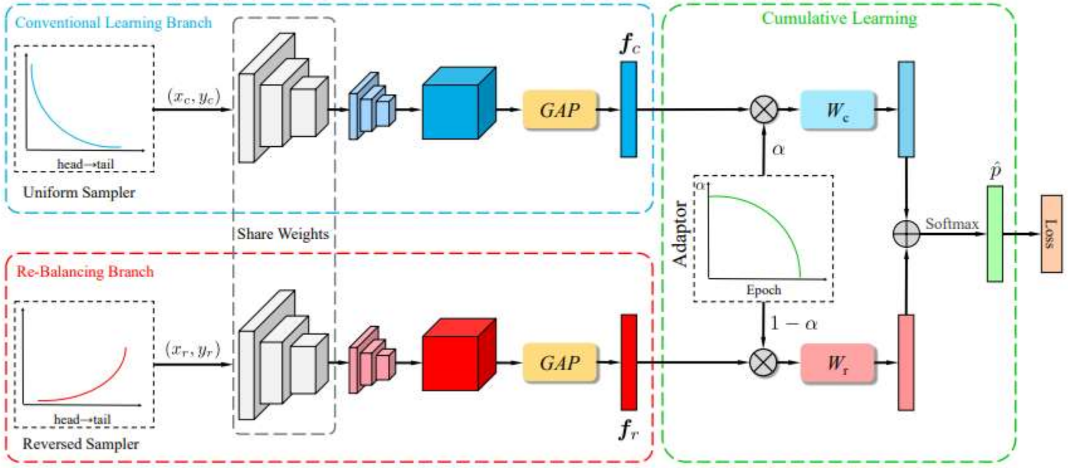
How Re-sampling Helps for Long-Tail Learning?

Jiang-Xin Shi[1 *] **Tong Wei**[2 *] **Yuke Xiang**[3] **Yu-Feng Li**[1 †]
[1]National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
[2]School of Computer Science and Engineering, Southeast University, Nanjing, China
[3]Consumer BG, Huawei Technologies, Shenzhen, China
{shijx,liyf}@lamda.nju.edu.cn, weit@seu.edu.cn, yuke.xiang@huawei.com

NeurIPS2023

Introduce

BBN[1]



probability of sampling

$$p_k = \frac{n_k^q}{\sum_{k' \in [K]} n_{k'}^q}$$

q=1: uniform sampling

q=0: class-balanced re-sampling

cRT[2]

- First, trains a preliminary model using the uniform sampler
- Second, fixes the representations and re-trains the linear classifier using a class-balanced sampler

[1] Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition

[2] Decoupling representation and classifier for long-tailed recognition

Introduce

Table 1: Test accuracy (%) of CE with uniform sampling, classifier re-training (cRT), and class-balanced re-sampling (CB-RS) on four long-tail benchmarks. We report the accuracy in terms of all, many-shot, medium-shot, and few-shot classes.

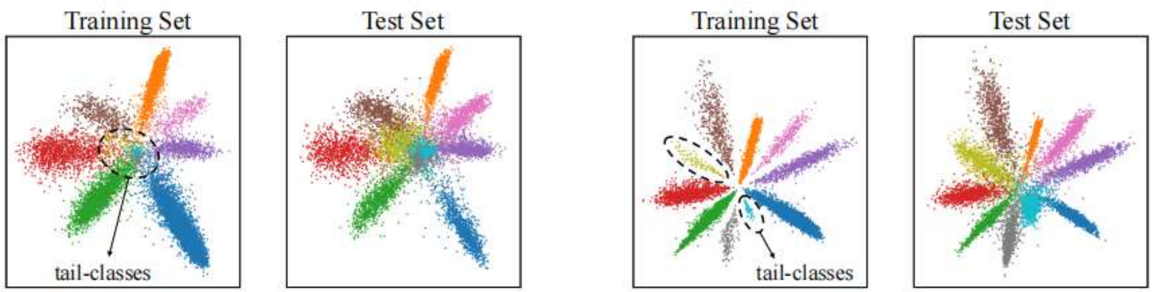
	MNIST-LT				Fashion-LT				CIFAR100-LT				ImageNet-LT			
	All	Many	Med.	Few	All	Many	Med.	Few	All	Many	Med.	Few	All	Many	Med.	Few
CE	65.8	**99.1**	89.9	0.0	45.6	**94.7**	43.1	0.0	39.1	**65.8**	36.8	8.8	35.0	**57.7**	26.5	4.7
cRT	82.5	96.6	89.4	58.8	60.3	77.1	61.4	42.1	**41.6**	63.0	**40.4**	**16.5**	**41.9**	52.9	**39.2**	**23.6**
CB-RS	**90.8**	98.7	**94.4**	**77.7**	**80.5**	86.6	**74.3**	**82.8**	34.1	59.5	31.1	6.2	37.6	47.5	36.5	16.7

