THE UNREASONABLE EFFECTIVENESS OF THE CLASS-REVERSED SAMPLING IN TAIL SAMPLE MEMORIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-tailed visual recognition poses significant challenges to traditional machine learning and emerging deep networks due to its inherent class imbalance. A common belief is that tail classes with few samples cannot exhibit enough regularity for pattern extraction. What makes things worse, the limited cardinality may lead to low exposure of tail classes in the training stage. Re-sampling methods, especially those who naively enlarge the exposure frequency, eventually fail with head classes under-represented and tail classes overfitted.

Arguing that long-tailed learning involves a trade-off between head class pattern extraction and tail class memorizing, we first empirically identify the regularity of classes under long-tailed distributions and find that regularity of the same training samples will be sharply decreased with the reduction of class cardinality. Motivated by the recent success of a series works on the memorization-generalization mechanism, we propose a simple yet effective training strategy by switching from instance-balanced sampling to class-reversed sampling to memorize tail classes without seriously damaging the representation of head classes. Closely afterwards, we give the theoretical generalization error upper bound to prove that class-reversed sampling is better than instance-balanced sampling during the last training stage. In our experiments, the proposed method can reach the state-of-the-art performance more efficiently than current methods, on several datasets. Further experiments also validate the superior performance of the proposed sampling strategy, implying that the long-tailed learning trade-off could be effectively tackled only in the memorization stage with a small learning rate and over-exposure of tail samples.

1 INTRODUCTION

With the prosperity of deep learning research field, visual recognition has witnessed the prominence of powerful representation learning approaches and high-quality, large-scale datasets, *e.g.*, ImageNet ILSVRC (Russakovsky et al. (2015)) and Places (Zhou et al. (2018)). These datasets are usually carefully balanced, exhibiting roughly uniform distributions of class labels. However, visual phenomena in real world tend to have skewed data distributions with long-tailed characteristics (Dong et al. (2017); Xiang & Ding (2020)), consisting of a few majority classes (*head classes*) and a large number of minority classes (*tail classes*). When dealing with such long-tailed data, many standard approaches fail to work well due to the extreme class imbalance trouble, leading to a significant drop in accuracy for tail classes.

In this paper, we propose to investigate long-tailed visual recognition from a memorization-generalization point of view. A recent work of Feldman (Feldman (2020)) introduced a new theoretical explanation suggesting that memorization is necessary for achieving close-to-optimal generalization when the data distribution is long-tailed, since useful samples from tail classes can be statistically indistinguishable from the useless one. To verify the impact of cardinality on each class during training, we firstly visualize the cumulative learned events and forgetting events of each sample (see Section A.4) and find that the *long-tail challenge is essentially a trade-off between the representation of high-regularity head classes and generalization to low-regularity tail classes*.

Conventional re-sampling methods, which tend to naively change the exposure frequency of each class to generate class-balanced data (Han et al. (2005); Buda et al. (2018)), are generally thought to be flawed since model may suffer from head classes being under-represented and tail classes overfitted (Zhang et al. (2019); Ye et al. (2020)). One recent study (Kang et al. (2020)) has suggested that the learning of feature representations and classifiers should be completely decoupled and different training stages need different data samplers. They thus train the feature extractor with instance-balanced sampling first and then use class-balanced sampling to re-train the classifier with a fixed backbone. This approach is intuitive, and has proven empirically successful. However, it is not without limitations: For instance, training the feature extractor with only instance-balanced sampling means head classes will always dominate the training procedure, leading under-representation for tail classes (Zhang et al. (2019); Zhou et al. (2020)). On the other hand, the re-trained classifier may be sub-optimal if the feature representation is sub-optimal (see Section A.6).

To challenge the Decouple (Kang et al. (2020)) hypothesis, we propose an end-to-end method by properly combining different samplers to overcome these limitations of complete decoupling. Our method is motivated by the memorization-generalization phenomenon in deep learning (Jiang et al. (2020)), which indicates that rare and low-regularity samples could be able to be learned based on the internal representations learned from strongest-domain regularities first. We argue that properly shifting learning focus from high-regularity head classes to low-regularity tail classes is unreasonably effective and we provide intuitive insight into solving long-tailed problems. To support this, we explore a joint training strategy which performs well **without decoupling**.

Specifically, we employ the standard training procedure with cross-entropy loss and instance-balanced sampler w.r.t. the original data distribution to ensure the learning of universal visual patterns. We only switch from instance-balanced sampler to class-reversed sampler for the last several epochs of training, when tail classes tend to be over-exposed. In earlier training, when head classes dominate the training data, the patterns and structures discovered in the regular examples are utilized to build a generalizable representation. In later training phase, the memorization of tail classes will not seriously disrupt the learned representation as the learning rate is much smaller than in earlier stages. Such a training strategy can simultaneously boost the representation and classification towards long-tailed distributions, avoiding the risk of excessive dependence on the feature extractor (see Section A.6).

We conduct extensive experiments across four benchmark long-tailed datasets, CIFAR10-LT, CIFAR100-LT, iNaturalist 2018 and ImageNet-LT, to evaluate the effectiveness of our proposed method. With such a simple training strategy, we obtain compare or better results and more efficient compared with previous state-of-the-art methods.

To summarize, the main contributions of this paper are as follows:

- We empirically identify that the *low regularity of tail classes* is the primary hurdle for learning an accurate model for long-tailed distributions and appropriately memorizing them is essential for better generalization across all classes.
- We propose a simple yet effective switching strategy to handle the trade-off between high-regularity head classes and low-regularity tail classes, and give the theoretical generalization error bound proving that class-reversed sampling is better than instance-balanced sampling during the last training stage.
- We investigate the effectiveness and efficiency of the proposed method through extensive experimentation and demonstrate that the tackling the long tail trade-off problem could only cost a few training epochs with a small learning rate and over-exposure of tail samples.

2 Related Work

2.1 Long-tailed visual recognition

Here we mainly introduce the re-sampling methods, other types of long-tailed recognition methods can be found in Section A.2.

Re-sampling strategies. Re-sampling strategies can be divided into two classical types: oversampling the minority classes by repeatedly adding augmented images (Drummond et al. (2003);

Han et al. (2005); Buda et al. (2018)); or under-sampling the majority classes by removing several images (Japkowicz & Stephen (2002); He & Garcia (2009)). All these re-sampling methods tend to provide a more balanced data distribution during training to solve the long-tailed problem. However, sometimes, over-sampling may cause over-fitting towards minority classes, while under-sampling may weaken the representation ability of networks.

Different from these, we do not artificially generate class-balanced batches or losses; instead, we simply emphasize the memorization of low-regularity tail class samples by only switching from the instance-balanced sampler to class-reversed sampler during the standard training procedure.

2.2 Memorization-generalization mechanism in deep learning

Memorization was once considered a failure of deep networks since it implies a lack of generalization. However, the view that memorization is harmful may be a misunderstanding towards deep learning. Zhang et al. (2017) were the first to demonstrate that standard deep learning algorithms can achieve high training accuracy even on large and randomly labeled datasets, generating a large wave of research interest in the topic of generalization for deep learning. Toneva et al. (2019) introduced the "forgetting event" to describe the learning dynamics of neural networks, where some instances flip flop between "learned" and "forgotten" states during training. In order to analyze how individual instances are treated by a model on the memorization-generalization continuum, Jiang et al. (2020) proposed the C-score to measure the consistency of a sample with respect to the rest of the training set. They found that samples having lower C-scores are learned more slowly, indicating the need for a stage-wise learning rate schedule during training.

A recent work of Feldman (2020) proposed a new theoretical explanation for the benefits of memorization. In their abstract model, algorithm can only get the frequency of a subpopulation through the empirical frequency of its representatives, thus it can only avoid the risk of missing subpopulations with significant frequency by memorizing examples. Further, Feldman & Zhang (2020) introduced the influence estimation to validate the necessity of memorizing useful examples for achieving close-to-optimal generalization error.

Different from them, we first empirically identity the correlation between cardinality and regularity under long-tailed distributions and propose a simple yet effective method to boost the performance of long-tailed visual recognition via memorizing low-regularity tail class samples.

3 Method

Long-tailed visual recognition follows a long-tailed distribution over classes, leading the model to exhibit under-fitting on tail classes and over-fitting to head classes. Since increasing the exposure of tail classes may lead to over-fitting while under-sampling head classes may weaken the representation ability of networks, the trade-off between the representation of head classes and generalization towards tail classes becomes the main dilemma in long-tailed problem. Here we propose to resolve this dilemma by only switching the standard instance-balanced sampler to a class-reversed sampler during the last training procedure, in order to learn low-regularity samples (tail classes) without seriously disrupting the representation of the strongest domain regularities (head classes) first.

3.1 Problem Setup and Notations

Let $f_\theta(\cdot)$ denote a feature extractor implemented by a CNN model with parameter θ, we get the class prediction through $\hat{y} = \arg\max g(f_\theta(\mathbf{x}))$, where $\mathbf{x}$ is the input image and $g(\cdot)$ is a classifier function. Given a training set $\mathcal{D} = \{x_i, y_i\}, i \in \{1, ..., n\}$ with C classes, let n_j denote the number of samples for class j and $n = \sum_{i=1}^{C} n_i$ be the total number of samples. Without loss of generality, we assume that the classes are sorted by cardinality in decreasing order, *i.e.*, if $i < j$, then $n_i \geq n_j$. For most sampling strategies, the probability p_j of sampling a data point from class j is given by:

$$p_j = \frac{n_j^q}{\sum_{i=1}^{C} n_i^q}, \tag{1}$$

with different values of q arise for different sampling strategies. The sampling of each data can be capsuled into the following two steps: 1) Randomly sample a class according to p_j; 2) Uniformly pick up a sample from class j. Sampling strategies that correspond to q = 1, q = 0, and q = -1 are introduced as below:

Instance-balanced sampling (IB). This is the most common and standard way of sampling data, where each sample of the training dataset is sampled only once with equal probability in a training epoch. For instance-balanced sampling, the probability p_j^{IB} is given by Equation 1 with q = 1, i.e., a sample from class j will be sampled proportionally to the cardinality n_j of the class.

Class-balanced sampling (CB). To alleviate the extreme data imbalance during training, class-balanced sampling is proposed to artificially generate class-balanced data. The probability p_j^{CB} is given by Equation 1 with q = 0, *e.g.*, $p_j^{CB} = 1/C$. In this scenario, the probability of each class j being selected is equal, independent to its cardinality n_j.

