

Long-Tailed Class Incremental Learning

Xialei Liu^{1,*}, †, Yu-Song Hu^{1,*}, Xu-Sheng Cao¹, Andrew D. Bagdanov², Ke Li³,
and Ming-Ming Cheng¹

TMCC, CS, Nankai University, China¹; MICC, University of Florence, Florence, Italy²
Tencent Youtu Lab³

Abstract. In class incremental learning (CIL) a model must learn new classes in a sequential manner without forgetting old ones. However, conventional CIL methods consider a balanced distribution for each new task, which ignores the prevalence of long-tailed distributions in the real world. In this work we propose two long-tailed CIL scenarios, which we term *ordered* and *shuffled* LT-CIL. *Ordered* LT-CIL considers the scenario where we learn from head classes collected with more samples than tail classes which have few. *Shuffled* LT-CIL, on the other hand, assumes a completely random long-tailed distribution for each task. We systematically evaluate existing methods in both LT-CIL scenarios and demonstrate very different behaviors compared to conventional CIL scenarios. Additionally, we propose a two-stage learning baseline with a learnable weight scaling layer for reducing the bias caused by long-tailed distribution in LT-CIL and which in turn also improves the performance of conventional CIL due to the limited exemplars. Our results demonstrate the superior performance (up to 6.44 points in average incremental accuracy) of our approach on CIFAR-100 and ImageNet-Subset. The code is available at <https://github.com/xialeiliu/Long-Tailed-CIL>.

1 Introduction

Deep neural networks have achieved spectacular success in many computer vision tasks. In general, most tasks assume a static world in which all data is available for training in a single learning session. The world is ever-changing, however, and future intelligent systems will have to master new tasks and adapt to new environments without forgetting previously acquired knowledge. Incremental learning, also known as continual or lifelong learning, is the paradigm of continually learning a sequence of tasks as new data becomes available [5,9,27,29]. The biggest challenge in incremental learning is avoiding *catastrophic forgetting* [28] when learning with only current task data and possibly a small memory of data from previous tasks.

Class incremental learning (CIL) considers a scenario in which no task boundary is provided during inference, which is significantly more challenging than task incremental learning with a known task boundary [38].

† Corresponding author (xialei@nankai.edu.cn)

* The first two authors contribute equally.

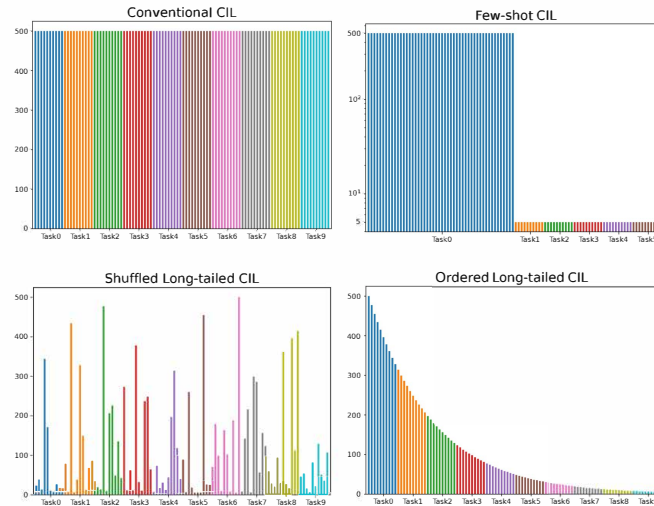


Fig. 1: An illustration of our proposed long-tailed CIL (LT-CIL) scenarios compared to conventional CIL [27] with balanced distribution and few-shot CIL [37].

Although conventional CIL has seen significant progress, it assumes that the data is sampled from a balanced distribution. However, data sampled from the real world often follows a long-tailed distribution [14,24,40,49] in which some classes have many more samples than others. Learning from long-tailed distributions has been approached by re-sampling [11,15] or re-weighting [21,36,48] head classes and tail classes to learn a balanced classifier. Recently, transfer learning [24,44] between head and tail classes, two-stage learning [16,50] to decouple representation and classifier learning, and ensemble learning [42,39] of different experts have achieved superior performance. However, all these works consider a static world in which all data is immediately available for training. It is not straightforward to extend these methods with new classes without suffering catastrophic forgetting.

Due to the long-tailed and incremental nature of real-world learning problems, it is crucial to investigate the class incremental learning in the more realistic scenario with long-tailed distribution for different tasks. In this work, we propose two new scenarios based on long-tailed distributions: *Ordered* Long-tailed CIL (LT-CIL) and *Shuffled* Long-tailed CIL. As shown in Fig. 1, *Ordered* LT-CIL considers a scenario in which all classes are ordered according to the number of samples per class and then they are divided into different tasks. In contrast, *Shuffled* LT-CIL assumes that classes appearing in different tasks are random and each task may have varying degrees of imbalance to their distributions. Compared to current common CIL and few-shot CIL scenarios, LT-CIL considers more natural data distributions from the real-world.

We compare existing state-of-the-art CIL algorithms on these new LT-CIL scenarios, and our results show that they all perform much worse under long-tailed distributions. They are also less robust to different datasets and less consistent with increasing number of exemplars compared to conventional CIL. Therefore, we propose a two-stage strategy with a learnable weight scaling layer for LT-CIL to boost the performance of existing methods. As an extra bonus, we find that the two-stage strategy can also help on the conventional CIL scenario, where a limited memory for previous data and the large amount of current data can cause unbalanced data distribution as well. Importantly, our two-stage approach can be integrated into any CIL method.

The main contributions of this paper are:

- we propose two new CIL scenarios (Ordered and Shuffled LT-CIL) that consider long-tailed class distributions more common in the real world;
- we evaluate conventional CIL algorithms comprehensively and report several findings in these two new long-tailed scenarios;
- we design a two-stage training strategy with a learnable weight scaling layer for LT-CIL scenarios which is complementary to existing CIL methods and show that it improves conventional CIL and both LT-CIL scenarios on CIFAR-100 and ImageNet.

2 Related Work

Here we review recent work from the literature on class incremental and long-tailed learning most relevant to our proposed approach.

2.1 Class incremental learning

Class incremental learning (CIL) is one of the primary scenarios for continual learning [38]. There are three main approaches to tackling this problem: regularization-based methods, parameter-isolation methods, and replay-based methods. Elastic Weight Consolidation (EWC) is a popular regularization-based method which identifies which parameters are more important for previous tasks and updating these less during learning of new tasks [18]. R-EWC [22], Synaptic Intelligence (SI) [45], and Memory Aware Synapses (MAS) [3] adopt the same strategy but with different techniques to identify important weights. Learning without Forgetting (LwF) is a widely-used baseline that uses knowledge distillation technique to constrain the output probabilities of new tasks [20].