highly semantically related to labels

complex contexts

cRT: uses uniform sampling to learn the representation and class-balanced resampling to fine-tune the classifier.

CB-RS: class-balanced re-sampling for the whole training process.



(a) Uniform sampling.

(b) Class-balanced re-sampling.

Figure 2: Visualization of learned representation of training and test set on MNIST-LT. Using class-balanced re-sampling yields more discriminative and balanced representations.

re-sampling is sensitive to the contexts in training samples

Introduce

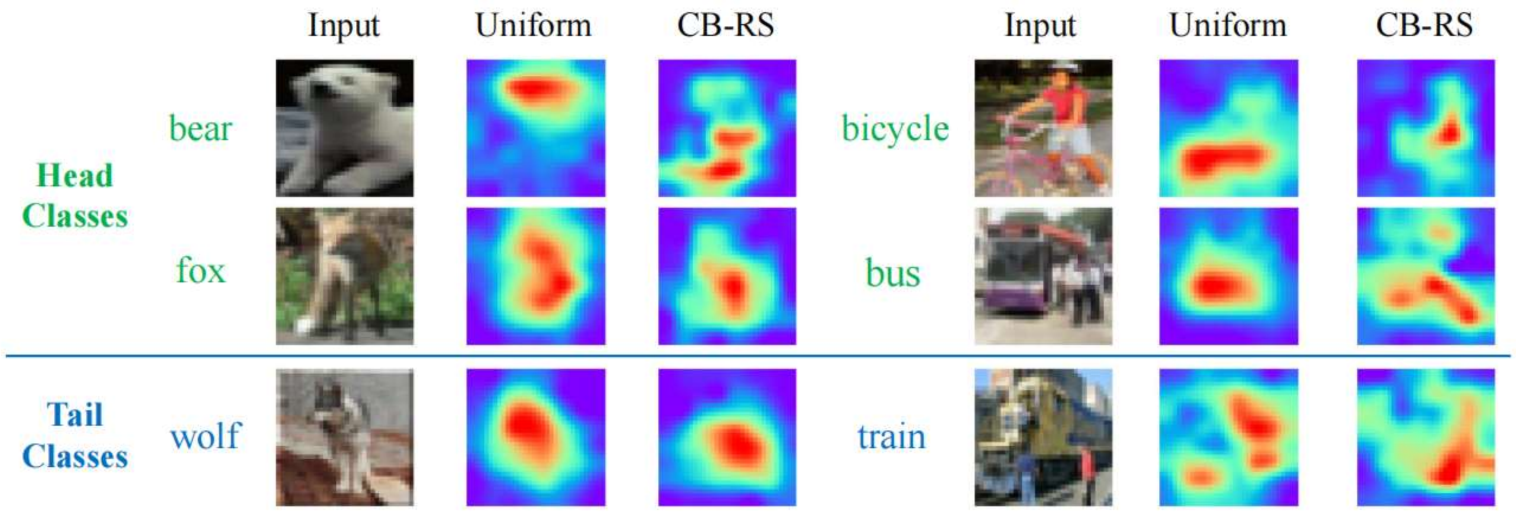
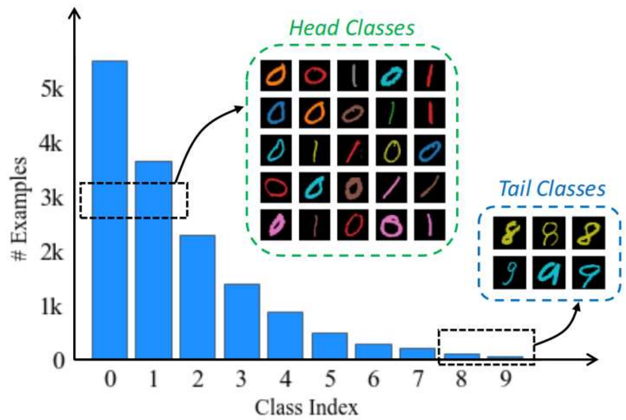


