes. Figure 2 illustrates the classifier alignment of our proposed approach. Achieving such alignment of the classifier through global variance requires the presence of robust features and distinct class separations marked by smoother boundaries. To meet this prerequisite, we introduce the incorporation of mixup classes during the initial stage. This strategic addition contributes to the culti-

vation of robust feature representations, ultimately enhancing the approach’s effectiveness.

Figure 1 demonstrates that the proposed classifier alignment strategy, coupled with robust feature learning, enhances the performance of tail classes without reducing the performance of long classes. This implies an improvement in the overall performance across all classes. The paper makes the following contributions:

- We introduce a novel two-stage approach, termed **GVA**lign (Global Variance-Driven Classifier Alignment) encompassing robust feature learning in the first stage and classifier alignment in the second stage using global variance as a informative measure to address the issues in long-tail class incremental learning.
- Extensive experiments conducted on datasets CIFAR-100 and ImageNet-Subset demonstrate the effectiveness of our proposed approach over state-of-the-art methods across various long-tail CIL settings.

2. Related works

This section summarises the works related to incremental and long-tail learning.

2.1. Class Incremental Learning

Class Incremental Learning (CIL) aims to progressively acquire knowledge about new classes without relying on task-specific information. However, learning from newly annotated class data with abundant samples presents the challenge of catastrophic forgetting, where the model forgets the representations of old class data. The CIL approaches in the literature can be categorized into three groups based on their strategies to mitigate the problem of catastrophic forgetting: 1) *Regularization-based* methods [2, 20, 32, 41] incorporate penalty-based loss terms at each incremental step on the learnable model weights according to their importance. 2) *Distillation-based* Recent CIL approaches adopt distillation-based methods, using teacher-student distillation loss [14] to mitigate catastrophic forgetting. In Learning without Forgetting (LwF) [23], distillation loss is used alongside cross-entropy. Similarly, iCaRL [33] combines distillation loss with older-task exemplars selected through herding. Methods like BiC [38] introduce new-class bias correction layers, and LwM [10] introduce information-preserving penalties or attention loss to counter model bias towards new classes. UCIR [16] combines distillation loss with cosine normalization and inter-class separation constraint, while PODNET [11] proposes polled distillation loss to address catastrophic forgetting. Some recent works [40, 46, 47] focus on non-exemplar-based methods without access to old class exemplars. 3) *Architecture-based methods* [1, 18, 28, 29, 34, 39]

These methods modify the network’s width and depth at each incremental step. Network expansion is often proposed to learn new tasks, but this can be computationally intensive. An alternative approach is to select sub-networks from the entire architecture using masks [1, 28, 29], storing the learned masks in memory. However, these methods require task-specific labels at inference time, which may not always be available in practical scenarios.

In this work, both UCIR [16] and PODNET [11] serve as stage 1 baselines in the context of Long-Tail CIL. However, the proposed approach can also function as a module within other CIL methods.

2.2. Long-Tail Learning

The long-tailed learning problem has garnered extensive attention due to the prevalence of data imbalance issues in real-world scenarios. To tackle this challenge, various approaches have been explored to mitigate the disparity between the distribution of majority and minority classes. Some of the prominent techniques are: 1) *Data Processing Methods* [4, 6, 7, 12, 13] such as over-sampling to amplify tail data, under-sampling to reduce head data, and data Augmentation to extend tail data. 2) *Class-level Reweighting* [8, 15, 17, 35] involves assigning different weights to classes to prioritize learning from the tail classes. Another approach, 3) *Decoupling* [19, 43, 45], also referred to as a two-stage approach, involves separating representation learning and classifier learning into distinct stages to enhance performance on tail classes.

2.3. Long-Tail Incremental Learning

Recently, Liu et al. [25] introduced long-tail distributions into class incremental learning. They drew inspiration from the decoupling strategy’s learnable weight scaling (LWS) approach [19], wherein an additional two-stage process involves training added layers using a balanced dataloader. This strategy necessitates careful design of learning approaches to effectively learn these supplementary weights [19].

In our proposed GVAAlign approach, we also employ a two-stage strategy. However, distinct from the aforementioned method, we align the classifiers using prototypes and covariance without the need for a balanced dataloader or additional layers. Our strategy is more generalised due to the exploration of the feature space through sampled data points using global variance as informative measure, resulting in improved alignment of the classifiers without compromising on long classes’ performance. Next, we will discuss the proposed methodology.

