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, 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-
categories 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 CIFAR100 [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 (Global Variance-Driven Classifier Alignment). 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-
makes the following contributions:

- We introduce a novel two-stage approach, termed GVAlign (Global Variance-Driven Classifier Alignment) encompassing robust feature learning in the first stage and classifier alignment in the second stage using global variance as an 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 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 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 Re-weighting [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 GVAlign 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,
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 $\{D^{(0)}, D^{(1)}, \ldots, D^{(T)}\}$, and the corresponding classes set represented by $\{C^{(0)}, C^{(1)}, \ldots, C^{(T)}\}$. One assumption in CIL is there are no common classes across different tasks i.e. $C^{(k)} \cap C^{(l)} = \emptyset$ when $k \neq l$. At each task $t$, the model has access to data $D^{(t)} = \{x_i, y_i\}_{i=1}^{n^{(t)}}$, where $n^{(t)}$ represents the number of samples in $D^{(t)}$, and $y_i \in C^{(t)}$. In CIL, the number of samples for each class is the same and equal to $\frac{n^{(t)}}{|C^{(t)}|}$; however, in the case of long-tail CIL, the data distribution of $D^{(t)}$ adheres to a long-tail distribution. In both CIL and long-tail CIL, during training task $t$, alongside $D^{(t)}$, the model has access to an exemplar bank $E$ comprising a limited number of samples from earlier classes $C^{(0:t-1)}$ and the end goal after task $t$ is to classify all the classes seen so far i.e. $C^{(0:t)}$. The addition of exemplar bank $E$ creates more imbalance in training data and makes long-tail CIL more challenging.

The model that learns sequentially is denoted as $\Theta = \{F_\theta, \psi\}$, where $F_\theta$ represents the feature extractor with parameters $\theta$, and classifiers are represented by $\psi$. 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 $\{D^{(t)} \cup E\}$, at the end of task $t$, the trained model $\Theta^{(t)} = \{F_\theta, \psi^{(0:t)}\}$ is able to classify all classes from $C^{(0:t)}$.

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 $L_{inc}$. 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 $L_{inc}$, 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 $D^{(t)}$, 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, \( \lambda \) is drawn from a Beta distribution, i.e., \( \lambda \sim Beta(1, 1) \). The mixup loss is then calculated as \( \mathcal{L}_{\text{mix}} = \mathcal{L}_{CE}(\tilde{x}, \tilde{y}) \), 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

\[ \mathcal{L}_{\text{stage1}} = \mathcal{L}_{\text{inc}} + \mathcal{L}_{\text{mix}} \quad (3) \]

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:

\[ P_k = \frac{1}{N_k} \sum_{(x, y) \in (D^{(i)} \cup \mathcal{E})} I(y = k) \mathcal{F}_\theta(x) \quad (4) \]

where \( N_k \) represents the number of samples in \( k \)th class, and the indicator variable \( I(y = k) \) 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 \( \Sigma_G \) 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 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 \( \Sigma_G \) 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.

---

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

**Input:** \( \Theta = \{ \mathcal{F}_\theta, \psi \} \leftarrow \text{Model}; \)

\{\( D^{(0)}, D^{(1)}, ..., D^{(T)} \)\} \leftarrow \text{Data stream};

\( e_1 \leftarrow \text{No. of epochs in stage 1}; \)

\( e_2 \leftarrow \text{No. of epochs in stage 2}; \)

\( \mathcal{E} = \{ \} \leftarrow \text{Empty exemplar buffer}; \)

**for** \( t \leftarrow 0 \) to \( T \) **do**

**for** \( e \leftarrow 1 \) to \( e_1 \) **do**

**for** \( \in \mathcal{D}^{(t)} \cup \mathcal{E} \) **do**

\( \mathcal{B} = \text{SampleMiniBatch}(\mathcal{D}^{(t)} \cup \mathcal{E}); \)

\( \mathcal{O}^{(0:t)} = \psi^{(0:t)}(\mathcal{F}_\theta(\mathcal{B})); \)

\( \mathcal{L}_{inc} = \text{IncrementalLoss}(\mathcal{B}, \mathcal{O}^{(0:t)}); \)