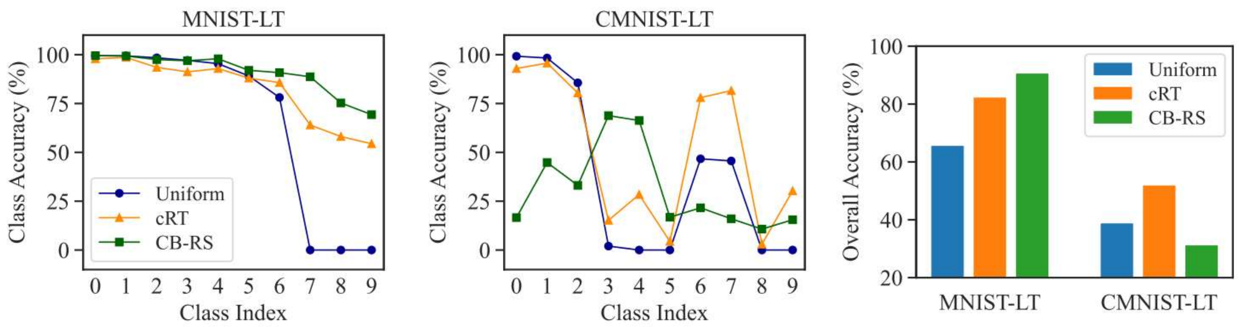
Figure 3: Visualization of features with Grad-CAM [17] on CIFAR100-LT. Uniform sampling mainly learns label-relevant features, while re-sampling overfits the label-irrelevant features.

Introduce

Colored MNIST-LT (CMNIST-LT)

- First, head classes are prone to have rich contexts, so we inject different colors into the samples of each head class
- Second, tail classes have limited contexts, so we inject an identical color into the samples of each tail class





(a) Comparison of class accuracy. (b) Overall accuracy.

Figure 4: Comparison of Uniform sampling, cRT, and CB-RS on MNIST-LT and CMNIST-LT.

re-sampling does not always fail, it can help for long-tail learning if avoiding the irrelevant contexts.