3. Problem Definition and Motivation

In this section, we begin by providing a clear explanation of the notations used in this paper. Subsequently,

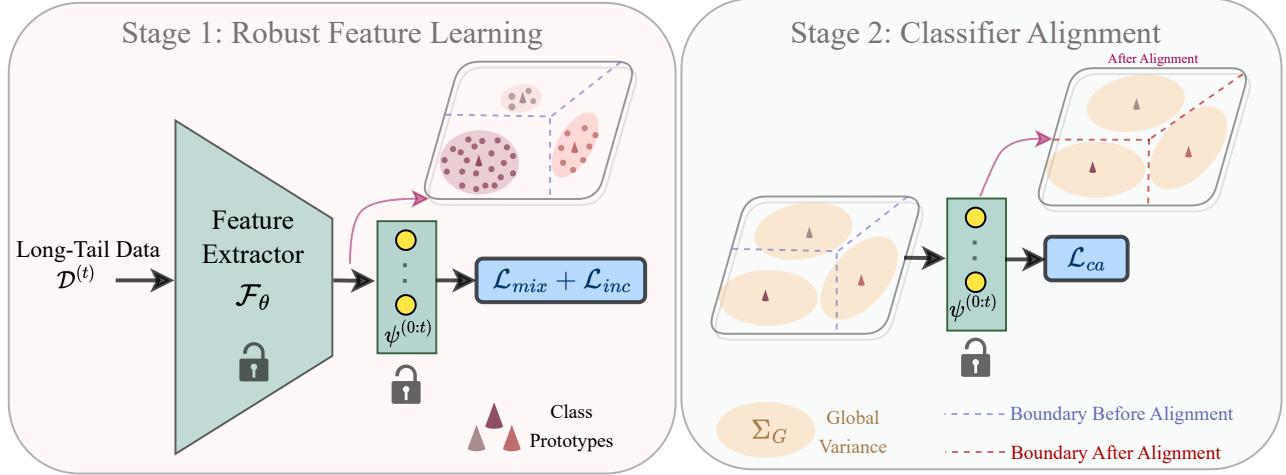


Figure 3. In stage 1, the model is trained using incremental learning approaches' loss \mathcal{L}_{inc} , supplemented by the mixup loss \mathcal{L}_{mix} , to obtain robust features. In stage 2, the feature extractor remains fixed, and only the classifiers are fine-tuned using the global variance and prototypes of the classes using classifier alignment loss \mathcal{L}_{ca} .

we delve into our two-stage training approach. In class incremental learning, the model is sequentially trained on a total of T tasks, with its data stream denoted as $\{\mathcal{D}^{(0)}, \mathcal{D}^{(1)}, \dots, \mathcal{D}^{(T)}\}$, and the corresponding classes set represented by $\{\mathcal{C}^{(0)}, \mathcal{C}^{(1)}, \dots, \mathcal{C}^{(T)}\}$. One assumption in CIL is there are no common classes across different tasks i.e. $\mathcal{C}^{(k)} \cap \mathcal{C}^{(l)} = \emptyset$ when $k \neq l$. At each task t , the model has access to data $\mathcal{D}^{(t)} = \{x_i, y_i\}_{i=1}^{n^{(t)}}$, where $n^{(t)}$ represents the number of samples in $\mathcal{D}^{(t)}$, and $y_i \in \mathcal{C}^{(t)}$. In CIL, the number of samples for each class is the same and equal to $\frac{n^{(t)}}{|\mathcal{C}^{(t)}|}$; however, in the case of long-tail CIL, the data distribution of $\mathcal{D}^{(t)}$ adheres to a long-tail distribution. In both CIL and long-tail CIL, during training task t , alongside $\mathcal{D}^{(t)}$, the model has access to an exemplar bank \mathcal{E} comprising a limited number of samples from earlier classes $\mathcal{C}^{(0:t-1)}$ and the end goal after task t is to classify all the classes seen so far i.e. $\mathcal{C}^{(0:t)}$. The addition of exemplar bank \mathcal{E} creates more imbalance in training data and makes long-tail CIL more challenging.

The model that learns sequentially is denoted as $\Theta = \{\mathcal{F}_\theta, \psi\}$, where \mathcal{F}_θ represents the feature extractor with parameters θ , and classifiers are represented by ψ . While the number of parameters in the feature extractor remains constant, the classifier layer parameters are incrementally added for each new task to accommodate novel classes. During the training task t , new classifiers $\psi^{(t)}$ are introduced alongside existing classifiers $\psi^{(0:t-1)}$ to classify all the classes seen so far. By using the training with data $\{\mathcal{D}^{(t)} \cup \mathcal{E}\}$, at the end of task t , the trained model $\Theta^{(t)} = \{\mathcal{F}_\theta, \psi^{(0:t)}\}$ is able to classify all classes from $\mathcal{C}^{(0:t)}$.

4. Proposed GVAlign Framework

In traditional long-tail learning, two-stage methods have shown promising results [19, 43, 45]. These approaches decouple the feature extraction in the first stage and classifier tuning in the second stage using balanced sampling techniques [19]. However, the direct application of such methodologies to the context of long-tail CIL encounters challenges posed by catastrophic forgetting [25]. To tackle the issues in long-tail CIL, we introduce a novel approach that entails learning robust feature representations in the first stage and refining classifier alignment in the second stage to mitigate the class imbalance problem. In the following sections, we elaborate on our stage 1 and stage 2 training procedures. Figure 3 shows the overall idea of our proposed two-stage approach.

