The Majority Can Help The Minority: Context-rich Minority Oversampling for Long-tailed Classification

Seulki Park[1]* Youngkyu Hong[2] Byeongho Heo[2] Sangdoo Yun[2] Jin Young Choi[1]
[1]ASRI, ECE., Seoul National University [2]NAVER AI Lab

Abstract

The problem of class imbalanced data lies in that the generalization performance of the classifier is deteriorated due to the lack of data of the minority classes. In this paper, we propose a novel minority over-sampling method to augment diversified minority samples by leveraging the rich context of the majority classes as background images. To diversify the minority samples, our key idea is to paste a foreground patch from a minority class to a background image from a majority class having affluent contexts. Our method is simple and can be easily combined with the existing long-tailed recognition methods. We empirically prove the effectiveness of the proposed oversampling method through extensive experiments and ablation studies. Without any architectural changes or complex algorithms, our method achieves state-of-the-art performance on various long-tailed classification benchmarks. Our code will be publicly available at link.

1. Introduction

Real-world data are likely to be inherently imbalanced [11, 18, 26, 27]. If models are trained on such an imbalanced dataset, they would be biased toward majority classes and tend to have poor generalization ability on recognizing minority classes (*i.e.,* overfitting).

A simple and straightforward method to overcome the class imbalance problem is to repeatedly oversample the minority classes [6,41]. However, these naive oversampling may rather intensify the overfitting problem since the repeatedly selected samples have less diversity with almost similar image contexts [36]. For example, consider a minority class of 'Snow goose' where the geese always stand upon grasses in the training images. If samples are drawn from these limited training samples [41] or even if new samples are produced by interpolating within the class [6], only **context-limited** images will be created as in Figure 1. Our goal is to solve the aforementioned problem by introducing a simple **context-rich** oversampling method.

*Works done while doing an internship at NAVER AI Lab.

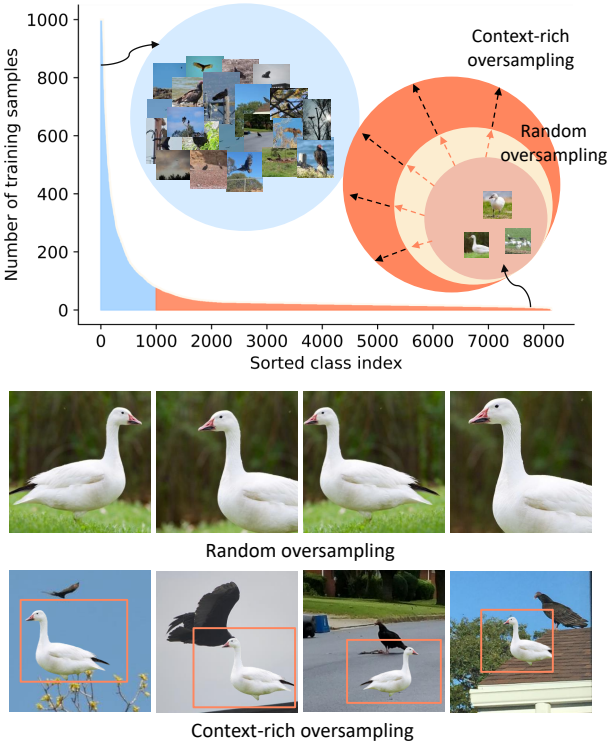

Figure 1. **Concept of context-rich minority oversampling**. In the real-world long-tailed dataset iNaturalist 2018 [18], the number of samples from head class and tail class is extremely different. Naive random oversampling method repeatedly produces context-limited images from minority classes. We propose a novel context-rich oversampling method to generate diversified minority images. Our key idea is to bring rich contexts from majority samples to minority samples.

We pay attention to the characteristics of long-tailed distributions; that is, majority class samples are data-rich and information-rich. Unlike the existing re-sampling methods that ignore (*i.e.*, undersample) majority samples, we use the affluent information of the majority samples to generate new minority samples. Specifically, our idea is to leverage the rich major-class images as the background for the newly created minor-class images. Figure 1 illustrates the

concept of our proposed context-rich oversampling strategy. Given an original image from a minority class, the object is cropped in various sizes and pasted to the various images from majority classes. Then, we can create images with more diverse contexts (*e.g.*, 'Snow goose' images with the sky, road, roof, crows, etc). Since this is an interpolation of the majority and minority class samples, it generates diversified data around the decision boundary, and as a result, it improves generalization performance for minority classes.

To this end, we adopt an image-mixing data augmentation, CutMix [46]. Reminding our key idea is transferring rich contexts from majority to minority samples, we propose a simple and effective data sampling strategy to generate new minority-centric images with majority's contexts. However, naive use of CutMix may exacerbate the overfitting problem towards majority classes since it may generate more majority-centric samples than minority samples. We solve this problem by sampling the background images and the foreground patches from different distributions to achieve the desired minority oversampling.

Our key contributions can be summarized as follows: (1) We propose a novel context-rich minority oversampling that generates various samples by leveraging the rich context of the majority classes as background images. (2) Our method requires little additional training cost and can be easily integrated into various end-to-end deep learning algorithms for long-tailed recognition. (3) We show that significant performance improvements can be achieved by applying the proposed oversampling to existing commonly used loss functions without any architectural changes or complex algorithms, and still achieve state-of-the-art performance. (4) We empirically prove the effectiveness of the proposed oversampling method through extensive experiments and ablation studies. We believe that our study can be a useful and universal minority oversampling method in long-tailed classification research.

2. Related Work

2.1. Long-tailed Recognition

Re-weighting methods. Re-weighting aims to assign different weights to training samples to adjust their importance either on class level or instance level. Class-level re-weighting methods include re-weighting samples by inverse class frequency [19, 45], CB loss [9], LDAM loss [5], Balanced softmax [34], LADE loss [17]. Instance-level re-weighting methods include Focal loss [25] and Influence-balanced loss [32].

Resampling methods. Resampling methods aim to modify the training distributions to decrease the level of imbalance [20]. Resampling methods include undersampling and oversampling. Undersampling methods [41, 49] which discard the majority samples can lose some valuable information, and it is infeasible when the imbalance between classes is too high.