Parameter-isolation methods increase model plasticity by adding more neurons, modules [31,33], branches [23] or masks [26,25,34]. Dynamically Expandable Networks (DEN) [19] performs selective retraining and dynamically expands network capacity, while Dark Experience Replay (DER) [43] dynamically expands the representation by freezing the previously-learned representation and augmenting it with additional feature dimensions from a new learnable feature extractor.

Replay-based methods are very effective and recall knowledge from previous tasks by maintaining a small memory of samples [2,4,7,13,32,41], representations [12], or synthetic data [35]. Incremental Classifier and Representation Learning (iCaRL) stores a fixed budget of exemplars to train and construct class means for classification [32]. Pooled Output Distillation (PODNET) applies various pooling operations to intermediate features to distill knowledge from past tasks [10].

These conventional CIL approaches all implicitly assume a balanced label distribution for each task. Recently, Kim et al. [17] proposed a multi-label classification problem with long-tailed distribution. Abdelsalam et al. [1] proposed another realistic CIL setting in which each class can have two granularity levels: each sample could have a high-level (coarse) label and a low-level (fine) label, but only one label is available for each task. In contrast to these works, we are interested in more realistic scenarios for CIL with long-tailed class distributions. We provide a comprehensive experimental evaluation of state-of-the-art CIL methods on such settings. Additionally, we propose a two-stage framework with a learnable weight scaling layer to further reduce the bias problem caused long-tailed distribution.

2.2 Long-tailed learning

The long-tailed learning problem has been comprehensively studied given the prevalence of the data imbalance problem in the real world [14,40,49,24]. Most previous works address this problem by re-sampling [11,15], re-weighting [21,36,48] or transfer learning [24,44]. Re-sampling methods can over-sample the tail classes or under-sample the head classes. Re-weighting methods assign different weights to different classes or instances. Transfer learning aims to fuse knowledge between head and tail classes. Data augmentation is another way to increase the tail distribution [30]. Bi-lateral Branch Networks (BBNs) [51] use two network branches, a conventional learning branch and a re-balancing branch, to address the long-tailed recognition problem [51]. Learning from Multiple Experts (LFME) [42] and Routing Diverse Experts (RIDE) [39] adopt the same idea of ensemble learning to aggregate knowledge from multiple experts.

Recently, a two-step training method decoupled the representation learning and classifier learning, achieving superior performance compared to previous methods [16]. Mixup Shifted Label-Aware Smoothing (MiSLAS) [50] uses a regularization technique mixup [46] to further improve in a two-stage framework. Different from these works addressing the long-tailed learning problem in a static world, we propose class incremental learning scenarios with long-tailed distributions. This requires continually learning different long-tailed classes in a dynamic world without catastrophic forgetting.

3 A Two-stage Approach to LT-CIL

In this section we first formulate two long-tailed CIL scenarios and then we adapt several existing state-of-the-art methods for conventional CIL to long-tailed CIL

scenarios. Finally, we propose a two-stage training method with a learnable weight scaling layer for long-tailed CIL.

3.1 Long-tailed CIL

In conventional CIL the model must sequentially learn different tasks where each new task consists of a set of new classes. Formally, for each training task t , the data is denoted as \mathcal{D}_t , where $\mathcal{D}_t = (\mathbf{x}_t^{(i)}, y_t^{(i)})_{i=1}^{n_t}$, and $\mathbf{x}_t^{(i)}$ is an input image, $y_t^{(i)}$ is the corresponding label and there are n_t samples in total at task t . Normally the number of samples per class is equally distributed and it can be calculated as $\frac{n_t}{C_t}$, in which C_t is the number of classes at task t . Therefore, the total number of classes learned up to current tasks is $C_{1:t}$. For replay-based methods, at each training task a memory of \mathcal{M} (known as the memory of exemplars) is stored for previous classes up to task $t - 1$, normally $\lfloor \frac{|\mathcal{M}|}{C_{1:t-1}} \rfloor$ samples stored for each class.

While class-incremental learning has many practical applications, the assumption of equally distributed samples is not always realistic. Most real-world class distributions are in face *long-tailed*. The long-tailed distribution follows an exponential decay in sample sizes across classes as described in [6]. This decay is parameterized by ρ which is the ratio between the most and least frequent classes. An example of different ratios can be found in Fig. 2, the larger the ratio, the more balanced the distribution. $\rho = 1$ is the conventional CIL case and ρ in $(0,1)$ indicates different degrees of long-tailed distribution.

Given a sampled long-tailed class distribution, we propose two long-tailed CIL scenarios constructed from it:

- **Ordered LT-CIL** which starts learning from the most frequent classes as first task and ends with the last task containing the least frequent classes; and
- **Shuffled LT-CIL** which first shuffles the long-tailed distribution randomly, and then constructs different tasks based on the Shuffled class order. It thus has varying degrees of imbalance in each task.

In both conventional and long-tailed CIL the test set contains uniformly distributed samples for each class. Ordered LT-CIL is representative of real world applications where we are able to learn from easy-to-sample classes first, and then gradually increase the difficulty of learning with less frequent samples. Shuffled LT-CIL, on the other hand, considers a more general scenario without any assumptions of the arriving data distribution.

3.2 Conventional CIL methods applied to LT-CIL

A classification model can be roughly divided into two parts: a feature extractor f_θ , usually a convolutional neural network with parameters θ , and a classification head h_ϕ with parameters ϕ . The cross entropy loss \mathcal{L}_{CE} at task t is defined as:

$$\mathcal{L}_{CE,t}(\mathbf{x}, y; \theta_t, \phi_t) = -\frac{1}{|\mathcal{D}_t| + |\mathcal{M}|} \sum_{(\mathbf{x}, y) \in \mathcal{D}_t \cup \mathcal{M}} \mathbf{y} \cdot \log(\mathbf{p}_{1:t}(\mathbf{x})), \quad (1)$$

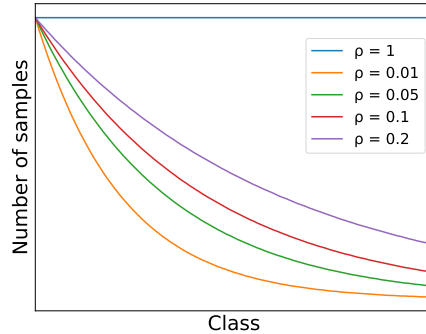


Fig. 2: An illustration imbalance ratios ρ for long-tailed distribution generation. $\rho = 1$ is the balanced distribution and corresponds to conventional CIL.

where \mathbf{y} is a one-hot vector with 1 at the position of the correct class, and $\mathbf{p}_{1:t}(\mathbf{x})$ is a vector containing the probability predictions for image \mathbf{x} over all classes up to task t .