Method

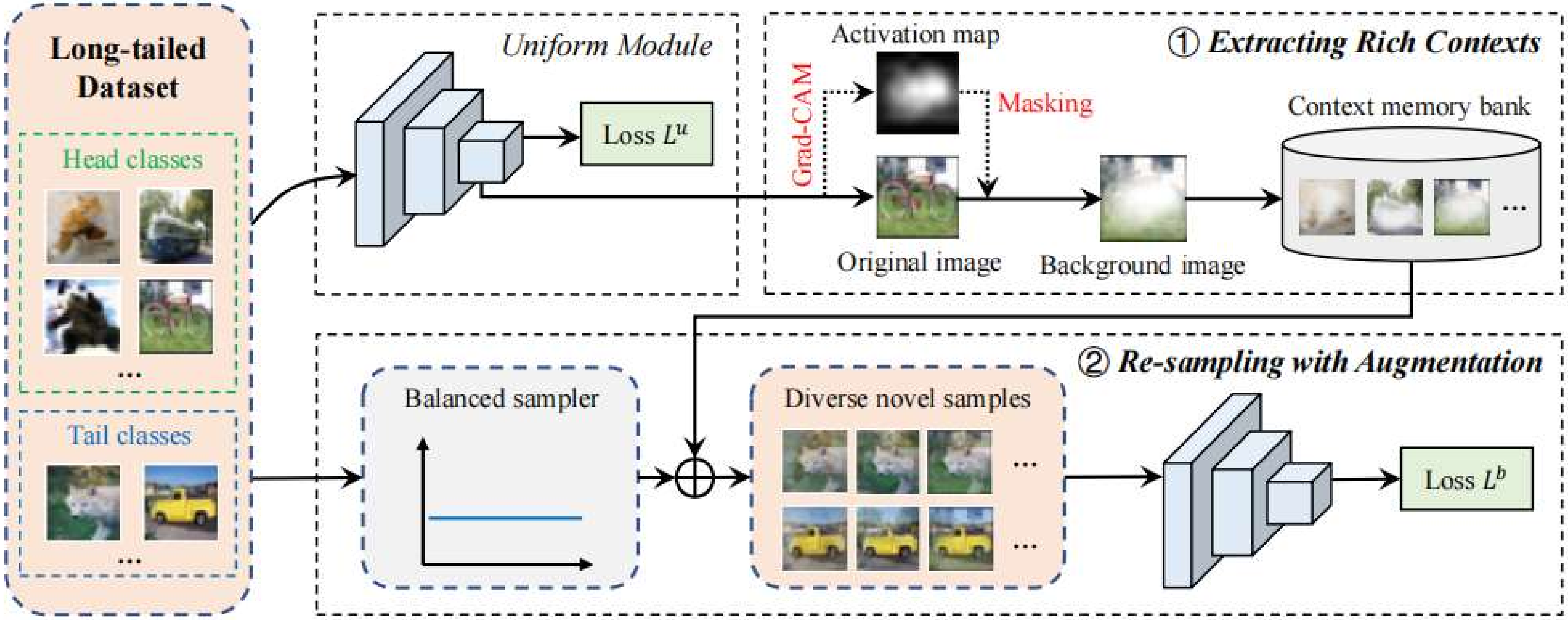
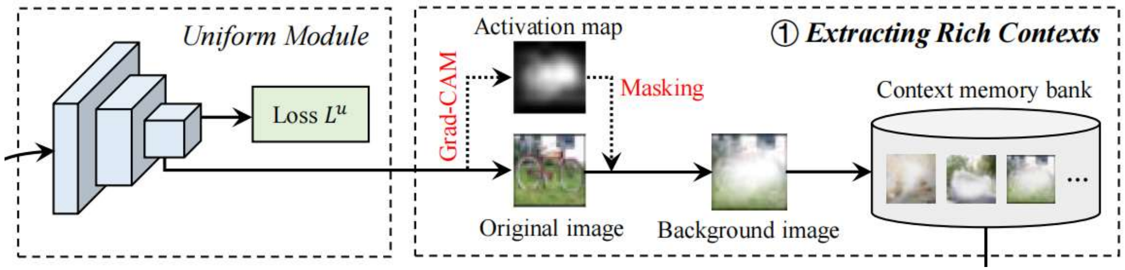
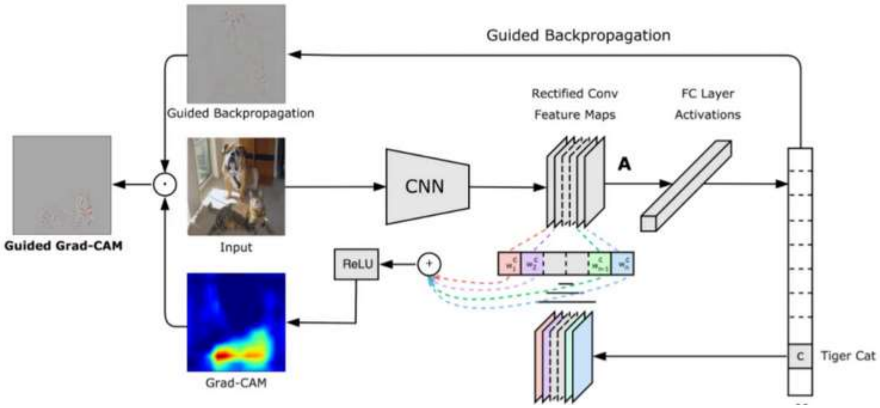


Figure 5: An overview of the proposed method.

Uniform Module



class activation map $\mathrm{CAM}(\boldsymbol{x}_i \mid f^u)$



Grad-CAM

background mask $\boldsymbol{M}_i = 1 - \mathrm{CAM}(\boldsymbol{x}_i \mid f^u)$

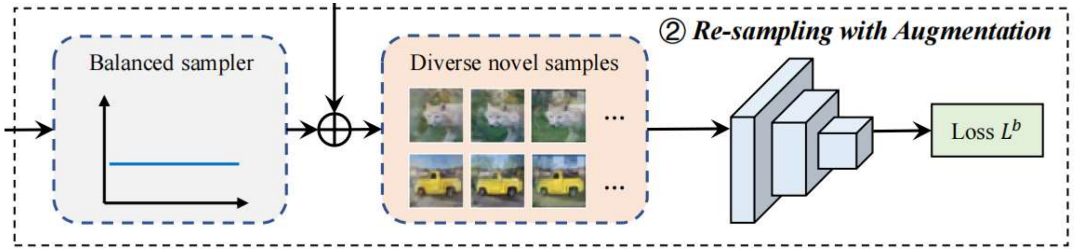
$$z_i^u = f^u(\boldsymbol{x}_i)$$
$$\mathcal{L}_i^u = \ell^u(z_i^u, y_i)$$