4.1. Stage 1: Robust Feature Learning

Inspired by [25], we have incorporated conventional CIL techniques, such as UCIR [16] and PODNET [11], into stage 1 to address class incremental learning. However, it's essential to note that our approach is not limited to these specific methods; we are adaptable to any CIL technique for stage 1. In each incremental task, the loss computed by CIL methods is denoted as \mathcal{L}_{inc} . As mentioned earlier, robust feature space representations are crucial for effectively tackling long-tail CIL challenges. To bolster the robustness of features at this stage, in conjunction with \mathcal{L}_{inc} , we propose the utilization of mixup loss [42]. This implicitly accounts for the incremental stages and complements the classifier tuning stage.

Suppose we have (x_m, y_m) and (x_n, y_n) from $\mathcal{D}^{(t)}$, where x_m and x_n represent images and y_m and y_n are one-

hot labels. We formulate the mixup samples and labels as follows:

$$\tilde{x} = \lambda x_m + (1 - \lambda)x_n \quad (1)$$

$$\tilde{y} = \lambda y_m + (1 - \lambda)y_n \quad (2)$$

Here, λ is drawn from a Beta distribution, i.e., $\lambda \sim Beta(1, 1)$. The mixup loss is then calculated as $\mathcal{L}_{mix} = \mathcal{L}_{CE}(\tilde{x}, \tilde{y})$, where $\mathcal{L}_{CE}(x, y) = -\sum_{k=1}^K y_k \log(p_x)_k$ is the cross-entropy loss calculated for K classes. $p_x = \Theta^{(t)}(x)$ represents softmax outputs of the model. The total loss for stage 1 is

$$\mathcal{L}_{stage1} = \mathcal{L}_{inc} + \mathcal{L}_{mix} \quad (3)$$

4.2. Stage 2: Global Variance-Driven Classifier Alignment

In the second stage, we use the class prototypes and the estimated global variance to perform the classifier alignment as described below.

Construction of Proto Bank: Following the completion of stage 1 training, we proceed to compute a comprehensive class prototype bank denoted as \mathcal{P} . The objective of this prototype bank is to facilitate the alignment of classifiers in stage 2 and constructed using prototypes of all classes seen so far. Specifically, for each class k , the corresponding class prototype P_k is calculated using the following equation:

$$P_k = \frac{1}{N_k} \sum_{\{\mathbf{x}, y\} \in (\mathcal{D}^{(t)} \cup \mathcal{E})} \mathbb{I}_{(y=k)} \mathcal{F}_\theta(\mathbf{x}) \quad (4)$$

where N_k represents the number of samples in k^{th} class, and the indicator variable $\mathbb{I}_{(y=k)}$ equals 1 if the sample belongs to the k^{th} class (i.e. $y = k$).

Estimation of Global Variance: In scenarios with long-tail data distributions, relying on tail classes for variance calculation can result in an inaccurate variance estimate that does not accurately capture the central tendencies of the class. To address this, the proposed approach uses the class with the largest number of samples during the base task ($t = 0$) for global variance estimation. By aligning all classifiers based on this reliable estimate, we significantly enhance the model's capacity to generalize and discriminate across diverse class distributions. The global variance Σ_G is computed as follows

$$\Sigma_G = \frac{1}{N_G - 1} \sum_{i=1}^{N_G} (X_i - \bar{X})(X_i - \bar{X})^T \quad (5)$$

where X is the matrix that contains data points from the class with the highest number of samples and \bar{X} is the mean

Algorithm 1: Proposed GVAlign Framework for Long-Tail Class Incremental Learning

```

Input:  $\Theta = \{\mathcal{F}_\theta, \psi\} \leftarrow$  Model;
 $\{\mathcal{D}^{(0)}, \mathcal{D}^{(1)}, \dots, \mathcal{D}^{(T)}\} \leftarrow$  Data stream;
 $e_1 \leftarrow$  No.of epochs in stage 1;
 $e_2 \leftarrow$  No.of epochs in stage 2;
 $\mathcal{E} = \{\} \leftarrow$  Empty exemplar buffer;
for  $t \leftarrow 0$  to  $T$  do
   $\mathcal{D}^{(t)} = \{x_i, y_i\}_{i=1}^{N_t};$ 
  for  $e \leftarrow 1$  to  $e_1$  do
     $\mathcal{B} = \text{SampleMiniBatch}(\mathcal{D}^{(t)} \cup \mathcal{E});$ 
     $\mathcal{O}^{(0:t)} = \psi^{(0:t)}(\mathcal{F}_\theta(\mathcal{B}));$ 
     $\mathcal{L}_{inc} = \text{IncrementalLoss}(\mathcal{B}, \mathcal{O}^{(0:t)});$ 
     $\mathcal{L}_{mix} = \text{MixUpLoss}(\mathcal{B});$ 
     $\Theta \leftarrow \text{UpdateParameters}(\mathcal{L}_{inc} + \mathcal{L}_{mix});$ 
  end for
   $\mathcal{P} \leftarrow \text{CalculatePrototypes}(\mathcal{D}^{(t)} \cup \mathcal{E});$ 
  if  $t=0$  then
     $\Sigma_G = \text{GlobalVaraince}(\mathcal{D}^{(0)});$ 
  end if
  for  $e \leftarrow 1$  to  $e_2$  do
     $\mathcal{P}' \leftarrow \text{SampleProtoFromGlobalVar}(\mathcal{P}, \Sigma_G);$ 
     $\mathcal{O}^{(0:t)} = \psi^{(0:t)}(\mathcal{P}');$ 
     $\mathcal{L}_{ca} = \text{ClassifierAlignLoss}(\mathcal{O}^{(0:t)});$ 
     $\psi^{(0:t)} \leftarrow \text{UpdateParameters}(\mathcal{L}_{ca});$ 
  end for
   $\mathcal{E} \leftarrow \text{UpdateExemplars}(\mathcal{D}^{(t)});$ 
return  $\Theta;$ 