Class-reversed sampling (CR). Zhou et al. (2020) firstly propose the reversed sampler to re-balance feature representation and particularly improve the classification accuracy on tail classes. Here we integrate p_j^{CR} into Equation 1 with q = -1. For class-reversed sampling, a data point from class j will be sampled proportionally to the reciprocal of its cardinality n_j, *i.e.*, the more samples in a class, the smaller sampling possibility that class has.

The sampling weights of IB, CB and CR are visualized in Figure 1(a).

3.2 Switching Data Samplers during Training

Switching is proposed to shift the learning focus from head classes to tail classes by simply switching the IB sampler to CR sampler at some epoch during training. Before the switching happens, the uniform IB sampler retains the characteristics of original distributions and almost the high-regularity samples from head classes are learned, the patterns and structures discovered in those head class samples can be used to build a generalizable representation. In later stages, the memorization of tail class samples will not seriously disrupt the learned representation as the learning rate is much smaller than the earlier stages.

Concretely, the number of total training epochs is denoted as T and the learning rate milestones are denoted as $[m_1, ..., m_n]$, where $m_1 < ... < m_n \leq T$. Let $\gamma \in (0, 1)$ becomes the multiplicative factor, learning rate will be decayed by γ once the epoch reaches one of the learning rate milestones during training. When training procedure reaches the $m_n + S$ epoch, we switch IB sampling to CR sampling and continuing training, where S is the hyper-parameter in our method indicating when to switch. The details of our switching strategy is shown in Algorithm 1.

Our method is simple and clean, which only switches the data sampler from IB to CR during training, without changing any structure of the original network or artificially generating class-balanced training batches or losses.

Algorithm 1 Hard Switching From IB to CR

```
Input: $D$: Training Set;    $M$: Mini-batch size;
Input: $T$: Total epochs;    $[m_1, ..., m_n]$: Learning rate decay milestones;    $S$: Switch epoch;
initial $p_j^{CR} = \frac{n_j^{-1}}{\sum_{i=1}^{C} n_i^{-1}}$;
for $t \in [1, T]$ do
    $B = \{\}$;
    if $t \leq m_n + S$ then
        Randomly sample M samples from $D$ as $B$.
    else
        for $i \in [1, M]$ do
            Sample one class $j$ from the $D$ according to $p_j^{CR}$.
            Uniformly pick up a sample $\{x_i, y_i\}$ from class $j$ without replacement.
            $B = \{B; \{x_i, y_i\}\}$.
        end for
    end if
    Optimize network by SGD based on $B$.
end for
```

3.3 Theoretical Analysis

In this section, we answer the following three questions: (1) what is the objective function of the optimization, (2) what is the generalization error upper bound of different sampling strategies, and (3) why CR is better than IB during the last training stage with small learning rate.

Objective function. Let $L_{ji}(\theta)$ denote the standard training error on the ith sample of class j:

$$L_{ji}(\theta) = \ell\left(f_\theta\left(x_{ji}\right), y_{ji}\right), \tag{2}$$

where ℓ is the loss function, *e.g.*, cross-entropy loss.

For the standard training process with IB sampling, where each sample of the training dataset is sampled with equal probability, the object function over the total training set $\mathcal{D}$ is given as follows:

$$L^s(\theta) = \frac{1}{n}\sum_{i=1}^{n} L_i(\theta) + R(\theta), \tag{3}$$

where $R(\theta)$ is the regular terms.

Now considering a more general scene, where sampling a data containing two steps: 1) Randomly chooses one class according to p_j; 2) Uniformly pick up one sample from its n_j samples, we have:

$$L(\theta) = \sum_{j=1}^{C}\sum_{i=1}^{n_j} \frac{p_j}{n_j} L_{ji}(\theta) + R(\theta). \tag{4}$$

Generalization error bound. Now we give the generalization theory of such a objective function. Let Θ be the family function of our learned neural network, we define $\mathfrak{R}_\mathrm{n}(\Theta)$ as the standard Rademacher complexity (Bartlett & Mendelson (2002)) of the set $\{(x,y) \mapsto \ell(f(x;\theta),y) : \theta \in \Theta\}$:

$$\mathfrak{R}_n(\Theta) = \mathbb{E}_{\mathcal{D},\xi}\left[\sup_{\theta\in\Theta}\frac{1}{n}\sum_{i=1}^{n}\xi_i \ell\left(f_\theta\left(x_{ji}\right), y_{ji}\right)\right], \tag{5}$$

where $\xi_1, \ldots, \xi_n$ are in-dependent uniform random variables taking values in {-1,1} (*i.e.* Rademacher variables).

Let M denote the least upper bound on the difference of individual loss values: $|\ell(f_\theta(x),y) - \ell\left(f_\theta\left(x'\right), y'\right)| \leq M$ for all $\theta \in \Theta$. For the standard training process with $L^s(\theta)$, for any $\delta > 0$, with probability at least $1-\delta$ over the training set $\mathcal{D}$, the following error bound holds for all $\theta \in \Theta$ (Kawaguchi & Lu (2020)):

$$\mathbb{E}^s_{(x,y)}[\ell(f_\theta(x),y)] \leq L^s(\theta) + 2\mathfrak{R}_n(\Theta) + M\sqrt{\frac{\ln(1/\delta)}{2n}}. \tag{6}$$

For the general objective function L_θ, for any $\delta > 0$, with probability at least $1-\delta$ over the training set $\mathcal{D}$, we have the following error bound for all $\theta \in \Theta$ (the proof is given in Section A.1):

$$\mathbb{E}_{(x,y)}[\ell(f_\theta(x),y)] \leq L(\theta) + 2\mathfrak{R}_n(\Theta) - \mathcal{Q}_n(\Theta;p,n) + M\sqrt{\sum_{j\in C}\frac{p_j^2}{n_j}}\sqrt{\frac{\ln(1/\delta)}{2}}, \tag{7}$$

where $\mathcal{Q}_n(\Theta;p,n) \quad = \quad \mathbb{E}_{\mathcal{D}}\left[\inf_{\theta\in\Theta}\sum_{j=1}^{C}\sum_{i=1}^{n_j}\left(\frac{p_j}{n_j}-\frac{1}{n}\right)\ell\left(f_\theta\left(x_{ji}\right), y_{ji}\right)\right]$.

Theorem 1. *With a small size of Θ (the last training stage with small learning rate) and a bounded M, the upper bound on the expected error for CR sampling is strictly **lower** than IB sampling if $\mathcal{Q}_n(\Theta;p,n) + L^s - L > 0$ or if $L^s - L > 0$* (the proof is given in Section A.1).

4 Experiments

To investigate the effectiveness and the efficiency of our proposed approach, we conduct several qualitative and quantitative experiments on imbalance controlled datasets like long-tailed CIFAR-10 and CIFAR-100 as well as large-scale real-world long-tailed dataset, i.e. iNaturalist 2018 and ImageNet-LT, as described in Section A.3.

4.1 Comparison with the state-of-the-art on long-tailed datasets

In this section, we compare the performance of the proposed scheme to other recent works that report state-of-the-art results on four common long-tailed benchmarks: Long-tailed CIFAR-10, Long-tailed CIFAR-100, iNaturalist2018 and ImageNet-LT.

Table 1: Top-1 accuracy of ResNet-32 on long-tailed CIFAR-10 and CIFAR-100. Rows with †denote results directly copied from Zhou et al. (2020). * denotes results reproduced with the authors code.

Dataset	Long-tailed CIFAR-10			Long-tailed CIFAR-100		
Imbalance ratio	100	50	10	100	50	10
Cross Entropy†	70.4	74.8	86.7	38.3	43.9	55.7
Cross Entropy*	73.1	77.9	86.4	40.7	44.9	57.2
Focal† (Lin et al. (2017))	70.4	76.7	86.7	38.4	44.3	55.8
CE-DRW† (Cao et al. (2019))	76.3	80.0	87.6	41.5	45.3	58.1
CE-DRS† (Cao et al. (2019))	75.6	79.8	87.4	41.6	45.5	58.1
CB-Focal† (Cui et al. (2019))	74.6	79.3	87.1	39.6	45.2	58.0
LDAM-DRW† (Cao et al. (2019))	77.0	81.0	88.2	42.0	46.6	58.7
Decouple-cRT* (Kang et al. (2020))	73.8	80.7	86.7	40.1	46.4	57.7
Decouple-LWS* (Kang et al. (2020))	73.5	77.5	86.1	40.2	45.7	58.1
BBN† (Zhou et al. (2020))	**79.8**	82.2	88.3	42.6	47.0	59.1
Ours ($S = 1$)	**79.7**	**82.9**	**88.4**	**44.7**	**49.5**	**59.5**

Table 2: Top-1 accuracy of ResNet-50 on iNaturalist 2018. Rows with †denote results directly copied from their original paper. We present results when training for 90 / 200 epochs.

Dataset	iNaturalist 2018
Cross Entropy†	57.2
CE-DRW†	63.7
CE-DRS†	63.6
CB-Focal†	61.1
LDAM-DRW†	68.0
Decouple-cRT†	65.2 / 67.6
Decouple-LWS†	65.9 / 69.5
BBN†	66.3 / 69.6
Ours ($S = 1$ / $S = 1$)	**66.7 / 70.4**
Ours ($S = 10$ / $S = 40$)	**66.8 / 70.0**

Table 3: Top-1 accuracy of on large-scale long-tailed datasets ImageNet-LT for different backbone architectures. Rows with †denote results directly copied from Kang et al. (2020). * denotes results reproduced with the authors code.