Algorithm 1 Training procedure of context-shift augmentation

Input: training data $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$; context memory bank Q, maximum volume size V; model parameters ϕ, f^u, f^b; loss functions ℓ^u, ℓ^b;
Procedure:
Initialize model parameters ϕ, f^u, f^b;
Re-sampling a class-balanced dataset $\widetilde{\mathcal{D}} = \{(\tilde{\boldsymbol{x}}_i, \tilde{y}_i)\}_{i=1}^N$;
Empty memory bank Q;
for epoch $= 1, \ldots, T$ **do**
  repeat
    Draw a mini-batch $(\boldsymbol{x}_i, y_i)_{i=1}^B$ from $\mathcal{D}$;
    Draw a mini-batch $(\tilde{\boldsymbol{x}}_i, \tilde{y}_i)_{i=1}^B$ from $\widetilde{\mathcal{D}}$;
    // uniform module
    for $i = 1, \ldots, B$ **do**
      Calculate $\boldsymbol{z}_i^u = f^u(\phi(\boldsymbol{x}_i))$ and $\mathcal{L}_i^u = \ell^u(\boldsymbol{z}_i^u, y_i)$;
      if $p(y = y_i \mid \boldsymbol{x}_i, \phi, f^u) \geq \delta$ **then**
        Calculate background mask $\boldsymbol{M}_i$ of $\boldsymbol{x}_i$;
        Push $(\boldsymbol{x}_i, \boldsymbol{M}_i)$ into Q;
      end if
    end for
    Calculate $\mathcal{L}^u = \frac{1}{B}\sum_{i=1}^B \mathcal{L}_i^u$;
    // balanced re-sampling module
    if Size of Q reaches V **then**
      Obtain contexts $(\check{\boldsymbol{x}}_i, \boldsymbol{M}_i)_{i=1}^B$ from Q;
      $\lambda \sim \mathrm{Uniform}(0, 1)$;
      for $i = 1, \ldots, B$ **do**
        $\tilde{\boldsymbol{x}}_i = \lambda \boldsymbol{M}_i \odot \check{\boldsymbol{x}}_i + (1 - \lambda \boldsymbol{M}_i) \odot \tilde{\boldsymbol{x}}_i$;
        Calculate $\boldsymbol{z}_i^b = f^b(\phi(\tilde{\boldsymbol{x}}_i))$ and $\mathcal{L}_i^b = \ell^b(\boldsymbol{z}_i^b, \tilde{y}_i)$;
      end for
      Calculate $\mathcal{L}^b = \frac{1}{B}\sum_{i=1}^B \mathcal{L}_i^b$;
    else
      Assign $\mathcal{L}^b = 0$;
    end if
    // total objective function
    Calculate $\mathcal{L} = \mathcal{L}^u + \mathcal{L}^b$;
    Update model parameters ϕ, f^u, f^b with $\mathcal{L}$;
  until all training data are traversed.
end for

Balanced Module

context memory bank Q



loss

$$\lambda \sim \text{Uniform}(0,1)$$

$$\tilde{\boldsymbol{x}}_i = \lambda \boldsymbol{M}_i \odot \check{\tilde{\boldsymbol{x}}}_i + (1-\lambda \boldsymbol{M}_i) \odot \tilde{\boldsymbol{x}}_i$$

$$\boldsymbol{z}_i^b = f^b(\tilde{\boldsymbol{x}}_i)$$

$$\mathcal{L}_i^b = \ell^b(\boldsymbol{z}_i^b, \tilde{y}_i)$$

overall loss

$$\mathcal{L} = \mathcal{L}^u + \mathcal{L}^b = \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}_i^u + \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}_i^b$$

Algorithm 1 Training procedure of context-shift augmentation

Input: training data $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$; context memory bank Q, maximum volume size V; model parameters ϕ, f^u, f^b; loss functions ℓ^u, ℓ^b;