```

vector of those samples. N_G is the number of samples in that class.

Classifier Alignment: This stage 2 training involves leveraging the computed global variance Σ_G as an informative measure to explore the feature space around the prototypes \mathcal{P} of all classes. This exploration aids in aligning the classifiers effectively, facilitating improved classification performance. At this stage only classifiers are tuned using pseudo-augmented samples $\mathcal{P}' \sim \mathcal{N}(\mathcal{P}, \Sigma_G)$ generated from normal distribution by employing prototypes as means and the global variance as covariance information. The classifier alignment loss calculated during this stage as follows

$$\mathcal{L}_{ca} = - \sum_{q \in \mathcal{P}'} \sum_{k=1}^K \hat{y}_k \log(\psi^{(0:t)}(q))_k \quad (6)$$

where \hat{y} represents the prototype one-hot label and K represents the all classes seen so far. The exemplar set \mathcal{E} is updated using herding [33] technique, which is a commonly employed technique in CIL approaches to store exemplars. The complete training procedure is summarized in Algorithm 1.

Long-tail distribution type → Method ↓	Ordered long-tail				Shuffled long-tail			
	CIFAR-100		ImageNet-Subset*		CIFAR-100		ImageNet-Subset*	
	5 tasks	10 tasks	5 tasks	10 tasks	5 tasks	10 tasks	5 tasks	10 tasks
UCIR	42.69	42.15	56.45	55.44	35.09	34.59	46.45	45.31
UCIR + LWS (ECCV 2022)	45.88	45.73	57.22	55.41	39.40	39.00	49.42	47.96
UCIR + GVAAlign (Ours)	47.13	46.82	58.08	56.68	42.80	41.64	50.69	47.58
PODNET	44.07	43.96	59.16	57.74	36.64	34.84	47.61	47.85
PODNET + LWS (ECCV 2022)	44.38	44.35	60.12	59.09	36.37	37.03	49.75	49.51
PODNET + GVAAlign (Ours)	48.41	47.71	61.06	60.08	42.72	41.61	52.01	50.81

Table 1. Experimental results on long-tail class incremental learning (* signifies that we have rerun all experiments on the Imagnet-Subset 100 dataset. Comprehensive dataset details and data can be found in the GitHub repository for reproducibility).

5. Experiments

In this section, we discuss the datasets used, implementation details, and the results obtained in both long-tail and conventional CIL settings.

5.1. Datasets and Evaluation Protocol

To evaluate the efficacy of our proposed framework, we conducted experiments using two benchmark datasets specifically designed for long-tail CIL [25]: CIFAR100 and the ImageNet Subset. For a comprehensive and fair comparison, we adopted the data task splits recommended in [25], utilizing 50 classes for the base task. Then in the 5-task configuration (T=5), we progressively introduced 10 new classes during each incremental task i.e. (50–10–⋯–10). Similarly, in the 10-task setup (T=10), we incorporated 5 new classes in each incremental task i.e. (50–5–⋯–5). Our approach followed the same long-tail distributions as proposed in [25].

CIFAR-100: This dataset comprises 50,000 training images and 10,000 test images, each image consisting of 32x32 pixels. These images are distributed across 100 classes.

ImageNet Subset: The ImageNet Subset consists of 100 classes, sampled from the larger ImageNet dataset [22]. All images were resized to 256x256 pixels and subsequently randomly cropped to 224x224 pixels during the training phase. We evaluated all methods on this dataset to ensure reliable and accurate evaluation.