Directly minimizing Eq. 1, even if replaying past-task exemplars sampled from \mathcal{M} , can result in catastrophic forgetting. Normally, a method-specific auxiliary loss L_{aux} is added to mitigate forgetting using regularization or by replaying past-task exemplars from \mathcal{M} . Examples of such an auxiliary loss include the many techniques based on knowledge distillation [10,13,20]. The total loss L_t for task t is thus:

$$L_t = L_{\text{CE},t} + L_{\text{aux},t} \quad (2)$$

where L_{aux} is the method-specific loss. In Section 4 we evaluate the LwF [20], EEIL [7], LUCIR [13] and PODNET [10] methods on our Ordered and Shuffled LT-CIL scenarios. These methods can be directly applied to both Ordered LT-CIL and Shuffled LT-CIL scenarios, but since they are not specifically designed for LT-CIL we are interested in how they perform in these more challenging scenarios.

3.3 A two-stage method with a learnable weight scaling layer

Two-stage methods have shown state-of-the-art performance in long-tailed recognition [8,16,47,50]. In general, two-stage learning decouples representation learning from classifier learning:

- In the **first stage**, it aims to learn a better feature extractor f_θ using an instance-balanced sampler (also known as random sampler where each data point has the *same probability* of being sampled) that generalizes well;
- In the **second stage**, a class-balanced sampler in which *each class* is sampled uniformly first and then each instance is sampled uniformly within it (also known as balanced sampler) is used to retrain the classifier h_ϕ to obtain better classification accuracy.

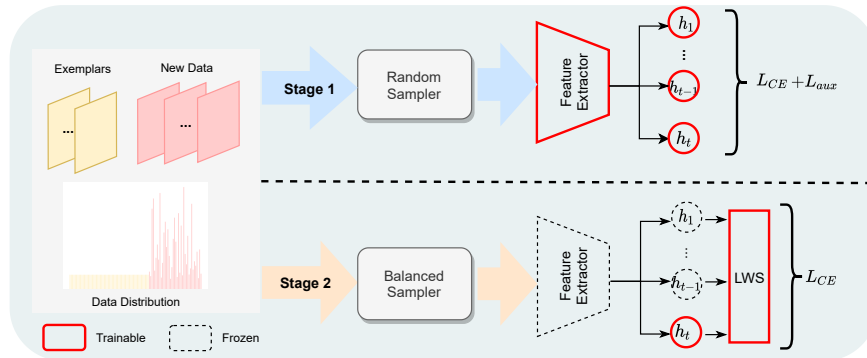


Fig. 3: An overview of our two-stage method with a learnable weight scaling layer for LT-CIL. Note that new data for the current task follows a long-tailed distribution and the memory contains a few samples from previous tasks. In the first stage, random sampling is used to learn a better feature extractor together with L_{CE} and method-specific loss L_{aux} to reduce forgetting. In the second stage, a balanced sampler is used to learn a balanced classifier together with a layer of learnable scaling weights (LWS). To reduce representation drift for future tasks, we fix the previous classifiers $h_{1:t-1}$ and only train the current head and LWS with cross entropy loss L_{CE} .

In LT-CIL, for task t we first learn a model from the first stage (seen in the top part of Fig. 3) using Eq. 2 with both cross entropy loss $L_{CE,t}$ and method-specific loss $L_{aux,t}$. In the second stage, we attach a single trainable layer, which we call a Learnable Weight Scaling (LWS) layer, $\mathbf{W} \in \mathbb{R}^{C_{1:t} \times 1}$ with dimension equal to the number of classes $C_{1:t}$ in the output of classifier $h_{1:t}$ at task t (as shown in the bottom part of Fig. 3).

The LWS is used to balance between classes with different numbers of samples in the long-tailed distribution. The final output of the model $\hat{\mathbf{z}}$ is calculated using an element-wise product of the classifier output with the LWS:

$$\hat{\mathbf{z}} = \mathbf{W} \odot h_{\phi_{1:t}}(f_{\theta_t}(\mathbf{x})) \quad (3)$$

We found it to be essential to fix previous head $h_{\phi_{1:t-1}}$ at the second stage when learning together with LWS layer, otherwise the modified $h_{\phi_{1:t-1}}$ can back-propagate in the future tasks and damage representation learning in the first stage. Note that only L_{CE} loss is used in the second stage no matter which loss is chosen for L_{aux} , and that the LWS layer \mathbf{W} is applied only in the second stage for training and evaluation. The scaling layer for task t is discarded in the first stage training for the next task $t + 1$.

Discussion. Conventional long-tailed learning considers a fixed set of classes, therefore it is challenging to apply two-stage methods directly to incremental learning for LT-CIL. In the first stage, a good representation must be learned and catastrophic forgetting avoided. Whereas in the second stage, the classifier

learned from balanced sampling will be the initialization for the future tasks. Thus, the modified classifier can lead to representation drift back-propagated to future tasks, which harms the generalization of feature representation in the dynamic and incremental process of continual learning.

4 Experimental Results

In this section we first introduce the experimental setup and then compare different existing CIL methods applied to LT-CIL benchmarks. Finally we evaluate our two-stage method and conduct ablation study of key elements.

4.1 Experimental setup

Implementation details. We experiment on two datasets: CIFAR-100 and ImageNet-Subset with 100 classes. We use the publicly available implementations of existing CIL methods in the framework FACIL [27] and implement our two-stage algorithm with long-tailed data loader in the same framework for fair comparison. We follow LUCIR and PODNET by starting with a large first task with half of the classes in each dataset and equally dividing the remaining classes in subsequent tasks.

We use ResNet-32 for CIFAR-100 ResNet-18 for ImageNet-Subset. We use an initial learning rate of 0.1, and divide it by 10 after 80 and 120 epochs (160 epochs in total) for CIFAR-100. ImageNet-Subset, the learning rate starts from 0.1 and is divided by 10 after 30 and 60 epochs (90 epochs in total). The batch size is 128 for all experiments. For stage two training, the learning rate is set to 0.1 and we train it for 30 epochs.

Evaluation protocols. We use average accuracy over all classes and average incremental accuracy over all tasks as evaluation metrics. We first evaluate different methods on LT-CIL scenarios with an imbalance ratio $\rho = 0.01$ and 20 exemplars per class, and report varying imbalance ratios and number of exemplars in Section 4.4.