Procedure:

Initialize model parameters ϕ, f^u, f^b;
Re-sampling a class-balanced dataset $\widetilde{\mathcal{D}} = \{(\tilde{\boldsymbol{x}}_i, \tilde{y}_i)\}_{i=1}^N$;
Empty memory bank Q;
for epoch $= 1, \dots, T$ **do**
  repeat
    Draw a mini-batch $(\boldsymbol{x}_i, y_i)_{i=1}^B$ from $\mathcal{D}$;
    Draw a mini-batch $(\tilde{\boldsymbol{x}}_i, \tilde{y}_i)_{i=1}^B$ from $\widetilde{\mathcal{D}}$;
    // uniform module
    for $i = 1, \dots, B$ **do**
      Calculate $\boldsymbol{z}_i^u = f^u(\phi(\boldsymbol{x}_i))$ and $\mathcal{L}_i^u = \ell^u(\boldsymbol{z}_i^u, y_i)$;
      if $p(y = y_i \mid \boldsymbol{x}_i, \phi, f^u) \geq \delta$ **then**
        Calculate background mask $\boldsymbol{M}_i$ of $\boldsymbol{x}_i$;
        Push $(\boldsymbol{x}_i, \boldsymbol{M}_i)$ into Q;
      end if
    end for
    Calculate $\mathcal{L}^u = \frac{1}{B}\sum_{i=1}^B \mathcal{L}_i^u$;
    // balanced re-sampling module
    if Size of Q reaches V **then**
      Obtain contexts $(\check{\tilde{\boldsymbol{x}}}_i, \boldsymbol{M}_i)_{i=1}^B$ from Q;
      $\lambda \sim \text{Uniform}(0, 1)$;
      for $i = 1, \dots, B$ **do**
        $\tilde{\boldsymbol{x}}_i = \lambda \boldsymbol{M}_i \odot \check{\tilde{\boldsymbol{x}}}_i + (1 - \lambda \boldsymbol{M}_i) \odot \tilde{\boldsymbol{x}}_i$;
        Calculate $\boldsymbol{z}_i^b = f^b(\phi(\tilde{\boldsymbol{x}}_i))$ and $\mathcal{L}_i^b = \ell^b(\boldsymbol{z}_i^b, \tilde{y}_i)$;
      end for
      Calculate $\mathcal{L}^b = \frac{1}{B}\sum_{i=1}^B \mathcal{L}_i^b$;
    else
      Assign $\mathcal{L}^b = 0$;
    end if
    // total objective function
    Calculate $\mathcal{L} = \mathcal{L}^u + \mathcal{L}^b$;
    Update model parameters ϕ, f^u, f^b with $\mathcal{L}$;
  until all training data are traversed.
end for

Experiments

Table 2: Test accuracy (%) on CIFAR datasets with various imbalanced ratios.

Dataset	CIFAR100-LT			CIFAR10-LT		
Imbalance Ratio	100	50	10	100	50	10
CE	38.3	43.9	55.7	70.4	74.8	86.4
Focal Loss [31]	38.4	44.3	55.8	70.4	76.7	86.7
CB-Focal [7]	39.6	45.2	58.0	74.6	79.3	87.1
CE-DRS [15]	41.6	45.5	58.1	75.6	79.8	87.4
CE-DRW [15]	41.5	45.3	58.1	76.3	80.0	87.6
LDAM-DRW [15]	42.0	46.6	58.7	77.0	81.0	88.2
cRT [6]	42.3	46.8	58.1	75.7	80.4	88.3
LWS [6]	42.3	46.4	58.1	73.0	78.5	87.7
BBN [14]	42.6	47.0	59.1	79.8	82.2	88.3
mixup [29]	39.5	45.0	58.0	73.1	77.8	87.1
Remix [33]	41.9	-	59.4	75.4	-	88.2
M2m [32]	43.5	-	57.6	79.1	-	87.5
CAM-BS [13]	41.7	46.0	-	75.4	81.4	-
CMO [27]	43.9	48.3	59.5	-	-	-
cRT+mixup [34]	45.1	50.9	62.1	79.1	84.2	89.8
LWS+mixup [34]	44.2	50.7	62.3	76.3	82.6	89.6
CSA (ours)	45.8	49.6	61.3	80.6	84.3	89.8
CSA + mixup (ours)	**46.6**	**51.9**	**62.6**	**82.5**	**86.0**	**90.8**

Table 3: Test accuracy (%) on ImageNet-LT dataset.