Method	ResNet-50	ResNext-50
Cross Entropy†	41.6	44.4
OLTR† (Liu et al. (2019))	-	37.7
Decouple-cRT†	47.3	49.5
Decouple-LWS†	47.7	**49.9**
BBN*	45.9	47.1
Ours ($S = 1$)	**47.9**	49.2
Ours ($S = 10$)	**48.2**	49.3

Long-tailed CIFAR. We conduct extensive experiments on long-tailed CIFAR datasets with three different imbalanced ratios: 10, 50 and 100. Table 1 reports the accuracy of various methods. For CIFAR-10 series, our method achieves comparable or better results comparing other complicated methods. When working on CIFAR-100 series, our method achieves the best results across all imbalance ratios, compared with the two-stage fine-tuning strategies (i.e., CE-DRW/CE-DRS) and also previous state-of-the-arts (i.e., Decouple (Kang et al. (2020)) and BBN (Zhou et al. (2020))). Especially for long-tailed CIFAR-100 with imbalanced ratio 100 (the most extreme imbalance case), we get **44.7%** accuracy which is **2.1%** higher than BBN (Zhou et al. (2020)).

iNaturalist 2018. We further evaluate our methods on the iNaturalist 2018 dataset. Similar to Kang et al. (2020) and Zhou et al. (2020), we present results training after 90 and 200 epochs. From the Table 2 we can see, we surpass other complicated methods even with a plain ResNet-50 model. When S =1, where total training epochs are 10 epoch less than Decouple, we get **1.5%** gains compared with the totally decouple training strategy cRT. We can achieve further improvements with the same training epochs as Decouple (see $S = 10$ and $S = 40$).

ImageNet-LT. Table 3 presents results for ImageNet-LT. We present results for two backbones: ResNet-50 and more powerful ResNext-50. The results of BBN are conducted using the author's open-sourced codebase. From the table we see that our simple method, without bells and whistles outperform the current state-of-the-art for ResNet-50, about **0.9%** higher than Decouple and **2.1%** higher than BBN. With more powerful backbone, our method performs slightly worser than Decouple, while still surpasses the more complicated architecture BBN about **2.2%**.

Fine-grained analysis To better validate our assumption that memorizing low-regularity samples with small learning rate can avoid seriously damage the representation of high-regularity samples, we further report accuracy on three splits of the set of classes: *Many-shot* (more than 100 images), *Medium-shot* (20∼100 images) and *Few-shot* (less than 20 images).

As shown in Table 4 and Table 5, standard training process (see Cross Entropy) always perform best on Many-shot since the head class samples dominate the training batch all the time. Meanwhile, our method can improve the performance of tail classes by a large margin due to the CR sampling in the last training stage. It is worth to note that while greatly boosting the recognition of tail classes, our switching method only sightly damage the performance of head classes (compared with Decouple and BBN), indicating that memorizing tail class samples with small learning rate can better handle the trade-off between high-regularity head classes and low-regularity tail classes.

Table 4: Comprehensive results on the most skewed long-tailed CIFAR-100 (imbalance ratio: 100).

Method	Many-shot	Medium-shot	Few-shot	All
Cross Entropy	**68.2**	39.7	9.9	40.7
Decouple-cRT	58.1	40.3	18.0	40.1
Decouple-LWS	59.5	40.7	17.4	40.2
BBN	54.5	**51.0**	16.7	42.6
Ours ($S = 1$)	57.1	48.1	**26.5**	**44.7**

Table 5: Comprehensive results on ImageNet-LT with different backbone networks (ResNet-50, ResNext-50).

Method	ResNet-50				ResNext-50			
	Many	Medium	Few	All	Many	Medium	Few	All
Cross Entropy	**64.0**	33.8	5.8	41.6	**65.9**	37.5	7.7	44.4
Decouple-cRT	58.8	44.0	26.1	47.3	61.8	46.2	27.4	49.5
Decouple-LWS	57.1	45.2	29.3	47.7	60.2	**47.2**	30.3	**49.9**
BBN	56.2	**46.6**	14.1	45.9	58.1	**47.2**	15.6	47.1
Ours ($S = 1$)	53.5	45.2	**41.8**	**47.9**	55.2	46.7	**39.0**	49.2
Ours ($S = 10$)	54.5	45.0	**41.9**	**48.2**	56.5	46.1	**39.4**	49.3

4.2 Ablation Study

To find the optimal setting of S, which is the hyper-parameter controlling when to switch, we investigate the S value and corresponding results are shown in Table 6. Interestingly, our method achieves comparable results despite different values of S, indicating S is not dataset/distribution dependent or sensitive. This is consistent with our motivation: memorization of tail classes will not seriously disrupt the learned representation with smaller learning rate. Thus, once there is a CR sampling during the small learning rate training, model could jointly fine-tune both feature extractor and classifier to achieve better generalization, regardless of the specific value of S.

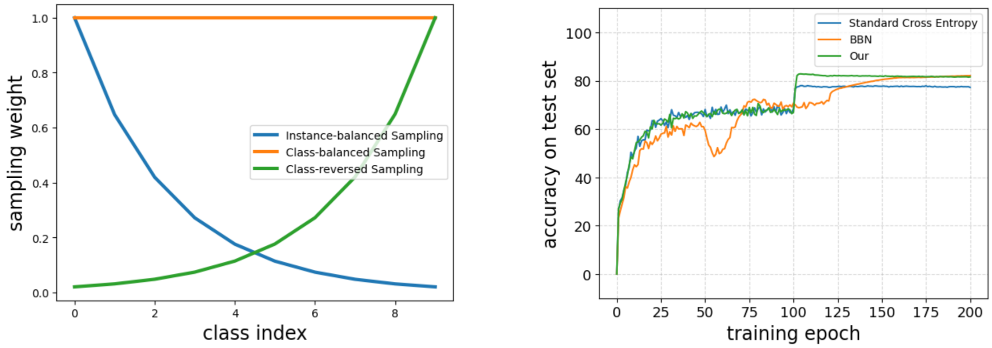
We also investigate the progressively switching in Table 7. For the first CB then CR setting, the results are almost the same as only CR, showing switching strategy is robust to the samplers used in earlier stages. However, first CR then CB will lead a great drop in accuracy, indicating memorizing low-regularity samples of tail classes should happen in the last training stage.

Table 6: Determining of the optimal S on long-tailed CIFAR-10 (imbalance ratio: 50) and CIFAR-100 (imbalance ratio: 50).

S	CIFAR-10	CIFAR-100
0	82.6	**49.7**
1	**82.9**	**49.5**
5	**82.9**	49.3
10	82.8	49.1
50	82.6	49.1

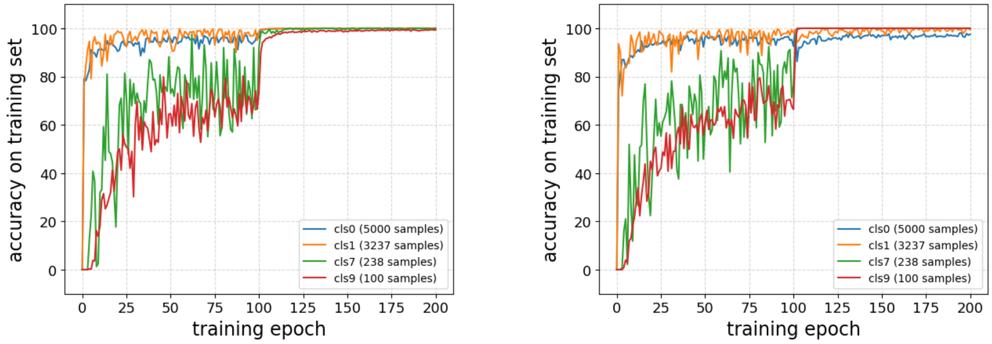
Table 7: Determining of switching strategies on long-tailed CIFAR-10 (imbalance ratio: 50).

First Sampler	Switching Sampler	Accuracy
IB	CR	**82.9**
	CB for 5 epochs, then CR	82.7
	CR for 5 epochs, then CB	79.2



(a) samling weights (b) test accuracy

Figure 1: Sampling weights and test performance of three methods with different sampling strategies on long-tailed CIFAR-10 with imbalance ratio 50. Left: instance-balanced, class-balanced and class-reversed sampling. Right: standard cross entropy, BBN and our switching.



(a) standard (b) ours

Figure 2: Learning speed of examples of 4 selected classes with SGD using stage-wise constant learning rate. Left: standard training procedure. Right: our switching training procedure.

4.3 Learning Speed

In order to further validate our method could learn long-tailed distributions more efficiently, we plot the test accuracy per epoch of three methods with different sampling strategies in Figure 1(b). Compared with using IB only, switching to CR can **immediately** improve the performance by a large margin. Meanwhile, although BBN could achieve comparable performance with ours, it converges more slowly since it optimizes two branches of feature extractor in turn during training.

Intuitively, a training example from head classes should be learned quickly since it is consistent with many others and the gradient steps for all consistent examples should be well aligned. As Jiang et al. (2020) indicates that strong regularities in a data set are not only better learned at asymptote leading to better generalization performance but are also learned sooner in the time course of training, we conjecture that head class samples will be learned sooner than tail class samples and plot average proportion correct as a function of training epoch for each class to validate it.

As shown in Figure 2, all examples are learned during training, where jumps in the graph correspond to points at which the learning rate is reduced. However, interestingly, the tail class samples are learned most slowly and head class samples are learned most quickly. Indeed, learning speed of each class is monotonically related to its cardinality. An interesting observation from the learning speed plot in Figure 2 is that the stage-wise learning rate decay has a greater impact for tail class examples. To explore this phenomenon further, we trained models with constant learning rates of 0.1, 0.02, 0.01, and 0.001, results are shown in Figure 6 and Figure 7 in Appendix. In addition, our switching strategy can significantly accelerate the learning of tail classes based on the internal representations with a slight damage to head classes, indicating the dilemma of trade-off between head class pattern extraction and tail class memorizing in long-tailed learning.

5 Discussion

To further reveal the mechanism of the proposed method, more empirical studies on the proposed method are investigated. Our findings and take-home messages are as follows:

1. **Why effective?** As shown in Figure 3 and Figure 4, reductions on class size lead to both lower regularity and lower exposure in the training stage (more evidences in Section A.4). According to Jiang et al. (2020), a training stage with a relatively small learning rate may benefit for low-regularity sample memorization. Therefore, the intuitive combination of such sampling stage and certain frequency increasing re-sampling strategy for low-regularity classes naturally comes up.