4.2 Conventional methods on LT-CIL scenarios

In this section we analyze the performance of four popular CIL methods: LwF [20] with exemplars, EEIL [7], LUCIR [13], and PODNET [10]). In Fig. 4(a) we first evaluate on conventional CIL as a reference. We then evaluate on the Ordered LT-CIL setting in Fig. 4(b). It is clear that LT-CIL is a more challenging scenario given that joint training drops from 68.64 to 36.94. Interestingly, LUCIR with nearest class mean (NCM) classifier obtains the best performance in average incremental accuracy. LUCIR is much better than PODNET for the tail classes with few samples in the end of training, except for the last task.

Similarly, as seen in Fig. 4(c), the overall performance on Shuffled LT-CIL scenario for all methods is much worse than in the conventional CIL scenario. LUCIR with NCM classifier again achieves the best performance.

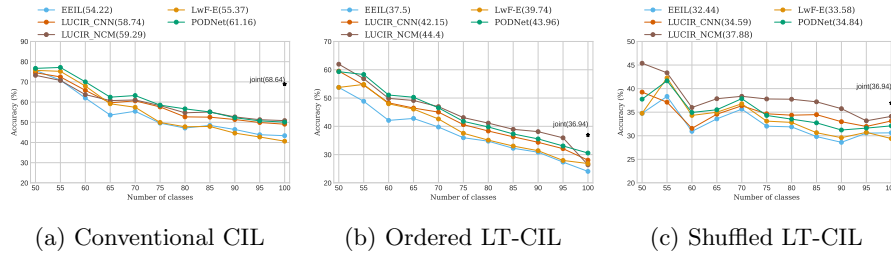


Fig. 4: Average accuracy for different scenarios on CIFAR-100. Average incremental accuracy is in the parentheses.

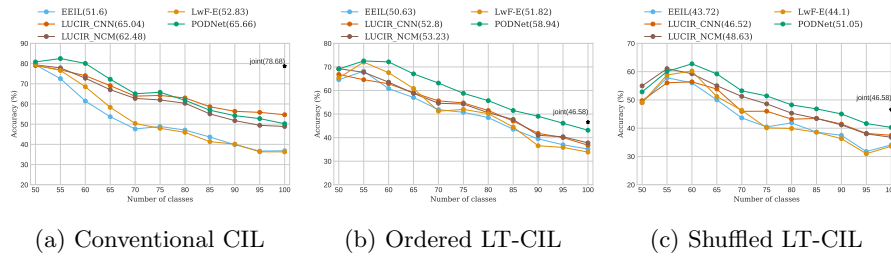


Fig. 5: Average accuracy for different scenarios on ImageNet-Subset. Average incremental accuracy is in parentheses.

In Fig. 5 we report on the same experiment performed on ImageNet-Subset. When we apply these methods to LT scenarios, PODNET achieves significantly better accuracy compared to other methods with average incremental accuracy of 58.94 and 51.05 for Ordered LT-CIL and Shuffled LT-CIL, respectively. For conventional CIL, methods often has similar rankings in terms of performance for different datasets as shown in Fig. 4 (a) and 5 (a). While for LT-CIL, we can see that LUCIR outperforms PODNET on CIFAR-100 but achieves worse results than PODNET on ImageNet-Subset. It suggests that these methods are not as robust as in the conventional CIL scenario.

4.3 Results for our two-stage method

Our method for LT-CIL. In Table 1 we integrate our proposed two-stage strategy into three existing methods: EEIL, LUCIR (with CNN classifier), and PODNET. In general, the two-stage strategy helps on all three methods in both 5- and 10-task settings. The improvement is especially noticeable in the Shuffled LT-CIL scenario. Specifically, for EEIL, our method only improves by a small margin on Ordered LT-CIL scenario but boosts significantly on Shuffled LT-CIL. It outperforms EEIL by 2.28 and 1.26 on CIFAR-100 when $T = 5$ and $T = 10$, respectively for Shuffled LT-CIL. The improvement is even larger

| Methods | CIFAR-100 | | ImageNet-Subset | | |
|-------------------------|------------|------------|-----------------|------------|------------|
| | 5 tasks | 10 tasks | 5 tasks | 10 tasks | |
| <i>Ordered LT-CIL</i> | EIL | 38.46 | 37.50 | 50.68 | 50.63 |
| | + (Ours) | 38.97+0.51 | 37.58+0.08 | 51.36+0.68 | 50.74+0.11 |
| | LUCIR | 42.69 | 42.15 | 52.91 | 52.80 |
| | + (Ours) | 45.88+3.19 | 45.73+3.58 | 54.22+1.31 | 55.41+2.61 |
| | PODNET | 44.07 | 43.96 | 58.78 | 58.94 |
| + (Ours) | 44.38+0.31 | 44.35+0.39 | 58.82+0.04 | 59.09+0.15 | |
| <i>Shuffled LT-CIL</i> | EIL | 31.91 | 32.44 | 42.87 | 43.72 |
| | + (Ours) | 34.19+2.28 | 33.70+1.26 | 49.31+6.44 | 48.26+4.54 |
| | LUCIR | 35.09 | 34.59 | 45.80 | 46.52 |
| | + (Ours) | 39.40+4.31 | 39.00+4.41 | 52.08+6.28 | 51.91+5.39 |
| | PODNET | 34.64 | 34.84 | 49.69 | 51.05 |
| + (Ours) | 36.37+1.73 | 37.03+2.19 | 51.55+1.86 | 52.60+1.55 | |
| <i>Conventional CIL</i> | EIL | 57.41 | 54.22 | 53.84 | 47.30 |
| | + (Ours) | 59.10+1.69 | 56.91+2.69 | 57.45+3.61 | 53.40+6.10 |
| | LUCIR | 61.15 | 58.74 | 67.21 | 65.04 |
| | + (Ours) | 63.48+2.33 | 60.57+1.83 | 68.82+1.61 | 67.44+2.40 |
| | PODNET | 63.15 | 61.16 | 70.13 | 65.66 |
| + (Ours) | 64.58+1.43 | 62.63+1.47 | 71.08+0.95 | 68.47+2.81 | |

Table 1: Comparison of average incremental accuracy on CIFAR-100 and ImageNet-Subset in the LT-CIL and conventional CIL scenarios.

on ImageNet-Subset with 6.44 and 4.54 improvement in absolute accuracy. For LUCIR, we see a consistent boost by adding our method, improving from 1.31 to 6.28 for CIFAR-100 and ImageNet-Subset, respectively. PODNET is the best baseline in most scenarios where we observe a smaller gain with our proposed method compared to LUCIR. Overall, PODNET and LUCIR with our method can achieve very competitive results, which improves the consistency for both Ordered LT-CIL and Shuffled LT-CIL.

Our method for conventional CIL. Surprisingly, as seen in Table 1, when we combine ours with existing methods the performance is improved not only in LT-CIL scenarios but also for conventional CIL. We believe this is due to the imbalance caused by limited memory for storing exemplars from previous tasks.