We employ the widely recognized CIL evaluation metric, average incremental accuracy [27, 33]. Here, let t represent the task ID, where $t \in 0, 1, \dots, T$. We define $Acc_{0:n}^t$ as the model’s accuracy on the test data of all tasks from 0 to n after learning task t , where $n \leq t$. Consequently, upon completion of task T , the average incremental accuracy is computed as $\frac{1}{T} \sum_{t=0}^T Acc_{0:t}^t$. We utilized the same model architectures as in [25] to ensure a fair comparison. Specifically, ResNet-32 was employed for CIFAR-100, while ResNet-18 was chosen for the ImageNet Subset dataset.

Our training protocol involved initiating the learning rate

at 0.1 and subsequently reducing it by a factor of 10 after the 250th, 350th, and 450th epochs, resulting in a total 500 epochs for CIFAR-100 training. As for the ImageNet Subset, the learning rate was set to 0.1 at the start and reduced by a factor of 10 after the 30th and 60th epochs, resulting in a total of 90 epochs for training. Throughout all experiments, a fixed batch size of 128 was used. During 2-stage classifier alignment training, we tuned only classifier layers with a learning rate of 0.1 for 100 epochs. We considered a standard of maximum 20 exemplars for each class to ensure a fair comparison with other methods. We used NVIDIA RTX A5000 24GB card to run all our experiments.

5.2. Results on Long-Tail CIL

First, we report the results on the long-tailed CIL task, which is the main focus of this work. We integrate the proposed GVAAlign framework with UCIR and PODNET as in [25] and compare with the state-of-the-art approach [25], which is the only work which addresses the challenging long-tailed CIL to the best of our knowledge. We observe from Table 1 that across both the datasets (CIFAR100 and ImageNet Subset) and different task setups (T=5 and T=10), our approach consistently achieves higher average incremental accuracy over the state-of-art long tail CIL method. This improvement is consistent for both ordered and shuffled long-tail distributions. Specifically, when combined with the UCIR approach on shuffled long-tail distributions, our method boosts CIFAR100 accuracy by 3.4% and ImageNet Subset accuracy by 1.2% in the 5-task scenario. On ordered long-tail distributions, we see a 1.25% increase for CIFAR100 and a 0.86% increase for ImageNet Subset in the same 5-task scenario. Notably, these gains are even more significant when PODNET serves as the baseline for CIL. With shuffled long-tail distributions, our approach achieves a remarkable 6.35% improvement on CIFAR100 and a substantial 2.26% increase on ImageNet Subset in the 5-task context. Similar improvements are seen in ordered long-tail scenarios, with gains of 4.03% on CIFAR100 and 0.94% on ImageNet Subset.

Method	CIFAR 100		ImageNet-Subset*	
	5 tasks	10 tasks	5 tasks	10 tasks
UCIR	61.15	58.74	69.11	65.15
UCIR + LWS (ECCV 2022)	63.48	60.57	68.83	66.47
UCIR + GVAAlign (Ours)	64.11	61.23	70.05	66.60
PODNET	63.15	61.16	67.92	62.39
PODNET + LWS (ECCV 2022)	64.58	62.63	69.43	62.12
PODNET + GVAAlign (Ours)	65.73	63.72	68.85	62.42

Table 2. Experimental results on traditional class incremental learning (* signifies that we have rerun all experiments on the Imagnet-Subset 100 dataset).

5.3. Results on Conventional CIL

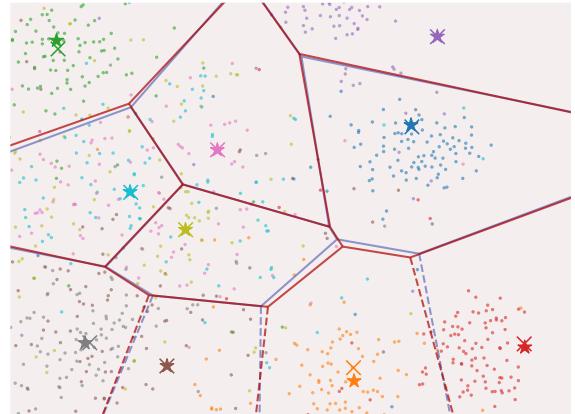
Conventional CIL is also inherently imbalanced, since during the incremental stages, there might be very few exemplars from the earlier classes available along with large number of examples of the new classes. Thus, it is important to also evaluate the effectiveness of the approaches developed for long-tailed CIL setting for the conventional CIL scenario. Table 2 reports the average incremental accuracy achieved by the proposed GVAAlign framework in conventional CIL setups. We observe that the proposed approach doesn't just excel in long-tail distributions; it also improves conventional CIL.