   However, evidences in Table 9 show the clear superiority of the CR sampler over other sampling method in the latter stage, implying a mild frequency increasing for under-represented class may not reach the balanced tradeoff. Given that class regularities share most correlation with class number, CR sampling is the near optimal choice when class regularities are agnostic. In fact, Table 10 shows that performance of class-reversed sampling and class-regularity-reversed[1] sampling differs slightly.
2. **Need decoupling?** Decouple proposed by Kang et al. (2020) emphasizes the advantage of classifier re-training for long tail learning, which indeed exists in the standard training procedure. However, evidences in Section A.6 illustrates that the quality of the classification is inherently restricted by the previously learned representations. When tail samples are better represented, the decoupling between representation learning and classifier learning seems futile (more evidences in Table 12).
3. **Why efficiency?** The efficiency of the proposed method is obviously contributed by its parsimony, i.e. without any extra training stage like Kang et al. (2020) or extra sophisticated structure like Zhou et al. (2020). According to Figure 1, the testing accuracy reaches maximum with minimum epochs after the sampler switching.

   Moreover interestingly, when extra switching to another sampler like CB is added 5 epochs afterwards (see Table 7), the long-tailed classification performance drops significantly. In this case, as long as the representation learning at large learning rates ends, our method with switching strategy will end shortly afterwards while maintaining reasonable accuracy.

6 Conclusion

In this paper, we challenge the hypothesis by Kang et al. (2020) that the learning of feature representation and classifier should be completely decoupled in long-tailed visual recognition problem settings. Instead, we propose to switch the original instance-balanced sampling to class-reversed sampling in mini-batch stochastic gradient descent at the last few training epochs for memorizing tail samples. The presented approach exhibits unreasonable effectiveness and efficiency, leading to the proposition that the decoupling paradigm seems futile when tail samples are better represented. Further empirical findings show the inevitability to deal the trade-off between head class representing and tail class memorizing in the memorization stage with small learning rate.

[1] a data point from class j will be sampled proportionally to its irregularity r_j, shown in Table 10.

References

Peter L. Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *J. Mach. Learn. Res.*, 3:463–482, 2002.

Peter L. Bartlett, Dylan J. Foster, and Matus Telgarsky. Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems, NIPS*, pp. 6240–6249, 2017.

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

Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Aréchiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems (NeurIPS)*, pp. 1565–1576, 2019.

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge J. Belongie. Class-balanced loss based on effective number of samples. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pp. 9268–9277, 2019.

Qi Dong, Shaogang Gong, and Xiatian Zhu. Class rectification hard mining for imbalanced deep learning. In *IEEE International Conference on Computer Vision, ICCV*, pp. 1869–1878, 2017.

Chris Drummond, Robert C Holte, et al. C4. 5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling. In *Workshop on learning from imbalanced datasets II*, volume 11, pp. 1–8. Citeseer, 2003.

Vitaly Feldman. Does learning require memorization? a short tale about a long tail. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy (eds.), *Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC*, pp. 954–959, 2020.

Vitaly Feldman and Chiyuan Zhang. What neural networks memorize and why: Discovering the long tail via influence estimation. *CoRR*, abs/2008.03703, 2020.

Priya Goyal, Piotr Dollár, Ross B. Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch SGD: training imagenet in 1 hour. *CoRR*, abs/1706.02677, 2017.

Hui Han, Wenyuan Wang, and Binghuan Mao. Borderline-smote: A new over-sampling method in imbalanced data sets learning. In *Advances in Intelligent Computing, International Conference on Intelligent Computing, ICIC Proceedings, Part I*, volume 3644 of *Lecture Notes in Computer Science*, pp. 878–887, 2005.

Haibo He and Edwardo A. Garcia. Learning from imbalanced data. *IEEE Trans. Knowl. Data Eng.*, 21(9):1263–1284, 2009.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pp. 770–778, 2016.

Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. Learning deep representation for imbalanced classification. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pp. 5375–5384, 2016.

Muhammad Abdullah Jamal, Matthew Brown, Ming-Hsuan Yang, Liqiang Wang, and Boqing Gong. Rethinking class-balanced methods for long-tailed visual recognition from a domain adaptation perspective. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, pp. 7607–7616, 2020.

Nathalie Japkowicz and Shaju Stephen. The class imbalance problem: A systematic study. *Intell. Data Anal.*, 6(5):429–449, 2002.

Ziheng Jiang, Chiyuan Zhang, Kunal Talwar, and Michael C. Mozer. Exploring the memorization-generalization continuum in deep learning. *CoRR*, abs/2002.03206, 2020.

Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *8th International Conference on Learning Representations, ICLR*, 2020.

Kenji Kawaguchi and Haihao Lu. Ordered SGD: A new stochastic optimization framework for empirical risk minimization. In *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS*, volume 108, pp. 669–679, 2020.

Kenji Kawaguchi, Leslie Pack Kaelbling, and Yoshua Bengio. Generalization in deep learning. *CoRR*, abs/1710.05468, 2017.

Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *IEEE International Conference on Computer Vision, ICCV*, pp. 2999–3007, 2017.

Jialun Liu, Yifan Sun, Chuchu Han, Zhaopeng Dou, and Wenhui Li. Deep representation learning on long-tailed data: A learnable embedding augmentation perspective. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, pp. 2967–2976, 2020.

Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X. Yu. Large-scale long-tailed recognition in an open world. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pp. 2537–2546, 2019.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. Adaptive computation and machine learning. MIT Press, 2012.

Wanli Ouyang, Xiaogang Wang, Cong Zhang, and Xiaokang Yang. Factors in finetuning deep model for object detection with long-tail distribution. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pp. 864–873, 2016.

Junran Peng, Xingyuan Bu, Ming Sun, Zhaoxiang Zhang, Tieniu Tan, and Junjie Yan. Large-scale object detection in the wild from imbalanced multi-labels. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, pp. 9706–9715, 2020.

Ashish Rastogi. *McDiarmid's Inequality*. Springer US, 2011.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Fei-Fei Li. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.*, 115(3):211–252, 2015.

Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J. Gordon. An empirical study of example forgetting during deep neural network learning. In *7th International Conference on Learning Representations, ICLR*, 2019.

Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Learning to model the tail. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems (NeurIPS)*, pp. 7029–7039, 2017.

Tong Wu, Qingqiu Huang, Ziwei Liu, Yu Wang, and Dahua Lin. Distribution-balanced loss for multi-label classification in long-tailed datasets. *CoRR*, abs/2007.09654, 2020.

Liuyu Xiang and Guiguang Ding. Learning from multiple experts: Self-paced knowledge distillation for long-tailed classification. *CoRR*, abs/2001.01536, 2020.

Han-Jia Ye, Hong-You Chen, De-Chuan Zhan, and Wei-Lun Chao. Identifying and compensating for feature deviation in imbalanced deep learning. *CoRR*, abs/2001.01385, 2020.

Xi Yin, Xiang Yu, Kihyuk Sohn, Xiaoming Liu, and Manmohan Chandraker. Feature transfer learning for face recognition with under-represented data. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pp. 5704–5713, 2019.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *5th International Conference on Learning Representations, ICLR*, 2017.

Junjie Zhang, Lingqiao Liu, Peng Wang, and Chunhua Shen. To balance or not to balance: An embarrassingly simple approach for learning with long-tailed distributions. *CoRR*, abs/1912.04486, 2019.

Yaoyao Zhong, Weihong Deng, Mei Wang, Jiani Hu, Jianteng Peng, Xunqiang Tao, and Yaohai Huang. Unequal-training for deep face recognition with long-tailed noisy data. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pp. 7812–7821, 2019.

Bolei Zhou, Àgata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(6): 1452–1464, 2018.

Boyan Zhou, Quan Cui, Xiu-Shen Wei, and Zhao-Min Chen. BBN: bilateral-branch network with cumulative learning for long-tailed visual recognition. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pp. 9716–9725, 2020.

A Appendix

A.1 Missing Proofs and Derivations in Section 3.3

Proof of the general upper bound

Given a long-tailed dataset $\mathcal{D}$ sampled from the main dataset $\mathcal{S}$, we define:

$$\Phi(\mathcal{D}) = \sup_{\theta\in\Theta} \mathbb{E}_{(x,y)}[\ell(f_\theta(x), y)] - L(\theta; \mathcal{D}). \tag{8}$$

To apply McDiarmid's inequality (Rastogi (2011)) to provide the upper bound on $\Phi(\mathcal{D})$, we first show that $\Phi(\mathcal{D})$ satisfies the remaining condition of McDiarmid's inequality. Let $\mathcal{D}$ and $\mathcal{D}'$ be two datasets differing by exactly one point of an arbitrary index i_0, *i.e.*, $\mathcal{D}_i = \mathcal{D}'_i$ for all $i \neq i_0$ and $\mathcal{D}_{i_0} \neq \mathcal{D}'_{i_0}$. Then, the upper bound on $\Phi\left(\mathcal{D}'\right) - \Phi(\mathcal{D})$ is given as follows:

$$\begin{aligned}
\Phi\left(\mathcal{D}'\right) - \Phi(\mathcal{D}) &\leq \sup_{\theta\in\Theta} L(\theta; \mathcal{D}) - L\left(\theta; \mathcal{D}'\right) \\
&= \sup_{\theta\in\Theta}\Big(\sum_{j\in C}\sum_{i\in n_j} \frac{p_j}{n_j} L(\theta; \mathcal{D}) - \sum_{j\in C}\sum_{i\in n_j} \frac{p_j}{n_j} L\left(\theta; \mathcal{D}'\right)\Big) \\
&\leq \sup_{\theta\in\Theta} \frac{p_j}{n_j} \left|L_{ji_0}(\theta; \mathcal{D}) - L_{ji_0}\left(\theta; \mathcal{D}'\right)\right| \\
&\leq \frac{p_j}{n_j} M
\end{aligned} \tag{9}$$

Therefore, $|\Phi(\mathcal{D}) - \Phi\left(\mathcal{D}'\right)| \leq \frac{p_j}{n_j} M$ since we also have $\Phi(\mathcal{D}) - \Phi\left(\mathcal{D}'\right) \leq \frac{p_j}{n_j} M$. Thus, according to McDiarmid's inequality, for any $\delta > 0$, with probability at least $1 - \delta$ we have:

$$\Phi(\mathcal{D}) \leq \mathbb{E}_{\mathcal{D}}[\Phi(\mathcal{D})] + \sqrt{\sum_{j\in C} \frac{p_j^2}{n_j}} \sqrt{\frac{\ln(1/\delta)}{2}} M. \tag{10}$$