Results on real-world long-tailed dataset We experiment with 100 classes chosen from the iNaturalist dataset. We randomly chose 100 classes from the pantae super category and tested LUCIR and LUCIR+ with the data separated into 5 tasks with a base task of 50 classes. Results show that LUCIR can achieve an accuracy of 32.34%, and LUCIR+ with two stage training about 1.46% higher. iNaturalist is a real-world dataset with long-tailed distribution, and thus the value of ρ is undefined. We estimate it to be about 0.01. It shows how our method perform in real-world dataset under long-tailed distribution.

Further results on AANets [23] and DER [43] We report results for AANets (based on LUCIR) on Shuffle LT-CIL scenario (CIFAR-100 with 10-task setting). AANets outperforms LUCIR by a large margin achieving 38.53 in average incremental accuracy, and adding our method still improves over it by about 1%. We found that DER does not work well on long-tail scenarios (with only 29.54 in average accuracy), but our method improves it by about 4%.

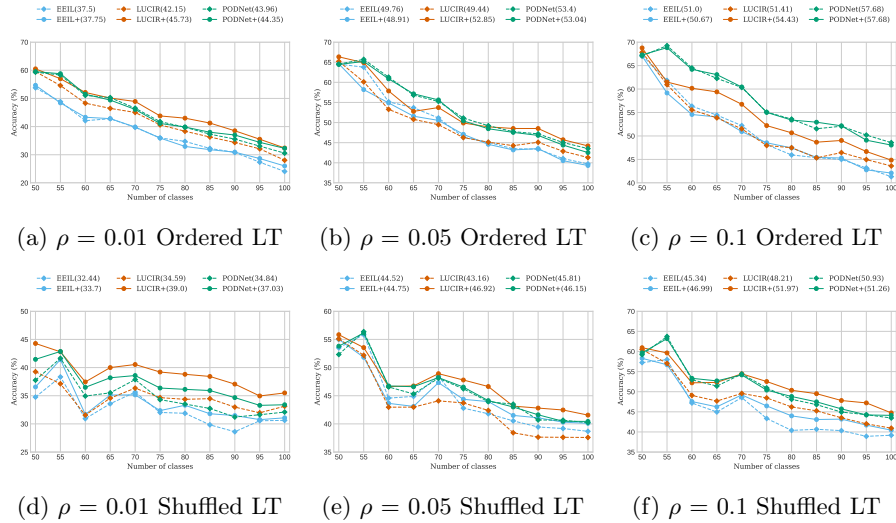


Fig. 6: Average accuracy on CIFAR-100 dataset for different imbalance ratios. The top row is for Ordered LT-CIL and the bottom row for Shuffled LT-CIL. The + suffix indicates our two-stage method applied to the corresponding baseline.

| Fix $h_{1:t-1}$ | LWS | Conventional | Ordered | Shuffled |
|-----------------|-----|--------------|--------------|--------------|
| | | 26.28 | 33.54 | 23.41 |
| ✓ | | 60.00 | 43.52 | 37.38 |
| | ✓ | 60.28 | 44.45 | 38.13 |
| ✓ | ✓ | 60.57 | 45.73 | 39.01 |

Table 2: Ablation study on effectiveness of different components. $h_{1:t-1}$ denotes the classification heads up to task $t - 1$.

4.4 Ablation study

Ablation on imbalance ratio. In this section we analyze three different imbalance ratios: $\rho = 0.01$, $\rho = 0.05$ and $\rho = 0.1$. The smaller the ratio, the more skewed the distribution. In Fig. 6 we give results for three different baselines (EEIL, LUCIR and PODNET) and our two-stage approach applied to them (EEIL+, LUCIR+ and PODNET+). As seen in Fig. 6(a-c), for the Ordered LT-CIL scenario PODNET surpasses LUCIR by a larger margin as imbalance ratio ρ increases. However, LUCIR obtains the best performance when $\rho = 0.01$ but is worse than PODNET by a large margin when $\rho = 0.1$. Overall, our two-stage method consistently boosts accuracy of most methods, especially for LUCIR+. For the Shuffled LT-CIL scenario in Fig. 6 (d-f), we see that PODNET outperforms LUCIR for all three ρ . The proposed two-stage method further improves performance, especially for LUCIR, resulting in LUCIR+ with the

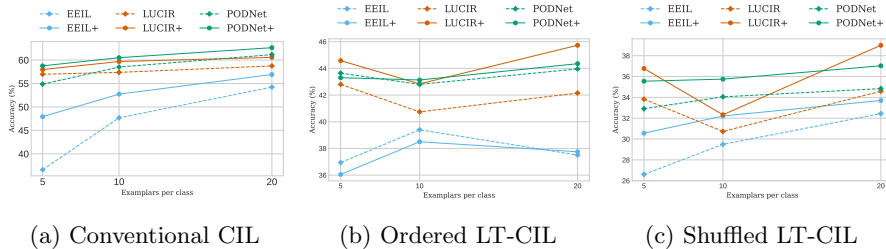


Fig. 7: Average incremental accuracy on CIFAR-100 for different scenarios as a function of stored exemplars. The + suffix indicates our two-stage method applied to the corresponding baseline.

| | EEIL | EEIL+ | LUCIR | LUCIR+ | PODNet | PODNet+ |
|--------------|-------|-------|-------|--------------|--------|--------------|
| Conventional | 51.07 | 53.24 | 52.02 | 56.25 | 58.42 | 60.47 |
| Ordered | 39.47 | 40.03 | 38.31 | 43.65 | 41.47 | 40.78 |
| Shuffled | 32.20 | 34.68 | 30.68 | 37.38 | 35.75 | 37.09 |

Table 3: Average accuracy on long sequences of 25 tasks for three different scenarios on CIFAR-100. The + suffix indicates our two-stage method applied to the corresponding baseline.

best overall performance. More results on ImageNet-Subset can be found in the supplementary material.

Ablation on exemplar memory size. We evaluate different methods with 5, 10, and 20 exemplars per class. As expected, we see in Fig. 7(a) that in the conventional CIL setting increasing exemplars results in better performance. However, for Ordered LT-CIL in Fig. 7(b) we see that LUCIR and our two-stage method LUCIR+ both drop when increasing from 5 to 10 exemplars, but recover with 20 exemplars. For both EEIL and EEIL+, the best performance is obtained with 10 exemplars, which may be due to the long-tailed distribution of the final tasks. Both PODNET and PODNET+ obtains better performance with more exemplars. Similarly, for Shuffled LT-CIL in Fig. 7(c) performance of both EEIL and PODNET increases with more exemplars, but LUCIR drops at 10 exemplars.