The simplest form of oversampling is random oversampling (ROS) [3, 41], which oversamples all minority classes until class balance is achieved. This method is simple and can be easily used for any algorithm, but since the same sample is repeatedly drawn, it can lead to overfitting [36]. As a more advanced method, Synthetic Minority Over-sampling Technique (SMOTE) [6], which oversamples minority samples by interpolating between existing minority samples and their nearest minority neighbors, was proposed. After the success of SMOTE, several variants have been developed: Borderline-SMOTE [14] which oversamples the minority samples near class borders, and Safe-level-SMOTE [4], which defines safe regions not to oversample samples with different classes. These methods have been widely used in classical machine learning algorithms, but there are difficulties in using them for large-scale image datasets due to the high computational complexity of calculating K-Nearest Neighbor for every sample. To solve this issue, Generative Adversarial Minority Oversampling (GAMO) [31] produces new minority samples by training a convex generator, inspired by the success of generative adversarial networks (GANs) [12] in image generation. However, training the generator incurs a lot of additional training cost, and GAMO can also suffer from the infamous mode collapse of GANs [2]. Another recent line of research is oversampling in the feature space rather than input space: Deep Over-sampling (DOS) [1], Feature-space Augmentation (FSA) [8], and Meta Semantic Augmentation (MetaSAug) [24]. These methods aim to augment minority classes in the feature space by sampling from the linear subspace in-class neighbors [1], using learned features from pretrained networks [8], or using implicit semantic data augmentation (ISDA) algorithm [44]. However, DOS [1] requires finding the nearest neighbors in feature space, FSA [8] requires the pre-trained feature subnetwork and the classifier for feature augmentation procedure. Lastly, MetaSAug [24] demands additional uniform validation samples that outnumber the number of samples in tail classes and hundreds and thousands of iterations for training. Consequently, these methods are less cost-efficient and technically difficult to perform. On the other hand, our method oversamples diverse minority samples with a simple data augmentation technique and outperforms all previous methods while maintaining reasonable training costs.

Other long-tailed methods. Recently, significant improvement has been achieved by two-stage algorithms: Deferred re-weighting (DRW) [5], classifier re-training (cRT) and learnable weight scaling (LWS) [22], and Mixup Shifted Label-Aware Smoothing model (MiSLAS) [50]. Meanwhile, bilateral branch network (BBN) [51] uses an additional network branch for re-balancing, and RIDE [43]

uses multiple branches named experts, each learning to specialize in different classes. Another line of recent research employs meta-learning methods; Meta-Weight-Net [37] learns an explicit loss-weight function, and Meta Sampler [34] is used to estimate the optimal class sample rate.

2.2. Data Augmentation and Mixup methods

Spatial-level augmentation methods have shown good performance in computer vision fields. Cutout [10] removes random regions while CutMix [46] fills the removed regions with patches from another training image. In addition, Mixup methods [38,42,48] linearly interpolate two images in a training dataset. Since data augmentation method is closely related to oversampling methods, some recent long-tailed recognition methods have used Mixup method. Zhou et al. [51] compares Mixup as a baseline method, and MiSLAS [50] uses Mixup in their Stage-1 training. However, these methods naively use Mixup, and little work has been done to explore appropriate data augmentation techniques for a long-tailed dataset. Recently, for an imbalanced dataset, Remix [7] assigns the label in favor of the minority classes when mixing two samples. Unlike these methods, we propose to sample images from different distributions, considering the specificity of long-tailed data.

3. Context-rich Minority Oversampling

3.1. Algorithm

We propose a new oversampling method called Context-rich Minority Oversampling (CMO). CMO utilizes the contexts of the majority samples to diversify the limited context of minority samples. In other words, the background images are sampled from majority classes and combined with foreground images of minority classes. Let $x \in \mathbb{R}^{W\times H\times C}$ and y denote a training image and its label, respectively. Then we aim to generate a new sample $(\tilde{x}, \tilde{y})$ by combining two training samples (x^b, y^b) and (x^f, y^f). Here, the image x^b is used as a background image, and the image x^f provides the foreground patch to be pasted on (x^b, y^b).

For the image combining method, we choose CutMix [46] data augmentation based on its simplicity and effectiveness. Following CutMix [46] setting, image and label pairs are augmented as

$$\begin{aligned} \tilde{x} &= \mathbf{M} \odot x^b + (\mathbf{1} - \mathbf{M}) \odot x^f \\ \tilde{y} &= \lambda y^b + (1-\lambda) y^f \end{aligned} \tag{1}$$

where $(\mathbf{1} - \mathbf{M}) \in \{0,1\}^{W\times H}$ denotes a binary mask indicating where to select the patch and paste into a background image. $\mathbf{1}$ is a binary mask filled with ones, and $\odot$ is element-wise multiplication. The combination ratio $\lambda \in \mathbb{R}$ between two images is sampled from the beta distribution $Beta(\alpha, \alpha)$. For sampling the mask and its coordinates, we follow the setting of original CutMix [46].

Since CutMix is originally designed for data augmentation on a class balanced dataset, Eq. 1 does not represent majority or minority class of samples. To change it to CMO, we put sampling data distributions for foreground (x_f, y_f) and background samples (x^b, y^b). In our design, the background samples (x^b, y^b) should be biased to majority classes. So, we sample the background samples from the original data distribution P. Meanwhile, the foreground samples (x^f, y^f) are sampled from minor-class-weighted distribution Q to be biased to the minority classes. In short, CMO consists of data sampling from two distributions $(x^b, y^b) \sim P$ and $(x^f, y^f) \sim Q$, and image combining of Eq. 1. The pseudo-code of the training procedure is presented in Algorithm 1.

Algorithm 1 Context-rich Minority Oversampling (CMO)

Require: Dataset $\mathcal{D}_{i=1}^N$, model parameters θ, P, Q, any loss function $L(\cdot)$.