Therefore,

$$\begin{aligned}
&\mathbb{E}_{\mathcal{D}}[\Phi(\mathcal{D})] \\
&= \mathbb{E}_{\mathcal{D}}\left[\sup_{\theta\in\Theta} \mathbb{E}_{(x,y)}\left[\ell\left(f_\theta\left(x\right), y\right)\right] - L^s(\theta; \mathcal{D}) + L^s(\theta; \mathcal{D}) - L(\theta; \mathcal{D})\right] \\
&\leq \mathbb{E}_{\mathcal{D}}\left[\sup_{\theta\in\Theta} \mathbb{E}_{(x,y)}\left[\ell\left(f_\theta\left(x\right), y\right)\right] - L^s(\theta; \mathcal{D})\right] - \mathcal{Q}_n(\Theta; p, n) \\
&\leq \mathbb{E}_{\xi, D, \mathcal{D}'}\left[\sup_{\theta\in\Theta} \frac{1}{n}\sum_{i=1}^{n} \xi_i \left(\ell\left(f_\theta\left(\bar{x}'_i\right), \bar{y}'_i\right) - \ell\left(f_\theta\left(\bar{x}_i\right), \bar{y}_i\right)\right)\right] - \mathcal{Q}_n(\Theta; p, n) \\
&\leq 2\mathfrak{R}_n(\Theta) - \mathcal{Q}_n(\Theta; p, n).
\end{aligned} \tag{11}$$

where

$$\mathcal{Q}_n(\Theta; p, n) \quad = \quad \mathbb{E}_{\mathcal{D}}\left[\inf_{\theta\in\Theta}\sum_{j=1}^{C}\sum_{i=1}^{n_j}\left(\frac{p_j}{n_j}-\frac{1}{n}\right)\ell\left(f_\theta\left(x_i\right), y_i\right)\right]. \tag{12}$$

Therefore, for any $\delta > 0$, with probability at least $1-\delta$ we have:

$$\Phi(\mathcal{D}) \leq 2\Re_n(\Theta) - \mathcal{Q}_n(\Theta; p, n) + M\sqrt{\sum_{j\in C}\frac{p_j^2}{n_j}}\sqrt{\frac{\ln(1/\delta)}{2}}. \tag{13}$$

Substituting Equation 13 into Equation 8 we have:

$$\mathbb{E}_{(x,y)}[\ell(f_\theta(x), y)] \leq L(\theta; \mathcal{D}) + 2\Re_n(\Theta) - \mathcal{Q}_n(\Theta; p, n) + M\sqrt{\sum_{j\in C}\frac{p_j^2}{n_j}}\sqrt{\frac{\ln(1/\delta)}{2}} \tag{14}$$

Proof of the Theorem 1

We assume that:

1. The range of Θ is narrow due to the learning rate is small and the network has converged in the previous training process, so $\Re_n(\Theta) \to 0$ as $n \to \infty$, which has been shown to be satisfied for various models and sets Θ (Bartlett & Mendelson (2002); Mohri et al. (2012); Kawaguchi et al. (2017); Bartlett et al. (2017)).
2. Without loss of generality, the classes are sorted by cardinality in decreasing order, thus sampling weight p_j of each class j is ordered. p_1 is weight of the class with most samples and p_c is the weight of the class with least samples.
3. $L_i \leq L_j$ if $i < j$. This is an empirical conclusion that average loss of tail class samples are always higher than its of head class samples with a model trained by instance-balanced sampling.

Now let's compare the value of $\mathbb{E}^s_{(x,y)}[\ell(f_\theta(x), y)]$ and $\mathbb{E}_{(x,y)}[\ell(f_\theta(x), y)]$. Since both $M\sqrt{\frac{\ln(1/\delta)}{2n}}$ and $M\sqrt{\sum_{j\in C}\frac{p_j^2}{n_j}}\sqrt{\frac{\ln(1/\delta)}{2}}$ will disappear as $n \to \infty$, the core is to discuss the $\mathcal{Q}_n(\Theta; p, n)$.

We first consider the situation which only exchange the sampling rate for class 1 and class c under the instance-balanced sampling ($p_1 > p_c$). Here we have:

$$\begin{aligned}\sum_{j=1}^{C}\sum_{i=1}^{n_j}\left(\frac{p_j}{n_j}-\frac{1}{n}\right) &= \sum_{i=1}^{n_1}(\frac{p_c}{n_1}-\frac{1}{n})L_{1i} + \sum_{i=1}^{n_c}(\frac{p_1}{n_c}-\frac{1}{n})L_{ci} \\ &= n_1(\frac{p_c}{n_1}-\frac{1}{n})\overline{L_1} + n_c(\frac{p_1}{n_c}-\frac{1}{n})\overline{L_c} \\ &= (p_c - p_1)\overline{L_1} + (p_1 - p_c)\overline{L_c} \\ &= (p_c - p_1)(\overline{L_1} - \overline{L_c}) > 0\end{aligned} \tag{15}$$

Therefore, for high probability we can hold that $\mathcal{Q}_n(\Theta; p, n) > 0$ if we only exchange the sampling weight of the class 1 and class c. Naturally, $\mathcal{Q}_n(\Theta; p, n) > 0$ will always hold if we exchange the sampling weight of class i and class j ($i < j$), which is exactly how class-reversed sampling works.

Now let's promote our conclusion to more general situations, what will happen if we just change the sampling weight of one class instead of exchanging? Let's increase the p_c from p_c to p'_c, then every

p_j will change to $p'_j = p_j \frac{1-p'_c}{1-p_c}$ due to the constraint $\sum_{j=1}^{C} p_j = 1$, now we have:

$$
\begin{aligned}
\sum_{j=1}^{C}\sum_{i=1}^{n_j}\left(\frac{p_j}{n_j}-\frac{1}{n}\right) &= (\frac{p'_c}{n_c}-\frac{1}{n})n_c\overline{L_c} + \sum_{j=1}^{C-1}(\frac{p_j}{n_j}\frac{1-p'_c}{1-p_c}-\frac{1}{n})n_j\overline{L_j} \\
&= (\frac{p'_c}{n_c}-\frac{1}{n})n_c\overline{L_c} - \sum_{j=1}^{C-1}\frac{p'_c-p_c}{1-p_c}p_j\overline{L_j} \\
&\geq (p'_c-p_c)\overline{L_c} - \sum_{j=1}^{C-1}\frac{p'_c-p_c}{1-p_c}p_j\overline{L_{c-1}} \\
&\geq (p'_c-p_c)\overline{L_c} - \frac{p'_c-p_c}{1-p_c}\overline{L_{c-1}}\sum_{j=1}^{C-1}p_j \\
&\geq (p'_c-p_c)\overline{L_c} - \frac{p'_c-p_c}{1-p_c}\overline{L_{c-1}}(1-p_c) \\
&\geq (p'_c-p_c)(\overline{L_c}-\overline{L_{c-1}}) \geq 0
\end{aligned} \tag{16}
$$

Therefore, for high probability we can hold that $\mathcal{Q}_n(\Theta; p, n) > 0$ if we increase the sampling weight of the last class, and we can extend it to any tail class similarly.

To sum up, we can draw the conclusion that $\mathcal{Q}_n(\Theta; p, n) > 0$ holds if the sampling weight of tail classes is increased. Thus, with $n \to \infty$ and M is bounded, the upper bound on the expected error of class-reversed sampling is strictly lower than that for instance-balanced sampling if $\mathcal{Q}_n(\Theta; p, n) + L^s - L > 0$ or if $L^s - L > 0$. Based on the empirical experience that L^s and L will always be very close after training (no matter using only IB or only CR, the final training loss will always be small), $L^s - L \to 0$ holds after the complete training, thus $\mathcal{Q}_n(\Theta; p, n) + L^s - L > 0 \iff \mathcal{Q}_n(\Theta; p, n) > 0$ holds.

A.2 Other related work in Long-tailed Learning

Re-weighting losses. Re-weighting methods usually allocate different weights for training samples of each class to re-balance data distribution (Huang et al. (2016); **?**); Cao et al. (2019); Wu et al. (2020)). Cui et al. (2019) assigns weights to each class based on the effective numbers of samples instead of the proportional frequency. Further, Jamal et al. (2020) utilizes both effective numbers (Cui et al. (2019)) and conditional weights to augment the classic class-balanced learning by explicitly estimating the differences between the class-conditioned distributions with a meta-learning approach.

Transfer learning. Transfer learning methods (Wang et al. (2017); Zhong et al. (2019); Yin et al. (2019)) learn general patterns from head classes with abundant samples and transfer feature representations to help to recognize tail class samples. Recently, Liu et al. (2020) constructs each feature into a "feature cloud" to recover the intra-class diversity of tail classes. Xiang & Ding (2020) splits the entire dataset into several cardinality-adjacent subsets and acquires knowledge from multiple models trained on those subsets. However, transfer learning methods are usually complicated, which may be hard to applied to real-world scenarios.

Two-stage fine-tuning. Various methods (Ouyang et al. (2016); Cao et al. (2019); Liu et al. (2019); Peng et al. (2020)) are proposed to modify re-balancing for further improvements in long-tailed recognition. These methods usually separate training process into two single stages. In general, they train the networks with instance-balanced sampling in the first stage and exploit re-sampling or re-weighting methods at the second stage to fine-tune the network. More radically, Kang et al. (2020) re-train the classifier from scratch in a class-aware manner in the second stage with backbone fixed. Our work indicates that we can effectively learn the long-tailed distribution with only proper sampling strategies combination.

A.3 Experiment Details

A.3.1 Datasets

Long-tailed CIFAR-10 and CIFAR-100. Both CIFAR-10 and CIFAR-100 contains 60,000 images, with 50, 000 for training and 10,000 for validation with category number of 10 and 100, respectively. For fair comparisons, we use the long-tailed versions of CIFAR datasets as the same as those used in Zhou et al. (2020) with controllable degrees of data imbalance. Imbalance factor β is utilized to describe the severity of the long tail problem with the number of training samples for the most frequent class and the least frequent class, *e.g.*, $\beta = \frac{n_{max}}{n_{min}}$. We use β as 10, 50, and 100 in our experiments.

iNaturalist 2018. The iNaturalist species classification dataset is a large-scale real-world, naturally long-tailed dataset, suffering from extremely imbalanced label distributions. We choose the 2018 version in our experiments, which consists of 437,513 images from 8,142 categories. Note that, besides the extreme imbalance, the iNaturalist datasets also face the fine-grained problem. For fair comparisons, we utilize the official splits of training and validation images.