6. Analysis & Ablation Studies

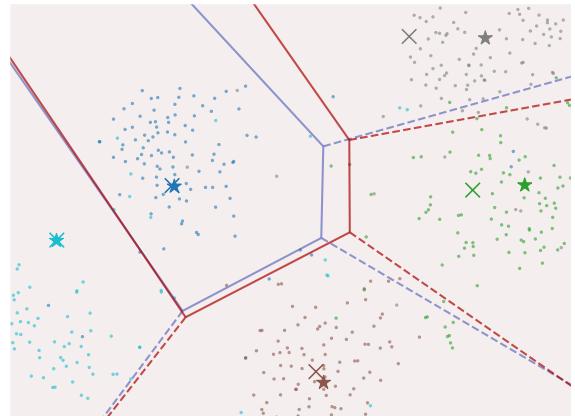
In this section, we delve into the analysis of our proposed approach. Firstly, we analyze the alignment of classifiers using Voronoi plots [3]. Next, we examine the impact of the number of exemplars in the context of long-tail CIL. Notably, our approach consistently outperforms state-of-the-art techniques in various long-tail CIL scenarios, irrespective of the number of exemplars utilized. Subsequently, we explore the benefits of our approach in a conventional setting where all new classes contain equal samples. We observe that due to robust learning and feature space exploration, our method enhances the separation between semantically similar classes. This improvement alleviates potential confusion between these classes, ultimately leading to enhanced performance.

6.1. Analyzing Classifiers

Our approach's key contribution lies in effectively aligning classifiers during incremental learning for long-tail data distributions. To highlight the significance of this alignment, we utilize Voronoi plots to illustrate the learning of new classes during task 1 in the context of shuffled long-tail CIL. Voronoi plots visually illustrate feature space regions assigned to different classes, providing insights into classifier behavior. We present visualizations for two different settings: $T = 5$, where 10 new classes are introduced, and $T = 10$, where 5 new classes are added in Figure 4. The alignment of classifiers results in a tangible enhancement in



(a) Scenario $T = 5$: Addition of 10 new classes (before alignment 33.91%, after alignment 38.88%).



(b) Scenario $T = 10$: Addition of 5 new classes (before alignment 43.61%, after alignment 50.01%).

Figure 4. Voronoi class boundaries in the shuffled long-tail CIL scenario after task 1. The symbols ‘ \times ’ indicate the initial classifiers and ‘ \star ’ represents the aligned classifiers. Class boundaries before alignment are marked by — and after alignment marked by —.

accuracy for the newly introduced classes. Specifically, in the $T=5$ scenario, the classification accuracy on task1 improves from 33.91% to 38.88%. Similarly, in the $T=10$ scenario, the accuracy rises from 43.61% to 50.01%.

6.2. Effect of Number of Exemplars

To understand the effect of the number of exemplars, we conduct extensive experiments with exemplar counts of $\{5, 10, 15, 20\}$, as depicted in Figure 5. Across both shuffled and ordered long-tail scenarios, our approach consistently outperforms existing long-tail CIL methods. Notably, it also exhibits strong performance in conventional CIL settings. As the number of exemplars increases, our approach's ability is even more pronounced due to the precise position of prototypes in the representation space.

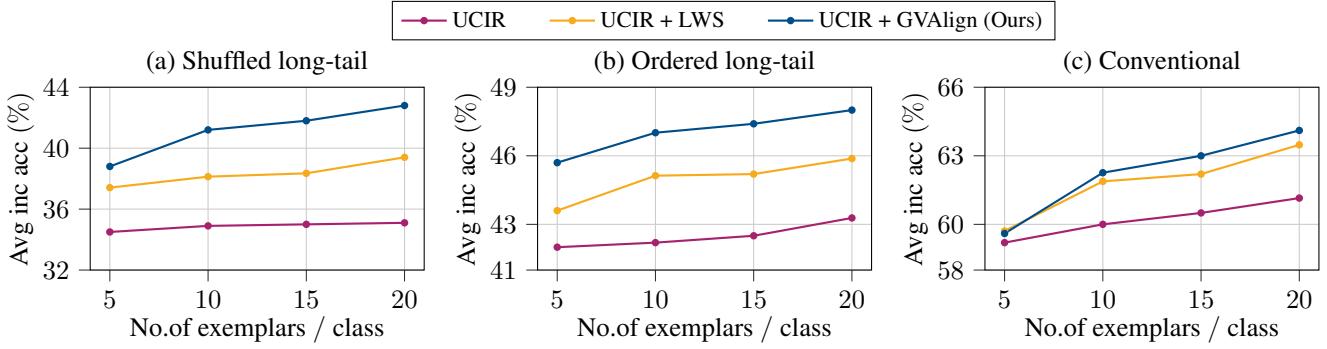


Figure 5. Illustrates how our proposed approach consistently improves with an increased number of exemplars, benefiting from precise prototype positioning as the number of exemplars increases.