Randomly initialize θ.
Sample weighted dataset $\tilde{\mathcal{D}}_{i=1}^N \sim Q$.
for epoch $= 1, \ldots, T$ **do**
  for batch $i = 1, \ldots, B$ **do**
    Draw a mini-batch (x_i^b, y_i^b) from $\mathcal{D}_{i=1}^N$
    Draw a mini-batch (x_i^f, y_i^f) from $\tilde{\mathcal{D}}_{i=1}^N$
    $\lambda \sim Beta(\alpha, \alpha)$
    $\tilde{x}_i = \mathbf{M} \odot x_i^b + (\mathbf{1} - \mathbf{M}) \odot x_i^f$
    $\tilde{y}_i = \lambda y_i^b + (1-\lambda) y_i^f$
    $\theta \leftarrow \theta - \eta \nabla L((\tilde{x}_i, \tilde{y}_i); \theta)$
  end for
end for

3.2. Minor-class-weighted distribution Q

To sample the foreground image from minority classes, we design the minor-class-weighted distribution Q by taking the idea from the re-weighting methods. The re-weighting approach, dating back to the classical importance sampling method [21], has provided a way how to assign appropriate weights to samples. Commonly used sampling strategies include ones which give a weight inversely proportional to class frequency [19, 45], a smoothed class frequency [29, 30], or the effective number [9].

Let n_k be the number of samples in k-th class, then for the entire C classes, the total number of samples is $N = \sum_{k=1}^{C} n_k$. Then, the generalized sampling probability for k-th class can be defined by

$$q(r,k) = \frac{1/n_k^r}{\sum_{k'=1}^{C} 1/n_{k'}^r}, \tag{2}$$

where the k-th class has a sampling weight inversely proportional to n_k^r. As r increases, the weight of the minor class becomes increasingly larger than that of the major class.

By adjusting the value of r, we can examine diverse sampling strategy. Setting $r = 1$ is to use the inverse class frequency [19,45] while setting $r = 1/2$ is to use the smoothed inverse class frequency as in [29, 30]. We can also use the effective number [9] instead of n_k^r which is defined as

$$E(k) = \frac{(1 - \beta^{n_k})}{(1 - \beta)}, \tag{3}$$

where $\beta = (N - 1)/N$. Since CMO is a new approach for long-tailed classification, it is hard to predict the performance of each sampling strategy for CMO. So, we evaluate the different sampling strategies on the CIFAR-100-LT [23] and select the best strategy $q(1, k)$ for the minor-class-weighted distribution Q of CMO. The experimental results are shown in Table 9 of the experimental section.

3.3. Regularization effect of CMO

A recent study [50] has reported that models trained on long-tailed datasets are more over-confident than the models trained on balanced data. Also, the study shows that the performance of long-tailed classification can be improved by solving over-confident issue. CMO also can be interpreted as a way to mitigate the over-confident issue for long-tailed classification. Inherited from CutMix, CMO uses a soft-target label $\tilde{y}$ as mentioned in Eq. 1. The soft-target label penalizes over-confident outputs, similar to the label smoothing regularization [39]. Therefore, we argue that CMO contributes not only to minority sample generation but also to mitigating the over-confident issues, which enable an impressive performance improvement on diverse long-tail settings. We will prove the effectiveness of CMO with various experiments in the experimental section.

4. Experiments

We present various experiments and analyses of CMO in this section. We first describe our experimental settings with implementation details in Section 4.1. Next, we show the effectiveness of CMO on three long-tailed classification benchmarks: CIFAR-100-LT, ImageNet-LT, and iNaturalist, where CMO consistently boosts the performance of baselines with state-of-the-art level accuracy (Section 4.2). We also present in-depth analyses of CMO to study its inherent characteristics in Section 4.3.

4.1. Experimental Settings

Datasets. We validate CMO on the most commonly used long-tailed recognition benchmark datasets: CIFAR-100-LT [5], ImageNet-LT [28], and iNaturalist 2018 [18] (see Table 1). CIFAR-100-LT and ImageNet-LT are artificially made imbalanced from their balanced versions (CIFAR-100 [23] and ImageNet-2012 [35]). The iNaturalist 2018 dataset is a large-scale real-world dataset that exhibits long-tailed imbalance. We used the official training and test splits in our experiments.

Table 1. **Summary of datasets**. The imbalance ratio ρ is defined by $\rho = \max_k\{n_k\}/\min_k\{n_k\}$, where n_k is the number of samples in the k-th class.

Dataset	# of classes	# of training	Imbalance ratio
CIFAR-100-LT	100	50K	{10, 50, 100}
ImageNet-LT	1,000	115.8K	256
iNaturalist 2018	8,142	437.5K	500

Evaluation Metrics. The performances are mainly reported as the overall top-1 accuracy. Following [28], we also report the accuracy of three disjoint subsets: Many-shot classes (classes with more than 100 training samples), Medium-shot classes (classes with 20 to 100 samples), and Few-shot classes (classes under 20 samples).

Comparison methods. We compare CMO with the minority oversampling methods, the state-of-the-art long-tail recognition methods, and their combinations.

- **Minority oversampling.** (1) No oversampling (Vanilla);(2) Random oversampling (ROS) [41], that oversamples minority samples to balance the classes in training data; (3) Remix [7], which oversamples minority classes by assigning higher weights to the minority labels when using Mixup [48]; (4) Feature space augmentation (FSA) [8];

- **Re-weighting.** (5) Focal loss [25], which is an instance re-weighting method; (6) LDAM loss [5], which regularizes the minority classes to increase margins to decision boundary; (7) IB loss [32], which re-weights samples by their influences; (8) Balanced Softmax [34], an unbiased extension of Softmax; (9) LADE [17], which disentangles the source label distribution from the model prediction in training.

- **Other state-of-the-art methods.** (10) Deferred re-weighting (DRW) [5] and (11) Decouple [22] are two-stage algorithms that re-balance the classifiers during fine-tuning; (12) BBN [51] and (13) RIDE [43] use additional network branches to handle class imbalance; (14) Causal Norm [40], which disentangles causal effects and adjusts the effects in training; (15) MiSLAS [50], the two-stage algorithm, enhances classifier learning and calibration with label-aware smoothing (LAS) in Stage-2.

Implementation. We use PyTorch [33] for all experiments. For CIFAR datasets, we use ResNet-32 [15]. The networks are trained for 200 epochs following the training strategy

Table 2. **State-of-the-art comparison on CIFAR-100-LT dataset.** Results with classification accuracy (%) for ResNet-32 architecture on CIFAR-100-LT with different imbalance ratios. $*$, † and ‡ are from the original paper, [40] and [17], respectively. The best results are marked in bold.