ImageNet-LT. ImageNet-LT is artificially truncated from their balanced versions so that the labels of the training set follow a long-tailed distribution. ImageNet-LT has 1000 classes and the number of images per class ranges from 1280 to 5 images. Note that the validation set is balanced of 1000 classes.

A.3.2 Implementation details

Implementations details on CIFAR. We adopt the plaining ResNet-32 (He et al. (2016)) as our model in all experiments. Standard mini-batch stochastic gradient descent (SGD) with momentum of 0.9, weight decay of 2×10^{-4} is utilized to optimize the whole network. We train all the models on one single NVIDIA 2080Ti GPU for 200 epochs with batch size of 64. The initial learning rate is set to 0.1 and the first five epochs is trained with the linear warm-up learning rate schedule (Goyal et al. (2017)). The learning rate is decayed at the 100th by 0.1. S is set to 1, which means we switch the instance-balanced sampling to class-reversed sampling at the 101st epoch.

Implementations details on iNaturalist. For fair comparisons, we utilize the plaining ResNet-50 (He et al. (2016)) as our backbone network in all experiments. We train all the models on eight NVIDIA 2080Ti GPUs with batch size of 512 for 90 epochs and 200 epochs, respectively. The initial learning rate is set to 0.05 and decayed by 0.1 at the 60th and 80th epoch for 90, 120th and 160th for 200. The batch size is 512 and S is set to 1, which is similar to experiments on CIFAR. For fair comparison with Decouple (Kang et al. (2020)), we also set the S as 10 and 40 respectively, which means to switch sampler and train it for additional 10 epochs after the same standard training procedure have done, with total number of training epochs as 100 epochs and 210 epochs.

Implementations details on ImageNet-LT. We adopt ResNet-50 and ResNext-50 as our backbone to analyze the effectiveness of our method. The initial learning rate is set to 0.2 and decayed by 0.1 at the 60th and 80th epoch for total 90 epochs. The batch size is 256 and S is set to 1, which is similar to experiments on CIFAR. For fair comparison with Decouple (Kang et al. (2020)), we also set the S as 10, which means to switch sampler and train it for additional 10 epochs after the same standard training procedure have done, with total number of training epochs as 100 epochs.

A.3.3 comparison methods

In experiments, we compare our method with four groups of methods:

Baseline methods. We employ plaining training with cross-entropy loss and focal loss (Lin et al. (2017)) as our baselines.

Re-weighting methods. For re-weighting methods, we compare with the CB-Focal (Cui et al. (2019)) and LDAM (Cao et al. (2019)), where effective numbers or margin-based generalization are utilized to alleviate the extreme data imbalance during training.

Two-stage fine-tuning strategies. To prove the effectiveness of our switching strategy, we compare it with the two-stage fine-tuning strategies proposed in Cao et al. (2019). Networks are trained with cross-entropy (CE) on imbalanced data first, and then are trained with class re-balancing strategy in the second stage. CE-DRW and CE-DRS refer to the two-stage baselines using re-weighting and re-sampling at the second stage. We also compare with Decouple (Kang et al. (2020)), which trains the network with instance-balanced sampling and uses class-balanced sampling to re-train the classifier in the second stage with backbone fixed.

State-of-the-art methods. For state-of-the-art methods, we compare with the recently proposed BBN(Zhou et al. (2020)) which achieves good classification accuracy on long-tailed datasets. BBN also utilizes class-reversed sampling to re-balance the feature extractor but it has a more complicated model structure, neglecting the proper combination of different data samplers itself.

A.4 Regularity

To investigate the memorization-generalization continuum in deep learning towards long-tailed distributions, we introduce the cumulative learned events and forgetting events (Toneva et al. (2019)) as follows:

Cumulative learned events. For sample $\{x_i, y_i\}$, $\hat{y}_i^t = \arg\max g(y_i|x_i;\theta^t)$ is the predicted label for sample x_i obtained after t epochs of SGD optimization. Let $acc_i^t = \mathbf{1}_{\hat{y}_i^t = y_i}$ be a binary variable indicating whether the sample is correctly classified at time epoch t, the cumulative learned events events at epoch t are defined as follows:

$$\mathbb{L}_i^t = \sum_{n=1}^{t} acc_i^n. \tag{17}$$

Forgetting events.: Let $for_i^t = \mathbf{1}_{acc_i^{t-1}=1, acc_i^t=0}$, the forgetting events of one sample $\{x_i, y_i\}$ at epoch t are defined as follows:

$$\mathbb{F}_i^t = \sum_{n=1}^{t} for_i^n. \tag{18}$$

Based on these notations, we intuitively give the metric to describe the regularity of one sample, which means with higher cumulative learned events as well as lower forgetting events, the higher regularity it will be and vice versa.

For Long-tailed CIFAR-10 with imbalance ratio 50, we run the ResNet-32 for 10 runs and plot the averaged cumulative learned events and forgetting events of each sample grouped by its class, as shown in Figure 3. We surprisingly find that the clustering degree of samples is almost proportional to its cardinality. To explore this phenomenon further, we plot the same events of same samples when learning the standard CIFAR-10 with class-balanced distributions in Figure 4. Compared with the same samples under class-balanced distributions, long-tailed distribution samples show different properties of each class: higher degree of clustering of head classes (cls0, cls1) and lower degree of clustering of tail classes (cls6, cls7, cls8, cls9). Differences between them indicate that the cardinality of one class can significantly affect the regularity itself during training: *regularity of the same training samples will be sharply decreased with the reduction of class cardinality*, which is easy to understand: the more samples one class have, the higher regular it will be.

In order to analysis this phenomenon, we propose a novel metric to quantize the regularity of each class. For class j containing n_j samples, regularity event of each sample $\{x_i, y_i\}$ can be denoted by its cumulative learned events and forgetting events as $r_{i,j} = \{\mathbb{L}_i^T, \mathbb{F}_i^T\}$, which is a point on the two-dimensional plane. There are three sub-procedures to calculate the regularity of each class j:

1) Let $\{(\mathbb{L}_i^T, \mathbb{F}_i^T)|1 \le i \le n_j\} = [\boldsymbol{LF}]$ denote the regularity set of class j, we calculate the covariance matrix as follows:

$$\boldsymbol{C}_j = \begin{bmatrix} \mathbb{E}[(\boldsymbol{L} - \mathbb{E}[\boldsymbol{L}])(\boldsymbol{L} - \mathbb{E}[\boldsymbol{L}])] & \mathbb{E}[(\boldsymbol{L} - \mathbb{E}[\boldsymbol{L}])(\boldsymbol{F} - \mathbb{E}[\boldsymbol{F}])] \\ \mathbb{E}[(\boldsymbol{F}^T - \mathbb{E}[\boldsymbol{F}])(\boldsymbol{L} - \mathbb{E}[\boldsymbol{L}])] & \mathbb{E}[(\boldsymbol{F} - \mathbb{E}[\boldsymbol{F}])(\boldsymbol{F} - \mathbb{E}[\boldsymbol{F}])] \end{bmatrix} \in R^{2\times 2}, \tag{19}$$

where $\mathbb{E}$ is the expectation.

2) calculate the F-norm of $\boldsymbol{C}_j$:

$$||\boldsymbol{C}_j||_F = \sqrt{\sum_{m=1}^{2}\sum_{n=1}^{2} |c_{mn}|^2}. \tag{20}$$

3) normalize the F-norm by its cardinality:

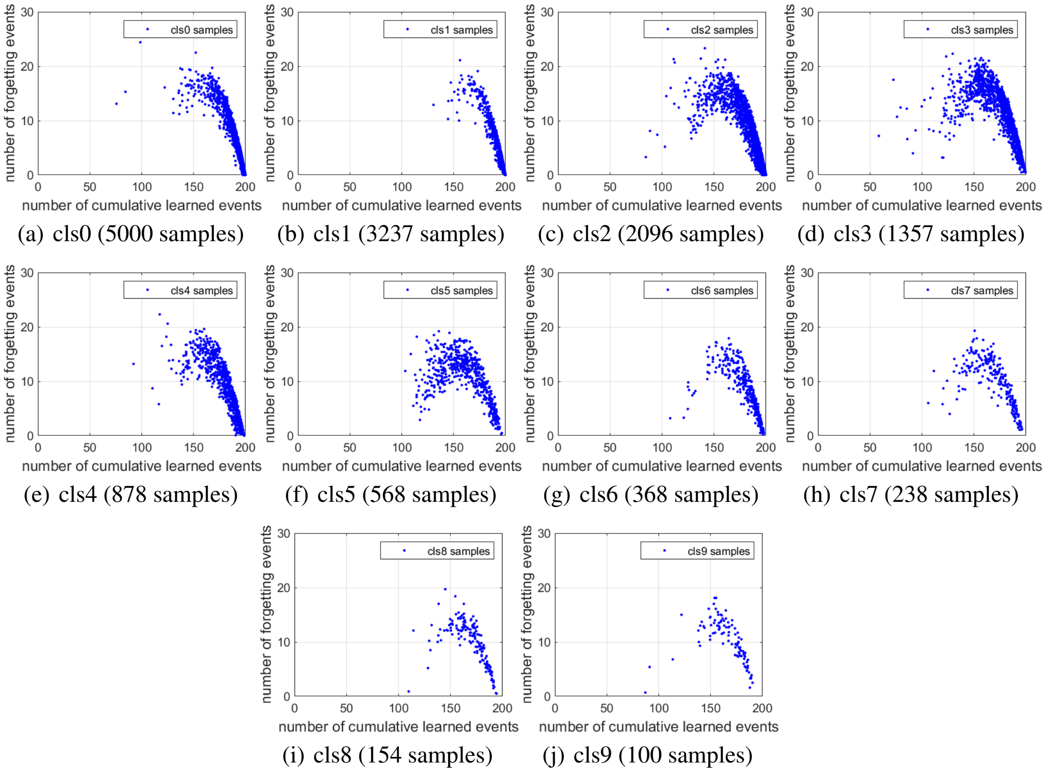
$$I_j = ||\boldsymbol{C}_j||_F / n_j. \tag{21}$$

This metric essentially indicates the deviation of each class, so we name it $Irregularity$.