	ResNet-10	ResNet-50			
	(All)	All	Many	Med.	Few
CE	34.8	41.6	64.0	33.8	5.8
Focal Loss [31]	30.5	-	-	-	-
OLTR [5]	35.6	-	-	-	-
FSA [28]	35.2	-	-	-	-
cRT [6]	41.8	47.3	58.8	44.0	26.1
LWS [6]	41.4	47.7	57.1	45.2	29.3
BBN [14]	-	48.3	-	-	-
CMO [27]†	-	49.1	67.0	42.3	20.5
CSA (ours)	42.7	49.1	62.5	46.6	24.1
CSA† (ours)	**43.2**	**49.7**	63.6	47.0	23.8

† denotes a longer training of 100 epochs.

Experiments

Table 4: Ablation study on the context bank Q.

	All	Many	Med.	Few
Ours	45.8	64.3	49.7	18.2
Ours w/o Q	41.2 (-4.6)	65.1 (+0.8)	41.9 (-7.8)	10.7 (-7.5)

Table 5: Influence of the threshold δ.

δ	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Accuracy	45.47	45.37	45.55	44.83	45.52	45.59	45.42	45.08	**45.83**	44.93

Experiments

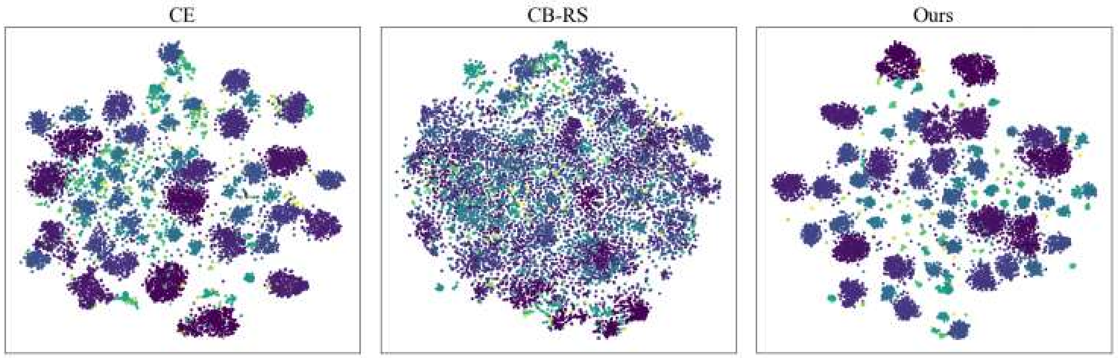


Figure 10: Visualization of learned representation on CIFAR100-LT.

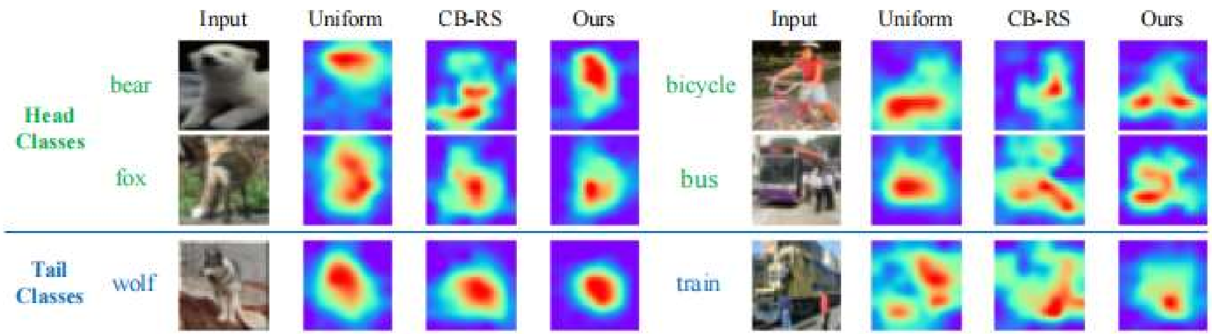


Figure 11: Visualization of features with Grad-CAM on CIFAR100-LT. Our method can alleviate the negative impact on head-class samples caused by the overfitting problem.

Thank you