Effectiveness of different components. We ablate the two main components of fixing previous head $h_{1:t-1}$ until $t-1$ task and using LWS in the second stage. As seen from Table 2, without using either component the performance is very poor in all three scenarios. Both fixing the previous head $h_{1:t-1}$ and using LWS significantly boost accuracy, in particular for the conventional scenario which is up to 2.5 times higher. Using both components results in the best overall performance.

Long task sequences. In this experiment we evaluate on a longer sequence of 25 tasks in CIFAR-100 for all three scenarios. As we see in Table 3, our method improves over all baselines in this more challenging setting. LUCIR gains the most

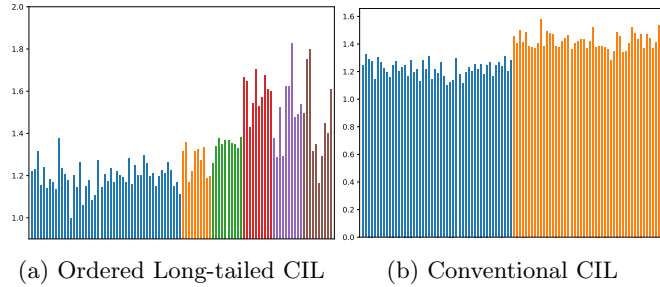


Fig. 8: LWS weights for Ordered LT-CIL (6 tasks) and conventional CIL scenarios (2 tasks) on ImageNet-Subset. Different colors indicate the weights for different tasks.

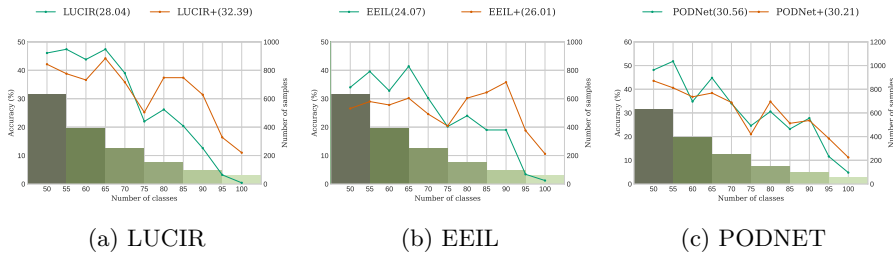


Fig. 9: The average accuracy curves with data distribution bars at the last task for Ordered LT-CIL on CIFAR100.

from our two-stage approach, with performance increasing by up to 6.7. More results on ImageNet-Subset can be found in the supplementary material.

Analysis of LWS layer. In Fig. 8 we plot the weights of the LWS layer for LUCIR+ after learning the last task. Since the classes in the end of training consist of fewer and fewer samples, the LWS is capable of learning larger weights for them to balance with previous classes. In the conventional CIL scenario, the LWS weight for the current task is significantly larger than the older one, which can help predict correct labels for current classifier without modifying the feature representations too much in a two-stage framework, and thus reducing the forgetting of previous knowledge.

Visualization of effectiveness of our method. As seen in Fig. 9, different methods and their corresponding two-stage versions are evaluated after learning the last task (5-tasks). It is clear that LUCIR+ and EEIL+ can significantly boost the performance of tail classes by losing relatively less for the head classes. PODNET+ improves over PODNET by a small margin for the tail classes with less drops for head classes.

Ablation on class order. To verify the robustness of our approach, we ran experiments with four different random seeds in the Shuffled LT-CIL scenario on CIFAR-100. As shown in Fig. 10, the compared methods behave differently but

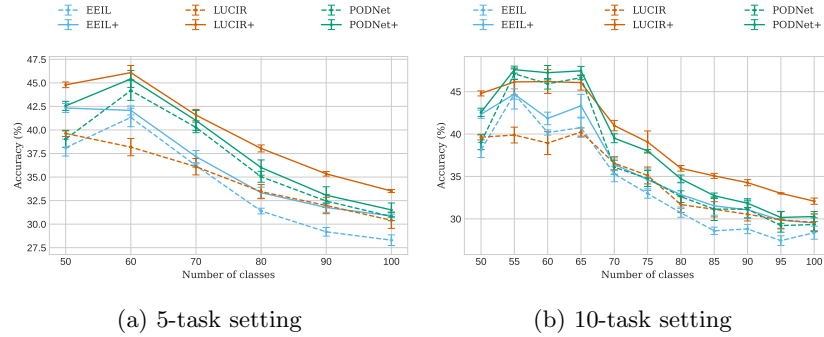


Fig. 10: Average accuracy for Shuffled LT-CIL with multiple random seeds on CIFAR-100. Error bars are shown at each task.

the overall trend for all methods are clear. LUCIR remains on the top in the last several tasks. All methods with the proposed two-stage strategy improve over the baselines in both settings.

5 Conclusions

In this paper we proposed two novel scenarios for class incremental learning over long-tailed distributions (LT-CIL). Ordered LT-CIL considers the case where subsequent tasks contain consistently fewer samples than previous ones. Shuffled LT-CIL, on the other hand, refers to the case in which the degree of imbalance for each task is different and randomly distributed. Our experiments demonstrate that the existing state-of-the-art in CIL is significantly less robust when applied to long-tailed class distribution. To address the problem of LT-CIL, we propose a two-stage method with a learnable weight scaling layer that compensates for class imbalance. Our approach significantly outperforms the state-of-the-art on CIFAR-100 and ImageNet100 with long-tailed class imbalance. Our two-stage approach is complimentary to existing methods for CIL and can be easily and profitably integrated into them. We believe that our work can serve as a test bed for future development of long-tailed class incremental learning.

Acknowledgments This work is funded by National Key Research and the Development Program of China (NO. 2018AAA0100400), NSFC (NO. 61922046), NSFC (NO. 62206135) the S&T innovation project from the Chinese Ministry of Education, and by the European Commission under the Horizon 2020 Programme, grant number 951911 – AI4Media.