Imbalance ratio	100	50	10
Cross Entropy (CE)	38.6	44.0	56.4
CE-DRW	41.1	45.6	57.9
LDAM-DRW [5]	41.7	47.9	57.3
BBN [51]†	42.6	47.1	59.2
Causal Norm [40]†	44.1	50.3	59.6
IB Loss [32]*	45.0	48.9	58.0
Balanced Softmax [34]‡	45.1	49.9	61.6
LADE [17]*	45.4	50.5	**61.7**
Remix [7]	45.8	49.5	59.2
RIDE [43]	48.6	51.4	59.8
CE + CMO	43.9	48.3	59.5
CE-DRW + CMO	47.0	50.9	**61.7**
LDAM-DRW + CMO	47.2	51.7	58.4
RIDE + CMO	**50.0**	**53.0**	60.2

in [5]. For ImageNet-LT, we use ResNet-50 as the backbone network. The network is trained for 100 epochs with an initial learning rate of 0.1. The learning rate is decayed at 60th and 80th epoch by 0.1. For iNaturalist 2018, we use ResNet-{50, 101, 152} and Wide ResNet-50 [47]. We train the networks for 200 epochs with an initial learning rate of 0.1, and decay the learning rate at epoch 75 and 160 by 0.1. All experiments are trained with stochastic gradient descent (SGD) with momentum 0.9.

4.2. Long-tailed classification benchmarks

4.2.1 CIFAR-100-LT

We conduct experiments on CIFAR-100-LT with different imbalance ratios: 10, 50, 100. We apply CMO to various methods to verify its effectiveness on different algorithms: vanilla cross-entropy loss, class-reweighting loss (LDAM [5]), two-stage algorithm (DRW [5]), and multi-branch architecture (RIDE [43]).

Comparison with state-of-the-art methods. The overall classification accuracies are provided in Table 2. It is a surprising result that CMO with basic cross-entropy (CE) loss shows comparable performance with complex long-tail recognition methods. Moreover, applying CMO to the state-of-the-art model (i.e., RIDE) further boosts the performance with a large gap, especially when the imbalance ratios are high as 50 and 100.

Comparison with oversampling methods. We further compare CMO with other oversampling techniques for performance improvement when combined with long-tailed recognition methods in Table 3. The results show that our method consistently significantly improves the performance

Table 3. **Comparison against baselines on CIFAR-100-LT.** Results with classification accuracy (%) of ResNet-32. The best results are marked in bold.

	Vanilla	+ROS [41]	+Remix [7]	+CMO
CE	38.6 (+0.0)	32.3 (-5.3)	40.0 (+1.4)	**43.9 (+5.3)**
CE-DRW [5]	41.1 (+0.0)	35.9 (-5.2)	45.8 (+4.7)	**47.0 (+5.9)**
LDAM-DRW [5]	41.7 (+0.0)	32.6 (-9.1)	45.3 (+3.6)	**47.2 (+5.5)**
RIDE [43]	48.6 (+0.0)	22.6 (-26.0)	44.0 (-4.6)	**53.0 (+4.4)**

Table 4. **State-of-the-art comparison on ImageNet-LT.** Results with classification accuracy (%) of ResNet-50 with state-of-the-art methods on ImageNet-LT. Baseline results and "$*$" are from the original papers. "†" and "‡" denote the results from [22] and [40], respectively. The best results are marked in bold.

	All	Many	Med	Few
Cross Entropy (CE)†	41.6	64.0	33.8	5.8
Focal Loss [25]‡	43.7	51.0	40.8	20.8
Decouple-$\tau-$norm [22]†	46.7	56.6	44.2	27.4
Decouple-cRT [22]†	47.3	58.8	44.0	26.1
Decouple-LWS [22]†	47.7	57.1	45.2	29.3
Remix [7]	48.6	60.4	46.9	30.7
LDAM-DRW [5]	49.8	60.4	46.9	30.7
CE-DRW	50.1	61.7	47.3	28.8
Balanced Softmax (BS) [34]	51.0	60.9	48.8	32.1
Causal Norm [40]‡	51.8	62.7	48.8	31.6
LADE [17]*	51.9	62.3	**49.3**	31.2
CE + CMO	49.1	**67.0**	42.3	20.5
CE-DRW + CMO	51.4	60.8	48.6	35.5
BS + CMO	**52.3**	62.0	49.1	**36.7**

of all long-tailed recognition methods. On the other hand, simply balancing the class distribution with ROS [41] leads to severe performance degradation. We speculate that this is because the naive balancing of the sampling distribution over classes hinders the model from learning generalized features for major classes and induces the model to memorize the minor class samples. Remix [7] improves the performance of some methods but degrades the performance when combined with a complex state-of-the-art method, RIDE [43]. This indicates that a simple labeling policy of Remix may not be effective when the model complexity gets large as in RIDE. Since Remix shows the best performance when combined with CE-DRW, from now on, we report the experimental results of Remix using this strategy unless it is specified.