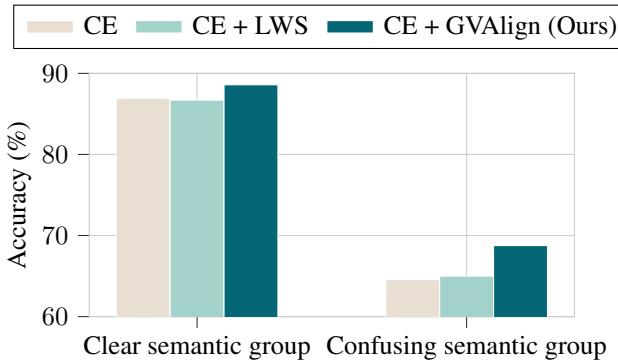


Figure 6. Shows the average accuracy of clear and confusing semantic groups.

6.3. Separating Semantic Similar Classes

The difference in performance among different training classes depends not only upon the characteristics of the training data (like number of examples per class), but also on the classes themselves. For example, few classes maybe semantically very close [9] and thus inherently confusing, which can often lead to reduced performance [5,36], even with the same number of training data per class. The proposed framework, though developed primarily for long-tailed setting, can also seamlessly account for these other challenges, since it tries to push the classifiers apart in the second stage. To verify this, we divide the total 50 base classes into two groups: (i) clear semantic group, where 25 classes are well-separated from the rest, and (ii) confusing semantic group, where the performance of these 25 classes is adversely affected due to confusion with other classes. This grouping is based on the sorting order of individual class performances. Figure 6 illustrates the average accuracy of these two groups. We observe that our proposed approach significantly improves performance, particularly for the confusing semantic group, justifying its effectiveness.

6.4. Ablation on Proposed Losses

Table 3 presents an analysis of the individual components of our proposed methodology. Clearly, the incorporation of losses at different stages contributes to the improvement of feature representations and the alignment of classifiers, leading to an enhancement in overall performance.

Distribution type →	Ordered long-tail		Shuffled long-tail	
	5 tasks	10 tasks	5 tasks	10 tasks
UCIR	✓	✗	✗	42.69
\mathcal{L}_{mix}	✓	✓	✗	45.31
\mathcal{L}_{ca}	✓	✓	✓	47.13
5 tasks	42.69	42.15	35.09	34.59
10 tasks	44.84	39.11	38.55	
			46.82	42.80
				41.64

Table 3. Presents an ablation study showcasing the impact of introducing different losses in our proposed methodology.

Conclusion

In conclusion, this paper presents a significant advancement in addressing the challenges of long-tail class incremental learning through a novel two-stage framework. Our approach excels in both learning new classes progressively and mitigating catastrophic forgetting in the presence of imbalanced data distributions. By incorporating robust feature learning in the first stage and harnessing the power of global variance as an informative measure in the second stage, we achieve effective classifier alignment without resorting to data balancing or additional layer tuning. Extensive experimental validation on various datasets corroborates the superiority of our approach compared to SOTA methods in various long-tail class incremental learning scenarios.

Acknowledgements: This work is partly supported through a research grant from SERB, Department of Science and Technology, Govt. of India (SPF/2021/000118).

References

[1] Davide Abati, Jakub Tomczak, Tijmen Blankevoort, Simone Calderara, Rita Cucchiara, and Babak Ehteshami Bejnordi. Conditional channel gated networks for task-aware continual learning. In *CVPR*, pages 3931–3940, 2020. 3

[2] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *ECCV*, pages 139–154, 2018. 3

[3] Franz Aurenhammer. Voronoi diagrams—a survey of a fundamental geometric data structure. *ACM Computing Surveys (CSUR)*, 23(3):345–405, 1991. 7

[4] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority oversampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002. 3

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, pages 1597–1607. PMLR, 2020. 8

[6] Hsin-Ping Chou, Shih-Chieh Chang, Jia-Yu Pan, Wei Wei, and Da-Cheng Juan. Remix: rebalanced mixup. In *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16*, pages 95–110. Springer, 2020. 3

[7] Peng Chu, Xiao Bian, Shaopeng Liu, and Haibin Ling. Feature space augmentation for long-tailed data. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIX 16*, pages 694–710. Springer, 2020. 3

[8] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *CVPR*, pages 9268–9277, 2019. 3

[9] Thomas Deselaers and Vittorio Ferrari. Visual and semantic similarity in imagenet. In *CVPR*, pages 1777–1784, 2011. 8

[10] Prithviraj Dhar, Rajat Vikram Singh, Kuan-Chuan Peng, Ziyan Wu, and Rama Chellappa. Learning without memorizing. In *CVPR*, pages 5138–5146, 2019. 3

[11] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *ECCV*, pages 86–102, 2020. 2, 3, 4

[12] Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. Borderline-smote: a new over-sampling method in imbalanced data sets learning. In *International conference on intelligent computing*, pages 878–887. Springer, 2005. 3

[13] Haibo He, Yang Bai, Edwardo A Garcia, and Shutao Li. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*, pages 1322–1328. Ieee, 2008. 3

[14] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *NeurIPS Workshop*, 2014. 3