As shown in Table 8, the *Irregularity* is almost proportionally to the reciprocal of its cardinality, which is consistent with our visual perception. To further validate the correlation between regularity and its cardinality, we exploit the Pearson correlation coefficient. Let $\boldsymbol{I} = \{I_i | 1 \leq i \leq C\}$ be the regularity set of all classes and $\boldsymbol{N} = \{n_i | 1 \leq i \leq C\}$ be the cardinality of all classes, we calculate the Pearson correlation coefficient as follows:

$$P = \frac{\sum \boldsymbol{N}\boldsymbol{I} - \frac{\sum \boldsymbol{N} \sum \boldsymbol{I}}{C}}{\sqrt{(\sum \boldsymbol{N}^2 - \frac{(\sum \boldsymbol{N})^2}{C})(\sum \boldsymbol{I}^2 - \frac{(\sum \boldsymbol{I})^2}{C})}}. \tag{22}$$

As Table 8 shows, the Pearson coefficient is **-0.6112**, indicating cardinality and regularity are significantly negatively correlated.



(a) cls0 (5000 samples) (b) cls1 (3237 samples) (c) cls2 (2096 samples) (d) cls3 (1357 samples)

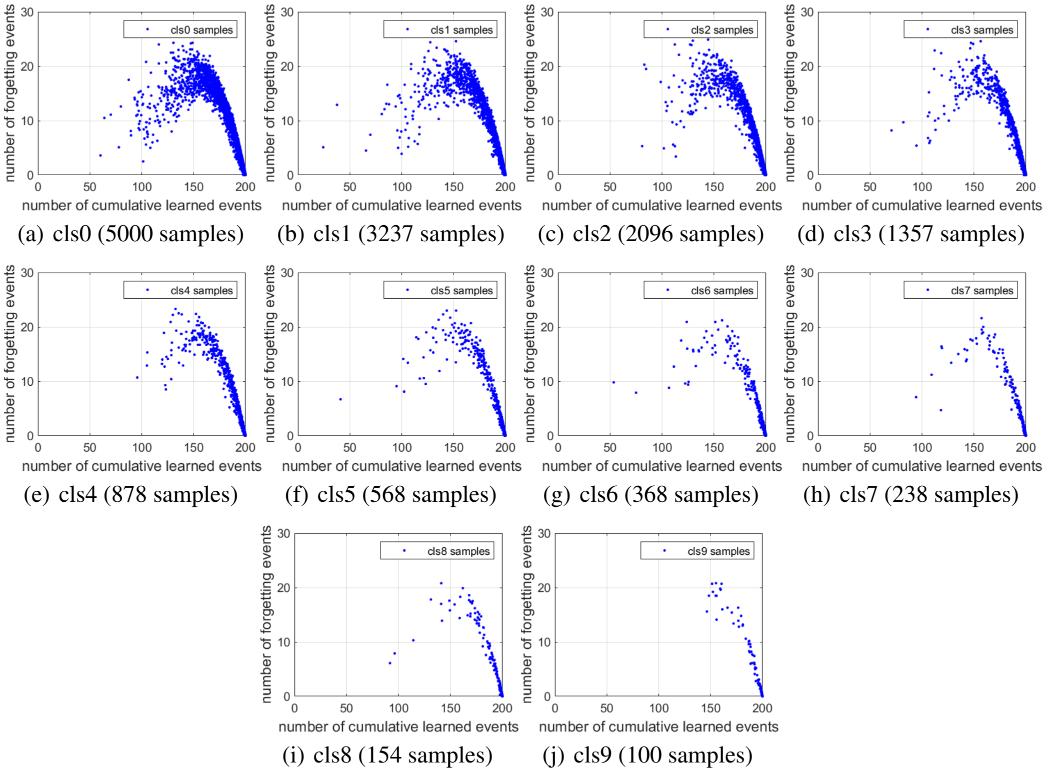
(e) cls4 (878 samples) (f) cls5 (568 samples) (g) cls6 (368 samples) (h) cls7 (238 samples)

(i) cls8 (154 samples) (j) cls9 (100 samples)

Figure 3: Cumulative learned events and forgetting events of each sample of Long-tailed CIFAR-10 with imbalance ratio 50. The regularity of one class will be higher with more samples gathering in the lower right corner of the picture.

A.5 Combinations of Sampling Strategies

In order to find the optimal sampler combination before and after switching, we conduct comprehensive experiments on long-tailed CIFAR-10 (imbalance ratio: 50) with combinations of different data samplers used in different stages. As shown in Table 9, our strategy, which switches instance-balanced sampling to class-reversed sampling in the small learning rate stage, achieves the best performance across all experimental settings. We draw the same conclusion with Decouple that *instance-balanced sampling gives the most generalizable representations*, for using instance-balanced in the first stage always performs better than other results. In addition, switching to class-reversed sampling can always bring a significant improvement no matter what samplers used in the



(a) cls0 (5000 samples) (b) cls1 (3237 samples) (c) cls2 (2096 samples) (d) cls3 (1357 samples)

(e) cls4 (878 samples) (f) cls5 (568 samples) (g) cls6 (368 samples) (h) cls7 (238 samples)

(i) cls8 (154 samples) (j) cls9 (100 samples)

Figure 4: Cumulative learned events and forgetting events of each sample of standard CIFAR-10. Samples here are in one-to-one correspondence with samples in Figure 3.

Table 8: Quantitative results of the regularity of each class on long-tailed CIFAR-10 (imbalance ratio: 50). All indexes are calculated based on the cumulative learned events and forgetting events.

Class	Cardinality	F-norm	Irregularity (F-norm / Cardinality)	Pearson
0	5000	85.7341	0.0171	-0.6112
1	3237	44.3574	0.0137	
2	2096	296.3582	0.1414	
3	1357	397.6501	0.2930	
4	878	267.5524	0.3047	
5	568	447.2937	0.7875	
6	368	253.5138	0.6889	
7	238	355.5270	1.4938	
8	154	279.5059	1.8150	
9	100	363.4100	3.6341	

first stage, except class-reversed sampling on the long-tailed CIFAR-100 with imbalance ratio 100 and 50 (see the last row in Table 9). We conjecture this is because class-reversed sampling can not learn the general representations on such extreme imbalanced data, since it mainly samples from the tail classes with low cardinality. Without generalizable representation and seeing samples from other classes, network can not generalize well across all classes.

As shown in Table 10, the performance of sampling weights based on cardinality and regularity of each class differs slightly, indicating the class-reversed sampling is the near optimal choice when class regularities are agnostic.

Table 9: Comprehensive results on long-tailed CIFAR-10 (imbalance ratio: 50) with combinations of different data samplers used in different stages.

Sampling Strategy Combination	Long-tailed CIFAR-10			Long-tailed CIFAR-100		
Imbalance ratio	100	50	10	100	50	10
IB $\Longrightarrow$ IB	73.1	77.9	86.4	40.7	44.9	57.2
IB $\Longrightarrow$ CB	77.1	81.9	87.9	44.2	48.7	59.2
IB $\Longrightarrow$ CR	**79.7**	**82.9**	**88.4**	**44.7**	**49.5**	**59.5**
CB $\Longrightarrow$ IB	66.5	73.8	86.7	33.3	37.1	55.0
CB $\Longrightarrow$ CB	73.0	78.5	87.3	36.2	40.3	56.9
CB $\Longrightarrow$ CR	74.8	80.7	87.9	38.6	42.4	57.8
CR $\Longrightarrow$ IB	63.8	74.8	85.2	24.7	28.7	51.3
CR $\Longrightarrow$ CB	64.2	72.7	86.3	24.7	29.2	52.5
CR $\Longrightarrow$ CR	68.4	76.3	86.8	22.9	28.0	53.2

Table 10: Sampling weights in the switching stage based on cardinality and irregularity respectively on long-tailed CIFAR-10 (imbalance ratio: 50).

Sampling Weight	Test accuracy
cardinality	**82.9**
irregularity	82.8

A.6 Do We Need Decoupling?

To further compare our method with Decouple, we investigate the factors of fixing feature extractor and re-training classifier towards learning long-tailed distributions, which are adopted in Decouple. From the results shown in Table 11, the following observations can be made:

- *Joint training is better.* Training the backbone and the classifier jointly always performs better than fixing the backbone. This phenomenon indicates that although instance-balanced sampling gives the most generalizable representations, it is not good enough. Fine-tuning the backbone with low-regularity tail class samples in the small learning rate stage can significantly improve its representation ability across tail classes.
- *Re-training is matter when no switching.* When training with the switching strategy, results with re-training or without re-training the classifier are much similar (see rows with CB, CR as the switching sampler). However, interestingly, re-training the classifier can bring improvements in the standard training procedure (see rows with IB as the switching sampler). We speculate that model trained by uniform instance-balanced sampling would have a strong bias towards tail classes in both backbone and classifier. Re-training classifier based on the learned general representations can alleviate it.
- *Switching and joint training are complementary.* We compare the results of only switching to only joint training, finding that while switching samplers and joint training can bring improvements respectively, their combination can improve the performance further. Fine-tuning with class-balanced or class-reversed distributions can boost the generalization ability further.

Further, we valid the quality of features learned by standard training procedure and our switching training procedure in Table 12, just like Decouple. Although a slightly lower with IB, re-training based on our feature can bring a significant improvements compare with stand features. These results also indicate a disadvantage of Decouple: performance of re-training classifier is depend on the performance of feature extractor. Once the feature representation is sub-optimal, the re-trained classifier is sub-optimal.

To validate our method could reach a better balance under bias-variance trade-off, we calculate the total error of each method in Table 13. The higher accuracy as well as lower total error indicate our switching performs better in the challenging trade-off compared with other methods.

Table 11: Comparisons between Decouple learning paradigm and our learning paradigm on long-tailed CIFAR-10 (imbalance ratio: 50), where Decouple indicates fixing the backbone and re-train the classifier from scratch while we continue to joint train both of them.

First Sampler	S	Switching Sampler	Joint Training	Re-training Classifier	Test accuracy
IB	1	IB			76.3
			✓		77.9
				✓	77.7
			✓	✓	78.9
		CB			81.9
			✓		81.9
				✓	81.8
			✓	✓	81.9
		CR			81.4
			✓		**82.9**
				✓	81.6
			✓	✓	82.4

Table 12: Feature quality of Decouple learning paradigm and our switching learning paradigm on long-tailed CIFAR-10 (imbalance ratio: 50). We firstly train the model with standard and switching procedure respectively, then re-train the classifier with different data samplers with backbone fixed.