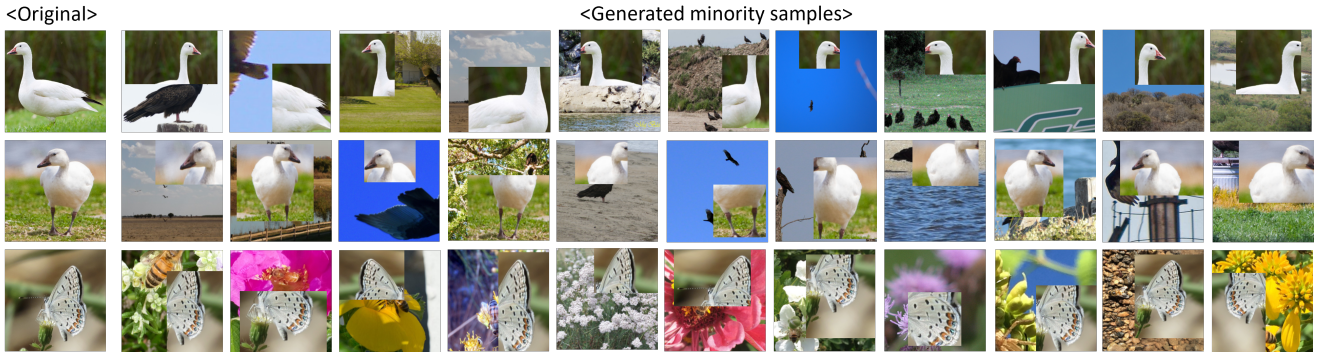

Figure 2. **Visualization of the minority images generated by CMO** (Minority class: Snow goose and Acmon blue (Butterfly)). We randomly choose generated images for each original image. Our method is able to generate context-rich minority samples with diverse contexts. For example, while the original 'Snow goose' class contains only 'Snow goose' images on the grass, the generated images have various contexts such as the sky, the sea, the sand, and a flock of crows. These generated images lead the model to learn a robust representation of minority classes.

Table 5. **Comparison against baselines on ImageNet-LT.** Results with classification accuracy (%) of ResNet-50. The best results are marked in bold.

	Vanilla	+Remix [7]	+CMO
CE	41.6 (+0.0)	41.7 (+0.1)	**49.7 (+8.1)**
CE-DRW [5]	50.1 (+0.0)	48.6 (-1.5)	**51.4 (+1.3)**
Balanced Softmax [34]	51.0 (+0.0)	49.2 (-1.8)	**52.3 (+1.3)**

4.2.2 ImageNet-LT

We assess the scalability of CMO on ImageNet-LT with a combination of various long-tailed recognition methods. We apply CMO on simple yet effective long-tailed recognition methods, CE-DRW [5] and Balanced Softmax (BS) [34].

Comparison with state-of-the-art methods. Results of our method and other long-tailed recognition methods are available in Table 4. Applying CMO to the basic training with CE loss improves the performance with a significant gap, outperforming most of the recent baselines. Greater performance improvement on ImageNet-LT compared to CIFAR-100 indicates that our method can benefit from as richer context information available in major classes of ImageNet-LT. In addition, consistent performance improvement of CMO when combined with DRW or BS bolsters the efficacy of CMO, as it can be easily integrated with modern state-of-the-art long-tailed recognition methods. It is noteworthy that as {CE-DRW + CMO } and {BS + CMO } especially achieve much higher few-shot class accuracy than other methods, our method is useful in achieving consistent performance over classes.

Table 6. **State-of-the-art comparison on iNaturalist2018.** Results with classification accuracy (%) of ResNet-50 on iNaturalist2018. "$*$" and "†" indicate the results from the original paper and [51], respectively. The best results are marked in bold.

	All	Many	Med	Few
Cross Entropy (CE)	61.0	73.9	63.5	55.5
IB Loss [32]*	65.4	-	-	-
FSA [8]*	65.9	-	-	-
LDAM-DRW [5]†	66.1	-	-	-
Decouple-cRT [22]*	68.2	73.2	68.8	66.1
Decouple-$\pi-$norm [22]*	69.3	71.1	68.9	69.3
Decouple-LWS [22]*	69.5	71.0	69.8	68.8
BBN [51]*	69.6	-	-	-
Balanced Softmax [34]	70.0	70.0	70.2	69.9
LADE [17]*	70.0	-	-	-
Remix [7]*	70.5	-	-	-
MiSLAS [50]*	71.6	-	-	-
CE + CMO	68.9	**76.9**	69.3	66.6
CE-DRW + CMO	70.9	68.2	70.2	72.2
BS + CMO	70.9	68.8	70.0	**72.3**
CE-DRW + CMO + LAS [50]	**71.8**	69.6	**72.1**	71.9

Comparison with oversampling methods. In Table 5, we compare performance improvement with other oversampling techniques. While CMO consistently improves performance for all methods, Remix [7] fails to improve performance of the long-tailed recognition methods and barely improves the model trained with cross-entropy loss. This implies that the labeling strategy of Remix is not enough to compensate for the adverse effect of naively using the same original distribution as two sampling distributions of Mixup, especially when the imbalance ratio gets as severe as 256 of ImageNet-LT. On the other hand, CMO generates more minority samples by using different distributions when selecting two images and shows much better performance in all tasks.

4.2.3 iNaturalist 2018

We further evaluate our proposed method on iNaturalist 2018, the real-world large-scale long-tailed dataset.

Comparison with state-of-the-art methods. Table 6 presents the classification results. On the naturally-skewed dataset, applying CMO to the simple training scheme of CE-DRW surpasses most of the state-of-the-arts. On iNaturalist 2018, as in ImageNet-LT, CMO dramatically improves the performance of cross-entropy loss (CE) by **7.9%p** ($61.0\% \rightarrow 68.9\%$). This is because the sample generation of CMO fully utilizes the abundant context of training data. Again, it can be seen that remarkable performance improvement is achieved in the few-shot classes. Lastly, applying label-aware smoothing (LAS) [50] to CE-DRW+CMO model achieves the new state-of-the-art performance. We apply the same stage-2 strategy from [50].

Results on large models. Since it is well-known that larger datasets can be fully utilized when the capacity of models is large enough, we investigate the performance of CMO and other oversampling methods with large deep networks of Wide ResNet-50 [47], ResNet-101, and ResNet-152 [15]. We compare CMO with the feature space augmentation method (FSA) [8]. While both methods improve the vanilla training with cross-entropy loss, our method shows superior performance than FSA. This indicates that augmenting samples by explicitly distinguishing the source of context and foreground information, and controlling the distribution of each source is much effective in improving the overall performance.

Table 7. **Results on large architectures.** Classification accuracy (%) of large backbone networks on iNaturalist 2018. The results are copied from the original paper.