[15] Youngkyu Hong, Seungju Han, Kwanghee Choi, Seokjun Seo, Beomsu Kim, and Buru Chang. Disentangling label distribution for long-tailed visual recognition. In *CVPR*, pages 6626–6636, 2021. 3

[16] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *CVPR*, pages 831–839, 2019. 2, 3, 4

[17] Chen Huang, Yining Li, Chen Change Loy, and Xiaou Tang. Learning deep representation for imbalanced classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5375–5384, 2016. 3

[18] Steven CY Hung, Cheng-Hao Tu, Cheng-En Wu, Chien-Hung Chen, Yi-Ming Chan, and Chu-Song Chen. Compact-ing, picking and growing for unforgetting continual learning. *arXiv preprint arXiv:1910.06562*, 2019. 3

[19] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. *ICLR*, 2020. 3, 4

[20] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017. 3

[21] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Master’s thesis, Department of Computer Science, University of Toronto*, 2009. 2

[22] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *NeurIPS*, 25:1097–1105, 2012. 1, 6

[23] Zhihong Li and Derek Hoiem. Learning without forgetting. *IEEE TPAMI*, 40(12):2935–2947, 2017. 2, 3

[24] Hong Liu, Jeff Z HaoChen, Adrien Gaidon, and Tengyu Ma. Self-supervised learning is more robust to dataset imbalance. *ICLR*, 2022. 1

[25] Xialei Liu, Yu-Song Hu, Xu-Sheng Cao, Andrew D Bagdanov, Ke Li, and Ming-Ming Cheng. Long-tailed class incremental learning. In *ECCV*, pages 495–512. Springer, 2022. 1, 2, 3, 4, 6

[26] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *CVPR*, pages 2537–2546, 2019. 1

[27] David Lopez-Paz and Marc’Aurelio Ranzato. Gradient episodic memory for continual learning. *NeurIPS*, 30, 2017. 6

[28] Arun Mallya, Dillon Davis, and Svetlana Lazebnik. Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In *ECCV*, pages 67–82, 2018. 3

[29] Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *CVPR*, pages 7765–7773, 2018. 3

[30] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *CVPR*, pages 1520–1528, 2015. 1

[31] Wanli Ouyang, Xingyu Zeng, Xiaogang Wang, Shi Qiu, Ping Luo, Yonglong Tian, Hongsheng Li, Shuo Yang, Zhe Wang, Hongyang Li, et al. Deepid-net: Object detection with deformable part based convolutional neural networks. *IEEE TPAMI*, 39(7):1320–1334, 2016. 1

[32] Inyoung Paik, Sangjun Oh, Taeyeong Kwak, and Injung Kim. Overcoming catastrophic forgetting by neuron-level plasticity control. In *AAAI*, volume 34, pages 5339–5346, 2020. 3

[33] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *CVPR*, pages 2001–2010, 2017. 2, 3, 5, 6

[34] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016. 3

[35] Jingru Tan, Changbao Wang, Buyu Li, Quanquan Li, Wanli Ouyang, Changqing Yin, and Junjie Yan. Equalization loss for long-tailed object recognition. In *CVPR*, pages 11662–11671, 2020. 3

[36] Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Ying Wu. Learning fine-grained image similarity with deep ranking. In *CVPR*, pages 1386–1393, 2014. 8

[37] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Learning to model the tail. *NeurIPS*, 30, 2017. 1

[38] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *CVPR*, pages 374–382, 2019. 3

[39] Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. *arXiv preprint arXiv:1708.01547*, 2017. 3

[40] Lu Yu, Bartłomiej Twardowski, Xiaolei Liu, Luis Herranz, Kai Wang, Yongmei Cheng, Shangling Jui, and Joost van de Weijer. Semantic drift compensation for class-incremental learning. In *CVPR*, pages 6982–6991, 2020. 3

[41] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *ICML*, pages 3987–3995. PMLR, 2017. 3

[42] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 4

[43] Songyang Zhang, Zeming Li, Shipeng Yan, Xuming He, and Jian Sun. Distribution alignment: A unified framework for long-tail visual recognition. In *CVPR*, pages 2361–2370, 2021. 3, 4

[44] Yifan Zhang, Bingyi Kang, Bryan Hooi, Shuicheng Yan, and Jiashi Feng. Deep long-tailed learning: A survey. *IEEE TPAMI*, 2023. 1

[45] Zhisheng Zhong, Jiequan Cui, Shu Liu, and Jiaya Jia. Improving calibration for long-tailed recognition. In *CVPR*, pages 16489–16498, 2021. 3, 4

[46] Fei Zhu, Zhen Cheng, Xu-yao Zhang, and Cheng-lin Liu. Class-incremental learning via dual augmentation. *NeurIPS*, 34, 2021. 3

[47] Fei Zhu, Xu-Yao Zhang, Chuang Wang, Fei Yin, and Cheng-Lin Liu. Prototype augmentation and self-supervision for incremental learning. In *CVPR*, pages 5871–5880, 2021. 3