References

1. Abdelsalam, M., Faramarzi, M., Sodhani, S., Chandar, S.: Iirc: Incremental implicitly-refined classification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11038–11047 (2021) [4](#)
2. Ahn, H., Kwak, J., Lim, S., Bang, H., Kim, H., Moon, T.: Ss-il: Separated softmax for incremental learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 844–853 (2021) [4](#)
3. Aljundi, R., Babiloni, F., Elhoseiny, M., Rohrbach, M., Tuytelaars, T.: Memory aware synapses: Learning what (not) to forget. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 139–154 (2018) [3](#)
4. Belouadah, E., Popescu, A.: Il2m: Class incremental learning with dual memory. In: ICCV. pp. 583–592 (2019) [4](#)
5. Belouadah, E., Popescu, A., Kanellos, I.: A comprehensive study of class incremental learning algorithms for visual tasks. *Neural Networks* (2020) [1](#)
6. Cao, K., Wei, C., Gaidon, A., Arechiga, N., Ma, T.: Learning imbalanced datasets with label-distribution-aware margin loss. In: Advances in Neural Information Processing Systems (2019) [5](#)
7. Castro, F.M., Marín-Jiménez, M.J., Guil, N., Schmid, C., Alahari, K.: End-to-end incremental learning. In: European Conference on Computer Vision (2018) [4](#), [6](#), [8](#)
8. Chu, P., Bian, X., Liu, S., Ling, H.: 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. pp. 694–710. Springer (2020) [6](#)
9. Delange, M., Aljundi, R., Masana, M., Parisot, S., Jia, X., Leonardis, A., Slabaugh, G., Tuytelaars, T.: A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021) [1](#)
10. Douillard, A., Cord, M., Ollion, C., Robert, T., Valle, E.: Podnet: Pooled outputs distillation for small-tasks incremental learning. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16. pp. 86–102. Springer (2020) [4](#), [6](#), [8](#)
11. Han, H., Wang, W.Y., Mao, B.H.: Borderline-smote: a new over-sampling method in imbalanced data sets learning. In: International conference on intelligent computing. pp. 878–887. Springer (2005) [2](#), [4](#)
12. Hayes, T.L., Kafle, K., Shrestha, R., Acharya, M., Kanan, C.: Remind your neural network to prevent catastrophic forgetting. In: European Conference on Computer Vision. pp. 466–483. Springer (2020) [4](#)
13. Hou, S., Pan, X., Loy, C.C., Wang, Z., Lin, D.: Learning a unified classifier incrementally via rebalancing. In: International Conference on Computer Vision (2019) [4](#), [6](#), [8](#)
14. Huang, C., Li, Y., Loy, C.C., Tang, X.: Deep imbalanced learning for face recognition and attribute prediction. *IEEE transactions on pattern analysis and machine intelligence* **42**(11), 2781–2794 (2019) [2](#), [4](#)
15. Japkowicz, N., Stephen, S.: The class imbalance problem: A systematic study. *Intelligent data analysis* **6**(5), 429–449 (2002) [2](#), [4](#)
16. Kang, B., Xie, S., Rohrbach, M., Yan, Z., Gordo, A., Feng, J., Kalantidis, Y.: Decoupling representation and classifier for long-tailed recognition. In: International Conference on Learning Representations (2020) [2](#), [4](#), [6](#)
17. Kim, C.D., Jeong, J., Kim, G.: Imbalanced continual learning with partitioning reservoir sampling. In: European Conference on Computer Vision. pp. 411–428. Springer (2020) [4](#)

18. Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A.A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al.: Overcoming catastrophic forgetting in neural networks. *pnas* p. 201611835 (2017) [3](#)
19. Lee, J., Yun, J., Hwang, S., Yang, E.: Lifelong learning with dynamically expandable networks. In: *ICLR* (2018) [3](#)
20. Li, Z., Hoiem, D.: Learning without forgetting. *pami* **40**(12), 2935–2947 (2018) [3](#), [6](#), [8](#)
21. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: *Proceedings of the IEEE international conference on computer vision*. pp. 2980–2988 (2017) [2](#), [4](#)
22. Liu, X., Masana, M., Herranz, L., Van de Weijer, J., Lopez, A.M., Bagdanov, A.D.: Rotate your networks: Better weight consolidation and less catastrophic forgetting. In: *ICPR* (2018) [3](#)
23. Liu, Y., Schiele, B., Sun, Q.: Adaptive aggregation networks for class-incremental learning. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 2544–2553 (2021) [3](#), [10](#)
24. Liu, Z., Miao, Z., Zhan, X., Wang, J., Gong, B., Yu, S.X.: Large-scale long-tailed recognition in an open world. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 2537–2546 (2019) [2](#), [4](#)
25. Mallya, A., Davis, D., Lazebnik, S.: Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In: *ECCV*. pp. 67–82 (2018) [3](#)
26. Mallya, A., Lazebnik, S.: Packnet: Adding multiple tasks to a single network by iterative pruning. In: *CVPR*. pp. 7765–7773 (2018) [3](#)
27. Masana, M., Liu, X., Twardowski, B., Menta, M., Bagdanov, A.D., van de Weijer, J.: Class-incremental learning: survey and performance evaluation. *arXiv preprint arXiv:2010.15277* (2020) [1](#), [2](#), [8](#)
28. McCloskey, M., Cohen, N.J.: Catastrophic interference in connectionist networks: The sequential learning problem. In: *Psychology of learning and motivation*, vol. 24, pp. 109–165. Elsevier (1989) [1](#)
29. Parisi, G.I., Kemker, R., Part, J.L., Kanan, C., Wermter, S.: Continual lifelong learning with neural networks: A review. *Neural Networks* **113**, 54–71 (2019) [1](#)
30. Perez, L., Wang, J.: The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621* (2017) [4](#)
31. Rajasegaran, J., Hayat, M., Khan, S., Khan, F.S., Shao, L.: Random path selection for incremental learning. In: *Advances in Neural Information Processing Systems* (2019) [3](#)
32. Rebuffi, S.A., Kolesnikov, A., Sperl, G., Lampert, C.H.: icarl: Incremental classifier and representation learning. In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. pp. 2001–2010 (2017) [4](#)
33. Schwarz, J., Czarnecki, W., Luketina, J., Grabska-Barwinska, A., Whye Teh, Y., Pascanu, R., Hadsell, R.: Progress & compress: A scalable framework for continual learning. In: *Proceedings of International Conference on Machine Learning*. vol. 80, pp. 4528–4537 (2018) [3](#)
34. Serra, J., Suris, D., Miron, M., Karatzoglou, A.: Overcoming catastrophic forgetting with hard attention to the task. In: *ICML*. pp. 4555–4564 (2018) [3](#)
35. Shin, H., Lee, J.K., Kim, J., Kim, J.: Continual learning with deep generative replay. In: *NIPS* (2017) [4](#)
36. Shu, J., Xie, Q., Yi, L., Zhao, Q., Zhou, S., Xu, Z., Meng, D.: Meta-weight-net: Learning an explicit mapping for sample weighting. *arXiv preprint arXiv:1902.07379* (2019) [2](#), [4](#)