Method	ResNet-50	Wide ResNet-50	ResNet-101	ResNet-152
CE	61.0	-	65.2	66.2
FSA [8]	65.9	-	68.4	69.1
CMO	**70.9**	**71.9**	**72.4**	**72.6**

Visualization of generated images. To verify the context-richness of minority samples generated by CMO, we visualize the generated images for the minority classes in Figure 2. From the rarest minority classes, we randomly choose generated images for each original image. We can observe that CMO produces diverse minority samples with various contexts. For example, while the original 'Snow goose' class only contains images of gooses on the grass, the generated images have various contexts, such as the sky or sea. Likewise, the butterfly in the third row is newly created as diversified images with various contexts, containing bees and flowers of various colors and shapes. We argue that various combinations of context and minority samples encourage the model to learn a robust representation of the minority classes.

4.3. Analysis

Is distribution for augmenting images important? To justify the need of different distributions for background and foreground images, we compare CutMix and CMO. As can be seen from Table 8, CMO outperforms CutMix with a large gap for long-tailed classification. In particular, it shows remarkable performance improvement in Med and Few-shot classes. The performance gap is due to the absence of minor-class-weighted distribution in naive CutMix. Although naive CutMix can generate informative mixed samples, it has a limited effect to cope with the long-tailed distribution. Thus, we claim that the use of minor-class-weighted distribution is a key-point in data augmentation for long-tailed setting, which enhances the contribution and originality of CMO.

Table 8. **Comparison against CutMix.** We use cross-entropy loss for all experiments.

	All	Many	Med	Few
CIFAR-100-LT				
CutMix	35.6	71.0	37.9	4.9
CMO	43.9	70.4	42.5	14.4
ImageNet-LT				
CutMix	45.5	68.6	38.1	8.1
CMO	49.5	68.3	42.7	21.6

How to choose appropriate probability distribution Q? We evaluate different sampling strategies in Section 3.2 on CIFAR-100 with the imbalance ratio 100 and the results are reported in Table 9. Although the sensitivity for distribution is not significant, $q(1, k)$ shows the most balanced performance.

Table 9. **Impact of different sampling distribution Q.** Results on CIFAR-100-LT (imbalance ratio=100) according to different sampling probabilities Q.

	All	Many	Med	Few
$q(1/2, k)$	42.6	**71.6**	42.1	9.5
$q(1, k)$	**43.9**	70.4	**42.5**	**14.4**
$q(2, k)$	40.1	67.2	36.7	12.3
$E(k)$ [9]	39.5	70.4	38.0	4.7

This result is consistent with the common practice of balancing the dataset by giving weight in reciprocal to frequency. While $q(2, k)$, which imposes a higher probability on the minority class than $q(1, k)$, shows decent performance in few-shot classes, the overall performance

slightly deteriorates. We assume this is because we cannot sample more diverse images when imposing too high probabilities to the few-shot classes. Based on this result, we set Q as $q(1, k)$ in our all experiments.

Why should we oversample only for the foreground samples? Although we have discussed the desired minority oversampling in Figure 1, one may still wonder why applying the oversampling only for the foreground samples is better than oversampling both patches and background samples or oversampling only the backgrounds. To verify our design choice, we evaluate two variants of CMO. The first variant, CMO $_{back}$, samples background images from minor-class-weighted distribution and patches from original distribution, which is exactly the opposite of CMO, i.e., $(x^b, y^b) \sim Q, (x^f, y^f) \sim P$. The second variant, CMO $_{minor}$, samples both background and patches from minor-class-weighted distribution, i.e., $(x^b, y^b), (x^f, y^f) \sim Q$. We report the results of applying variants to the model training with CE loss and LDAM loss [5] in Table 10.

Table 10. **Ablation study.** Results on variants of CMO with ResNet-32 on imbalanced CIFAR-100, imbalance ratio of 100.

	All	Many	Med	Few
Cross Entropy (CE)	38.6	65.3	37.6	8.7
CE + CMO $_{minor}$	37.9	58.3	40.4	11.2
CE + CMO $_{back}$	40.1	64.7	40.2	11.3
CE + CMO	**43.9**	**70.4**	**42.5**	**14.4**
LDAM [5]	41.7	61.4	42.2	18.0
LDAM + CMO $_{minor}$	31.7	50.2	33.2	8.4
LDAM + CMO $_{back}$	44.2	59.2	46.6	24.0
LDAM + CMO	**47.2**	**61.5**	**48.6**	**28.8**

We first observe that CMO $_{minor}$ shows severe performance degradation in both methods. We suspect that this is because the rich context of the majority samples cannot be utilized. In contrast, CMO $_{back}$ shows decent performance improvements, but far less than the original CMO. This is because, in the CutMix strategy, the background image has a high probability that the object is overlapped by the foreground image. Therefore, we can expect the loss of information about minority classes in the background image, resulting in a limited performance boost.

Comparison with other minority augmentations. We further analyze the effectiveness of CutMix compared to the other augmentation strategy, such as Mixup [48], color jitter, and Gaussian blur. For Mixup, we use the same sampling strategy with CMO, and for color jitter and Gaussian blur, which do not interpolate two images, we apply augmentation only to the minority classes and oversample those classes. As shown in Table 11, other augmentation methods provide limited performance gain compared to the CutMix. We suspect that it is because the pixel-level transformations are not effective for producing minority samples with rich context. Gaussian blur and color jitter do not combine two images, thus it is hard to add a new context to minority samples. Although Mixup combines two images, it does not distinguish the role of two samples, limiting the control of the source of context and patch information. On the other hand, CutMix can create diverse images with larger changes at pixel-level compared to other methods.

Table 11. **Data augmentation strategies.** Comparison against different augmentation strategies for generating new minority samples on imbalanced CIFAR-100 with imbalance ratio of 100.

	All	Many	Med	Few
CMO w/ Gaussian Blur	31.1	54.7	28.8	6.2
CMO w/ Color Jitter	34.7	58.9	34.4	6.8
CMO w/ Mixup	38.0	54.8	40.2	**15.9**
CMO w/ CutMix	**43.9**	**70.4**	**42.5**	14.4

5. Conclusion

We have proposed a novel context-rich oversampling method, CMO, to solve data imbalance problem. We tackle the fundamental problem of previous oversampling methods that generate context-limited minority samples, which rather intensifies the overfitting problem. Our key idea is to transfer the rich contexts of majority samples to minority samples to augment context-rich minority samples. Implementation of CMO is simple and intuitive. Extensive experiments on various benchmark datasets not only show that our CMO brings a significant performance improvement, but also that adding our oversampling method to the basic losses renews the state-of-the-art.