\( \mathcal{L}_{mix} = \text{MixUpLoss}(\mathcal{B}); \)

\( \Theta \leftarrow \text{UpdateParameters}(\mathcal{L}_{inc} + \mathcal{L}_{mix}); \)

\( \mathcal{P} \leftarrow \text{CalculatePrototypes}(\mathcal{D}^{(t)} \cup \mathcal{E}); \)

**if** \( t=0 \) **then**

\( \Sigma_G = \text{GlobalVarance}(\mathcal{D}^{(0)}); \)

**for** \( e \leftarrow 1 \) to \( e_2 \) **do**

\( \mathcal{P}' \leftarrow \text{SampleProtoFromGlobalVar}(\mathcal{P}, \Sigma_G); \)

\( \mathcal{O}^{(0:t)} = \psi^{(0:t)}(\mathcal{P}'); \)

\( \mathcal{L}_{ca} = \text{ClassifierAlignLoss}(\mathcal{O}^{(0:t)}); \)

\( \psi^{(0:t)} \leftarrow \text{UpdateParameters}(\mathcal{L}_{ca}); \)

\( \mathcal{E} \leftarrow \text{UpdateExemplars}(\mathcal{D}^{(t)}); \)

**return** \( \Theta; \)

---

36
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].

**CIFAR100:** 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 \( t \in \{0, 1, \ldots, T\} \). We define \( Acc_{0:t} \) as the model’s accuracy on the test data of all tasks from 0 to \( t \) after learning task \( t \), where \( n \leq t \). Consequently, upon completion of task \( T \), the average incremental accuracy is computed as \( \frac{1}{T} \sum_{t=0}^{T} Acc_{0:t} \). 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 250\(^{th}\), 350\(^{th}\), and 450\(^{th}\) 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 30\(^{th}\) and 60\(^{th}\) 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.

<table>
<thead>
<tr>
<th>Long-tail distribution type</th>
<th>Ordered long-tail</th>
<th>Shuffled long-tail</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CIFAR-100</td>
<td>ImageNet-Subset*</td>
</tr>
<tr>
<td></td>
<td>5 tasks</td>
<td>10 tasks</td>
</tr>
<tr>
<td>UCIR</td>
<td>42.69</td>
<td>42.15</td>
</tr>
<tr>
<td>UCIR + LWS (ECCV 2022)</td>
<td>45.88</td>
<td>45.73</td>
</tr>
<tr>
<td>UCIR + GVAAlign (Ours)</td>
<td>47.13</td>
<td>46.82</td>
</tr>
<tr>
<td>PODNET</td>
<td>44.07</td>
<td>43.96</td>
</tr>
<tr>
<td>PODNET + LWS (ECCV 2022)</td>
<td>44.38</td>
<td>44.35</td>
</tr>
<tr>
<td>PODNET + GVAAlign (Ours)</td>
<td>48.41</td>
<td>47.71</td>
</tr>
</tbody>
</table>

Table 1. Experimental results on long-tail class incremental learning (* signifies that we have rerun all experiments on the ImageNet-Subset 100 dataset. Comprehensive dataset details and data can be found in the GitHub repository for reproducibility).
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 GVAlign 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. The alignment of classifiers results in a tangible enhancement in accuracy for the newly introduced classes. Specifically, in the \( T=5 \) scenario, the classification accuracy on task 1 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.
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.

### 6.4. Ablation on Proposed Losses

<table>
<thead>
<tr>
<th>Distribution type</th>
<th>Ordered long-tail</th>
<th>Shuffled long-tail</th>
</tr>
</thead>
<tbody>
<tr>
<td>UCIR $\mathcal{L}<em>{mix}$ $\mathcal{L}</em>{con}$</td>
<td>5 tasks 10 tasks</td>
<td>5 tasks 10 tasks</td>
</tr>
<tr>
<td>✓ × ×</td>
<td>42.69 42.15</td>
<td>35.09 34.59</td>
</tr>
<tr>
<td>✓ ✓ ×</td>
<td>45.31 44.84</td>
<td>39.11 38.55</td>
</tr>
<tr>
<td>✓ ✓ ✓</td>
<td>47.13 46.82</td>
<td>42.80 41.64</td>
</tr>
</tbody>
</table>

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


[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