37. Tao, X., Hong, X., Chang, X., Dong, S., Wei, X., Gong, Y.: Few-shot class-incremental learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12183–12192 (2020) [2](#)
38. Van de Ven, G.M., Tolias, A.S.: Three scenarios for continual learning. arXiv preprint arXiv:1904.07734 (2019) [1](#), [3](#)
39. Wang, X., Lian, L., Miao, Z., Liu, Z., Yu, S.X.: Long-tailed recognition by routing diverse distribution-aware experts. In: International Conference on Learning Representations (2021) [2](#), [4](#)
40. Wang, Y.X., Ramanan, D., Hebert, M.: Learning to model the tail. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. pp. 7032–7042 (2017) [2](#), [4](#)
41. Wu, Y., Chen, Y., Wang, L., Ye, Y., Liu, Z., Guo, Y., Fu, Y.: Large scale incremental learning. In: International Conference on Computer Vision (2019) [4](#)
42. Xiang, L., Ding, G., Han, J.: Learning from multiple experts: Self-paced knowledge distillation for long-tailed classification. In: European Conference on Computer Vision. pp. 247–263. Springer (2020) [2](#), [4](#)
43. Yan, S., Xie, J., He, X.: Der: Dynamically expandable representation for class incremental learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3014–3023 (2021) [3](#), [10](#)
44. Yin, X., Yu, X., Sohn, K., Liu, X., Chandraker, M.: Feature transfer learning for face recognition with under-represented data. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5704–5713 (2019) [2](#), [4](#)
45. Zenke, F., Poole, B., Ganguli, S.: Continual learning through synaptic intelligence. In: ICML. pp. 3987–3995. JMLR. org (2017) [3](#)
46. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. In: International Conference on Learning Representations (2018) [4](#)
47. Zhang, S., Li, Z., Yan, S., He, X., Sun, J.: Distribution alignment: A unified framework for long-tail visual recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2361–2370 (2021) [6](#)
48. Zhang, X., Fang, Z., Wen, Y., Li, Z., Qiao, Y.: Range loss for deep face recognition with long-tailed training data. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 5409–5418 (2017) [2](#), [4](#)
49. Zhong, Y., Deng, W., Wang, M., Hu, J., Peng, J., Tao, X., Huang, Y.: Unequal-training for deep face recognition with long-tailed noisy data. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7812–7821 (2019) [2](#), [4](#)
50. Zhong, Z., Cui, J., Liu, S., Jia, J.: Improving calibration for long-tailed recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16489–16498 (2021) [2](#), [4](#), [6](#)
51. Zhou, B., Cui, Q., Wei, X.S., Chen, Z.M.: Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9719–9728 (2020) [4](#)

| | EEIL | EEIL+ | LUCIR | LUCIR+ | PODNet | PODNet+ |
|--------------|-------|-------|-------|--------|--------------|--------------|
| Conventional | 44.14 | 55.92 | 53.60 | 56.96 | 60.25 | 64.75 |
| Ordered | 47.18 | 46.43 | 47.05 | 49.77 | 58.30 | 57.76 |
| Shuffled | 40.75 | 43.95 | 40.52 | 48.35 | 48.93 | 51.52 |

Table 4: The average accuracy on long sequence of 25 steps for three different scenarios on Imagenet-Subset. Methods with + sign indicate our two-stage method applied to the corresponding baseline.

A More results on ImageNet-Subset

Different imbalance ratios on ImageNet-Subset In this section, we analyze three different imbalance ratios $\rho = 0.01$, $\rho = 0.05$ and $\rho = 0.1$ on ImageNet-Subset, the smaller the ratio the more skewed the distribution. Compared to CIFAR100, ImageNet-Subset contains more samples, which results in a more skewed distribution on different continual training steps. We report three baselines, i.e. EEIL, LUCIR and PODNET, and our two-stage approach applied to them, denoted as EEIL+, LUCIR+ and PODNET+. As we can see from Figure 11 (a-c), with more samples, for the ordered LT-CIL scenario, PODNET surpasses other approaches consistently with a large margin, obtaining the best performance in all scenarios. We consider that PODNET can learn much more information when the data is sufficient. Overall our two-stage method can consistently boost accuracy for most methods, especially for LUCIR+ with a significant gain. For shuffled LT-CIL scenario from Figure 11 (d-f), PODNET+ and LUCIR+ are very competitive in all three imbalance ratio ρ . The proposed two-stage method further improves the performance, especially for EEIL and LUCIR. Interestingly, we can see that compared to conventional settings, long-tailed scenarios with a large imbalance ratio can achieve competitive performance with less samples, which may due to the imbalance effect of training data.

Long sequence on ImageNet-Subset We evaluate on long sequence of 25 steps for all three scenarios with three state-of-the-art methods on ImageNet-Subset, and collect the results in Table 4. As we can see, our method also improves over different baselines in this more challenging setting like on CIFAR100 except for PODNET on ordered LT-CIL scenario. Further more, we can see that for 25-step scenario, two-stage methods can get much larger gain than in 5-step and 10-step scenarios in most cases. It shows that the two-stage methods are more robust for longer sequences.

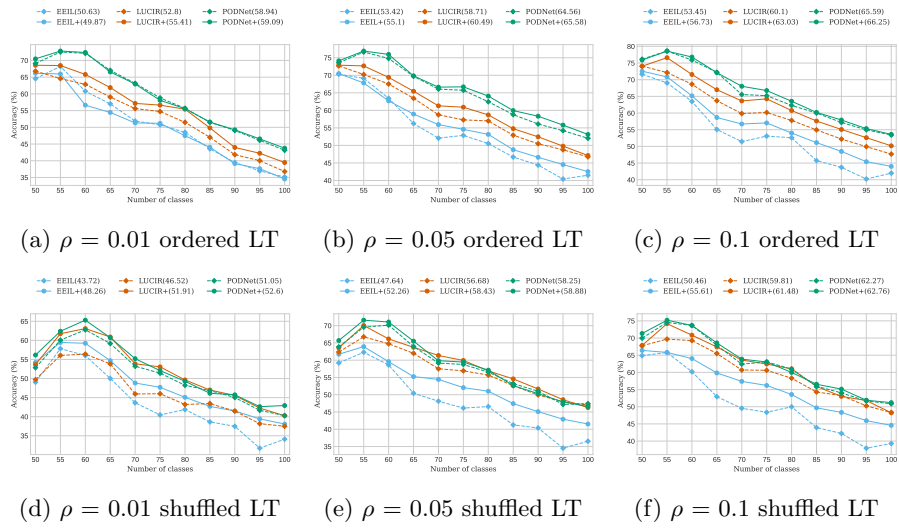


Fig. 11: Average accuracy on different imbalance ratios on Imagenet-Subset, the top row is on ordered LT-CIL and the bottom row is on shuffled LT-CIL. Methods with + sign indicate our two-stage method applied to the corresponding baseline.