Limitations. Not all but in some cases, the performance improvement for the minority classes occurs by sacrificing the performance of the majority classes. For example, in Table 4, comparing the case of using only CE-DRW and applying our method to CE-DRW, the performance increases by 6.7%p in the few-shot classes, but decreases by 0.9%p in the many-shot classes. Future works should therefore include follow-up works designed to improve the performance of all classes without sacrificing the performance of the many-shot classes.

Potential Negative Societal Impact. Since our method creates new samples, it can benefit more from longer training and deeper architectures. Thus, our method may lead to consuming more resources, which has a risk that the use of GPUs for machine learning could accelerate environmental degradation [16]. Nevertheless, we would like to emphasize that our method helps achieve better performance with the same training iteration and backbone network, without long training.

References

[1] Shin Ando and Chun Yuan Huang. Deep over-sampling framework for classifying imbalanced data. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2017.

[2] David Bau, Jun-Yan Zhu, Jonas Wulff, William Peebles, Hendrik Strobelt, Bolei Zhou, and Antonio Torralba. Seeing what a gan cannot generate. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[3] Mateusz Buda, Atsuto Maki, and Maciej A. Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106, 2018.

[4] Chumphol Bunkhumpornpat, Krung Sinapiromsaran, and Chidchanok Lursinsap. Safe-level-smote: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem. In Thanaruk Theeramunkong, Boonserm Kijsirikul, Nick Cercone, and Tu-Bao Ho, editors, *Advances in Knowledge Discovery and Data Mining*, 2009.

[5] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *Advances in Neural Information Processing Systems*, 2019.

[6] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote: Synthetic minority over-sampling technique. *J. Artif. Int. Res.*, 2002.

[7] Hsin-Ping Chou, Shih-Chieh Chang, Jia-Yu Pan, Wei Wei, and Da-Cheng Juan. Remix: Rebalanced mixup. In *Computer Vision – ECCV 2020 Workshops*, 2020.

[8] Peng Chu, Xiao Bian, Shaopeng Liu, and Haibin Ling. Feature space augmentation for long-tailed data. In *Computer Vision – ECCV 2020*, 2020.

[9] Y. Cui, M. Jia, T. Lin, Y. Song, and S. Belongie. Class-balanced loss based on effective number of samples. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[10] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

[11] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2014.

[13] 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*, 2017.

[14] Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. Borderline-smote: A new over-sampling method in imbalanced data sets learning. In De-Shuang Huang, Xiao-Ping Zhang, and Guang-Bin Huang, editors, *Advances in Intelligent Computing*, 2005.

[15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[16] C Herweijer and D Waughray. Harnessing artificial intelligence for the earth. *Fourth Industrial Revolution for the Earth Series*, 2018.

[17] Youngkyu Hong, Seungju Han, Kwanghee Choi, Seokjun Seo, Beomsu Kim, and Buru Chang. Disentangling label distribution for long-tailed visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6626–6636, June 2021.

[18] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alexander Shepard, Hartwig Adam, Pietro Perona, and Serge J. Belongie. The inaturalist species classification and detection dataset. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 2018.

[19] C. Huang, Y. Li, C. C. Loy, and X. Tang. Learning deep representation for imbalanced classification. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[20] Justin Johnson and Taghi Khoshgoftaar. Survey on deep learning with class imbalance. *Journal of Big Data*, 6:27, 2019.

[21] H. Kahn and A. W. Marshall. Methods of Reducing Sample Size in Monte Carlo Computations. *Operations Research*, 1(5):263–278, 1953.

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

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

[24] Shuang Li, Kaixiong Gong, Chi Harold Liu, Yulin Wang, Feng Qiao, and Xinjing Cheng. Metasaug: Meta semantic augmentation for long-tailed visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5212–5221, June 2021.

[25] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017.

[26] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollar, and Larry Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[27] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[28] 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)*, 2019.

[29] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. In *Computer Vision – ECCV 2018*, 2018.

[30] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, volume 26, 2013.

[31] Sankha Subhra Mullick, Shounak Datta, and Swagatam Das. Generative adversarial minority oversampling. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, 2019.

[32] Seulki Park, Jongin Lim, Younghan Jeon, and Jin Young Choi. Influence-balanced loss for imbalanced visual classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 735–744, October 2021.

[33] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017.

[34] Jiawei Ren, Cunjun Yu, Shunan Sheng, Xiao Ma, Haiyu Zhao, Shuai Yi, and Hongsheng Li. Balanced meta-softmax for long-tailed visual recognition. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[35] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 2015.

[36] Nikolaos Sarafianos, Xiang Xu, and Ioannis A Kakadiaris. Deep imbalanced attribute classification using visual attention aggregation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[37] Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. Meta-weight-net: Learning an explicit mapping for sample weighting. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 2019.

[38] Cecilia Summers and Michael J Dinneen. Improved mixed-example data augmentation. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2019.

[39] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[40] Kaihua Tang, Jianqiang Huang, and Hanwang Zhang. Long-tailed classification by keeping the good and removing the bad momentum causal effect. In *NeurIPS*, 2020.

[41] Jason Van Hulse, Taghi M. Khoshgoftaar, and Amri Napolitano. Experimental perspectives on learning from imbalanced data. In *Proceedings of the 24th International Conference on Machine Learning*, 2007.

[42] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *International Conference on Machine Learning*. PMLR, 2019.

[43] Xudong Wang, Long Lian, Zhongqi Miao, Ziwei Liu, and Stella Yu. Long-tailed recognition by routing diverse distribution-aware experts. In *International Conference on Learning Representations*, 2021.

[44] Yulin Wang, Xuran Pan, Shiji Song, Hong Zhang, Gao Huang, and Cheng Wu. Implicit semantic data augmentation for deep networks. In *NeurIPS*, 2019.

[45] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Learning to model the tail. In *Advances in Neural Information Processing Systems*, 2017.

[46] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *International Conference on Computer Vision (ICCV)*, 2019.

[47] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2016.

[48] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

[49] J. Zhang and I. Mani. KNN Approach to Unbalanced Data Distributions: A Case Study Involving Information Extraction. In *Proceedings of the ICML'2003 Workshop on Learning from Imbalanced Datasets*, 2003.