Feature	Re-training	Test accuracy
Standard	IB	77.7
	CB	80.7
	CR	82.2
Our	IB	77.0
	CB	81.4
	CR	**82.6**

Table 13: Total error (bias^2+variance) of different methods on the test set of long-tailed CIFAR-10 (imbalance ratio: 50).

Method	Test Accuracy ↑	Bias^2 ↓	Variance ↓	Total Error ↓
Cross Entropy (IB only)	0.779	0.049	0.168	0.217
Cross Entropy (CR only)	0.763	0.056	0.183	0.239
Decouple-cRT	0.807	0.037	0.148	0.185
BBN	0.822	0.032	0.146	0.178
Ours (S = 1)	**0.829**	**0.029**	**0.142**	**0.171**

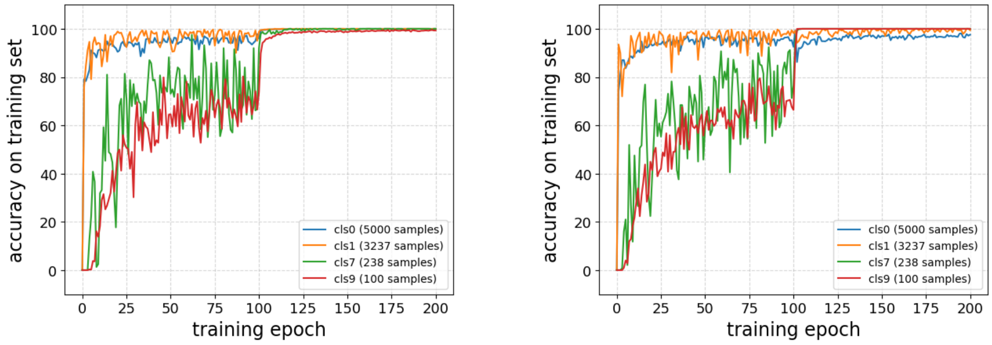
A.7 Learning Rate Scheduling

In Section 4.3 we visualize the learning speed of different classes to reveal the effectiveness of stage-wise constant learning rate scheduler during training. The observations lead to an interesting hypothesis for explaining why we should mainly memorizing tail class samples in the small learning rate stage. We provide more details here.

Figure 5 shows the learning speed of 4 selected classes with SGD using stage-wise constant learning rate scheduling. This is the same as Figure 2, replicated here for easy comparison. In Figure 6 we show the learning speeds of 4 selected classes trained with SGD using constant learning rate scheduling with the standard training procedure. The 4 panels show the results of different values of constant learning rate used in training. It is observed that faster convergence are achieved with smaller learning rate (see 0.1, 0.02 and 0.01). While the learning rate is so small, *e.g.*, 0.001, the learning speed of each class is significantly slowed down.

In Figure 7 we show the learning speeds of of our switching training procedure trained with SGD using constant learning rate scheduling. Similar to Figure 6, proper small learning rate could accelerate the convergence, with higher and more stable accuracy. It it worth noting that switching to class-reversed sampler always improves the accuracy of tail classes, but will damage the representative ability of head classes to some extent. Stage-wise constant learning bring the smallest damage to the head class representations, showing the necessity of building generalization representations first. Quantitative results of both standard training and switching training are shown in Table 14.

Here we manage to explain why class-reversed sampler is effective. The reason that switching to class-reversed sampler performs well is that it delayed the learning of low-regularity samples (tail classes samples) to later small learning rate stages. In the first stage, when almost head class samples are learned, the patterns and structures discovered in those high-regularity samples can be used to build a generalizable representation. In later stage, network is able to learn or memorize low-regularity samples of tail classes based on the representations from a clean subset of high-regularity samples. In addition, learning or memorizing tail class samples will not seriously disrupt the learned representation as the learning rate is much smaller than the earlier stages. In contrast, standard learning procedure without switching could not focus on the tail class samples since the extreme data imbalance, leading under-representation for tail classes, while SGD with (small) constant learning rate learns the examples across all classes fairly quickly, which can not learn the generalizable representation from high-regularity samples of head classes before.



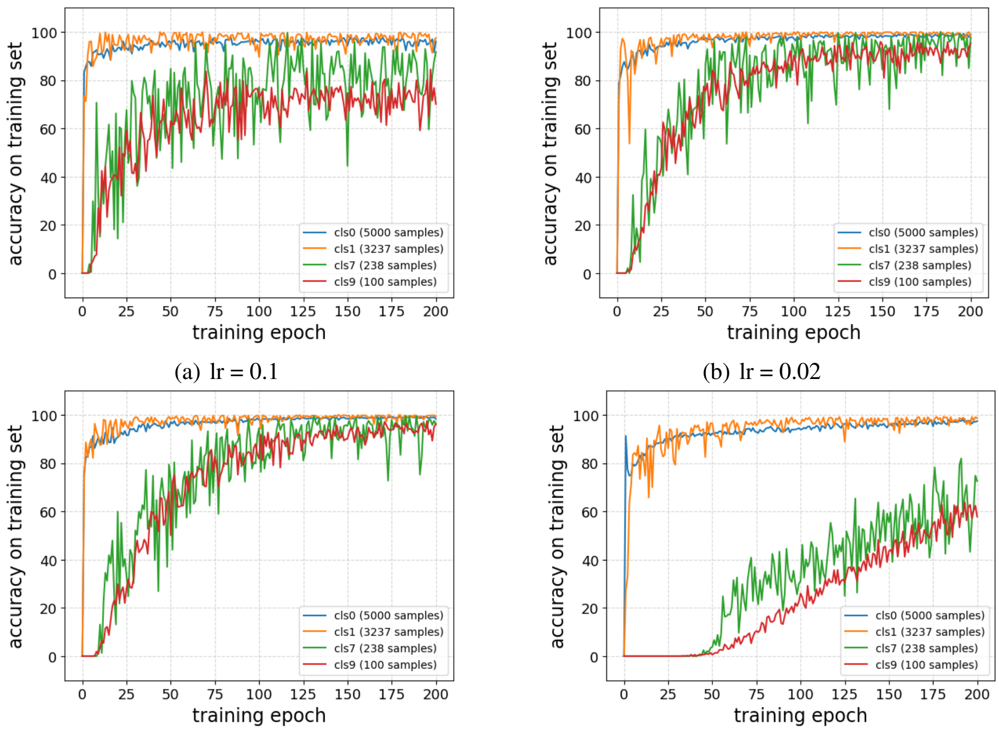
(a) standard

(b) ours

Figure 5: Learning speed of examples of 4 selected class with SGD using stage-wise constant learning rate. Left: standard training procedure. Right: our switching training procedure.

Table 14: Test performance of models trained with various learning rate schedulers on long-tailed CIFAR-10 (imbalance ratio: 50).

Standard			Our		
Optimizer	Learning Rate	Test Accuracy	Optimizer	Learning Rate	Test Accuracy
SGD	Stage-wise	**77.9**	SGD	Stage-wise	**82.9**
SGD	0.1	75.5	SGD	0.1	77.7
SGD	0.02	76.1	SGD	0.02	78.3
SGD	0.01	75.0	SGD	0.01	77.7
SGD	0.001	65.9	SGD	0.001	66.1



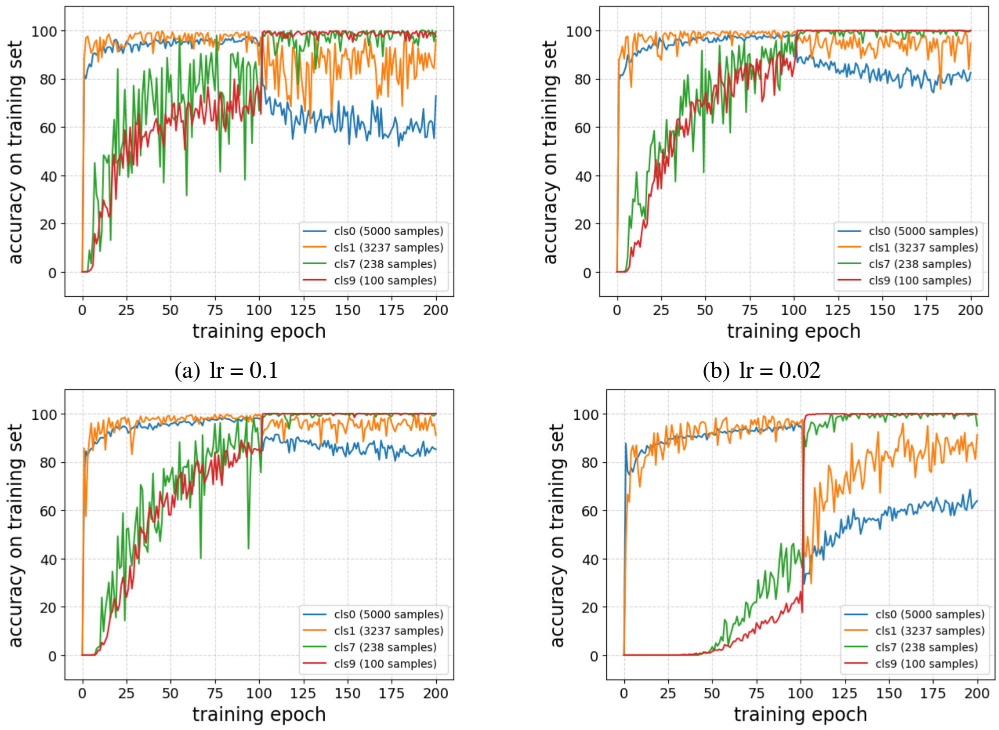
(a) lr = 0.1

(b) lr = 0.02

(c) lr = 0.01

(d) lr = 0.001

Figure 6: Learning speed of examples of 4 selected classes with SGD using constant learning rate with standard training strategy. The 4 different learning rates correspond to the constants used in the stage-wise scheduler.



(a) lr = 0.1

(b) lr = 0.02

(c) lr = 0.01

(d) lr = 0.001

Figure 7: Learning speed of examples of 4 selected classes with SGD using constant learning rate with our training strategy. The 4 different learning rates correspond to the constants used in the stage-wise scheduler.