[50] Zhisheng Zhong, Jiequan Cui, Shu Liu, and Jiaya Jia. Improving calibration for long-tailed recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[51] Boyan Zhou, Quan Cui, Xiu-Shen Wei, and Zhao-Min Chen. BBN: bilateral-branch network with cumulative learning for long-tailed visual recognition. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, 2020.

Supplementary Material

A. Implementation details

In this section, we describe more implementation details which are not included in Section 4.1.

CIFAR-100-LT. For fair comparison, we use the same random seed to make CIFAR-100-LT and follow the implementation of [5]. We train ResNet-32 [15] by SGD optimizer with a momentum of 0.9 and weight decay of 2×10^{-4}. As in [5], we use simple data augmentation [15] by padding 4 pixels on each side and apply random cropping or horizontal flipping to 32×32 size. We train for 200 epochs and use a linear warm-up of the learning rate [13] in the first five epochs. The learning rate is initialized as 0.1, and it is decayed at the 160th and 180th epoch by 0.01. The model is trained with a batch size of 128 on single GTX 1080Ti. We turn off applying CMO for the last three epochs so that the model can be finetuned in the original input space.
For experiments in Table 11, we use the same strategy for {CMO w/ Mixup}. For {CMO w/ Gaussian Blur} and {CMO w/ Color Jitter} which do not mix two images, we divide classes into two group: majority and minority. Then, for the minority group, we augment the data with color jitter and gaussian blur, respectively. We set brightness to 0.5 and hue to 0.3 for color jitter, and set kernel size as $(5, 7)$ and sigma as $(0.1, 5)$ for gaussian blur using PyTorch [33] implemented functions.

ImageNet-LT. For ImageNet-LT, we follow most of the details from [43]. As in [43], we perform simple horizontal flips, color jittering, and taking random crops of size 224×224. We use ResNet-50 as a backbone network. The networks are trained with a batch size of 256 on 4 GTX 1080Ti GPUs for 100 epochs using SGD with an initial learning rate of 0.1 decayed by 0.1 at 60 epochs and 80 epochs.

iNaturalist 2018. For iNatualist 2018, we use the same data augmentation strategy as in ImageNet-LT. Multiple backbone networks are experimented on iNaturalist 2018, including ResNet-50, ResNet-101, ResNet-152 [15], and Wide ResNet-50 [47]. All backbone networks are trained with a batch size of 512 on 8 Tesla V100 GPUs for 200 epochs using SGD with an initial learning rate of 0.1 decayed by 0.1 at 75 epochs and 160 epochs.

B. Comparison with oversampling methods.

We compare CMO with other oversampling methods for performance improvement on CIFAR-100 with imbalance ratio 50 and 10 in Table 12. As in the imbalance ratio of 100, our method consistently improves performance in all long-tailed recognition methods.

Table 12. **Comparison against baselines on CIFAR-100-LT.** Results with classification accuracy (%) of ResNet-32. The best results are marked in bold.

Imbalance ratio	50				10			
Method	Vanilla	+ROS [41]	+Remix [7]	+CMO	Vanilla	+ROS [41]	+Remix [7]	+CMO
CE	44.0 (+0.0)	39.7 (-4.3)	45.0 (+1.0)	**48.3 (+4.3)**	56.4 (+0.0)	55.6 (-0.8)	58.7 (+2.3)	**59.5 (+3.1)**
CE-DRW [5]	45.6 (+0.0)	41.3 (-4.3)	49.5 (+3.9)	**50.9 (+5.3)**	57.9 (+0.0)	56.4 (-1.5)	59.2 (+1.3)	**61.7 (+3.8)**
LDAM-DRW [5]	47.9 (+0.0)	38.3 (-9.6)	48.8 (+0.9)	**51.7 (+3.8)**	57.3 (+0.0)	53.9 (-3.4)	55.9 (-1.4)	**58.4 (+1.1)**
RIDE [43]	51.4 (+0.0)	31.3 (-20.1)	47.9 (-3.5)	**53.0 (+1.6)**	59.8 (+0.0)	49.4 (-10.4)	59.5 (-0.3)	**60.2 (+0.4)**

C. Pseudo-code of Context-rich Minority Oversampling

We present the PyTorch-syle pseudo-code of CMO algorithm in Algorithm 2. Note that CMO is easy to implement with just a few lines that are easily applicable to any loss, networks, or algorithms. Thus, CMO can be a very practical and effective solution for handling imbalanced dataset.

Algorithm 2 PyTorch-style pseudo-code for CMO

```
# original_loader: data loader from original data distribution
# weighted_loader: data loader from minor-class-weighted distribution
# model: any backbone network such as ResNet or multi-branch networks (RIDE)
# loss: any loss such as CE, LDAM, balanced softmax, RIDE loss

for epoch in Epochs:
   # load a batch for background images from original data dist.
   for x_b, y_b in original_loader:
       # load a batch for foreground from minor-class-weighted dist.
       x_f, y_f = next(weighted_loader)

       # get coordinate for random binary mask
       lambda = np.random.uniform(0,1)
       cx = np.random.randint(W) # W: width of images
       cy = np.random.randint(H) # H: height of images
       bbx1 = np.clip(cx - int(W * np.sqrt(1. - lambda))//2,0,W)
       bbx2 = np.clip(cx + int(W * np.sqrt(1. - lambda))//2,0,W)
       bby1 = np.clip(cy - int(H * np.sqrt(1. - lambda))//2,0,H)
       bby2 = np.clip(cy + int(H * np.sqrt(1. - lambda))//2,0,H)

       # get minor-oversampled images
       x_b[:, :, bbx1:bbx2, bby1:bby2] = x_f[:, :, bbx1:bbx2, bby1:bby2]
       lambda = 1 - ((bbx2 - bbx1) * (bby2 - bby1) / (W * H))# adjust lambda

       # output (x_f is attached to x_b)
       output = model(x_b)

       # loss
       losses = loss(output, y_b) * lambda + loss(output, y_f) * (1. - lambda)

       # optimization step
       losses.backward()
       optimizer.step()
```