SWRM: Similarity Window Reweighting and Margins for Long-Tailed Recognition

Anonymous authors
Paper under double-blind review

Abstract

Real-world data usually obeys a long-tailed distribution. Many previous works merely focus on the superficial phenomenon that tail classes lack samples in long-tailed datasets, yet they do not conduct in-depth analysis on the datasets and the model prediction results. In this paper, we experimentally find that due to the easily confusing visual features between head- and tail classes, the cross-entropy model is prone to misclassify tailed samples as head classes with high appearance similarity. We propose a Similarity Window Reweighting and Margins (SWRM) algorithm to tackle this problem. Specifically, we pretrain a cross-entropy model to model category similarity, then a sliding window is adopted upon the modeling result to constrain the impact of similarity. We design weights for different classes with the help of similarity window, which is named Similarity Window Reweighting (SWR). Besides, different margins computed inside the similarity window will be assigned to different classes, this is called Similarity Window Margin (SWM). In a nutshell, SWR considers the category frequency difference and the category similarity impact simultaneously, so that the weight coefficients computed by SWR are more reasonable. SWM prompts the model to learn fine-grained features and is conducive to the model's discriminative ability. Therefore, our methods alleviate the issue of misclassification effectively. In order to enhance the robustness and generalization of the model, we introduce a learnable similarity vector and further propose a Dynamic Similarity Window Reweighting and Margins (DySWRM) algorithm, which spends less computation cost compared with SWRM. Extensive experiments verify our proposed approaches effectiveness and superiority over SOTA reweighting and logit adjustment methods.

1 Introduction

With the advent of large-scale high-quality image datasets, the study of image recognition has witnessed incredible development (He et al., 2016; Liang et al., 2020). However, these datasets are usually artificially balanced. Real-world data is inherently imbalanced and obeys a long-tailed distribution, i.e., a few classes (head/majority classes) dominate most of the training samples, while considerable classes (tail/minority classes) only possess limited data points. The model trained on balanced datasets suffers catastrophic performance decline when evaluating it on long-tailed datasets. Class re-balancing (Chawla et al., 2002; Wang et al., 2019; Ren et al., 2020; Tang et al., 2020; Hong et al., 2021) is the mainstream method that addresses the long-tailed problem. Though these approaches can mitigate the model performance drop, most of them only focus on such a superficial phenomenon that tail classes are short of samples, and they do not analyze the long-tailed datasets and the model prediction results in depth.

In order to thoroughly explore the difficulty of long-tailed recognition, we train a cross-entropy model on Cifar100-LT training set and evaluate the model on the test set. We find that due to the easily confusable visual features residing in head- and tail classes, the model is apt to incorrectly classify tailed samples as head classes with similar visual appearance, as shown in Table 1. For instance, 18% of the samples of the minority class *woman* are mispredicted as class *baby*, and 24% of samples are mispredicted as class *boy*. Note that in Cifar100-LT training set, class *baby* and class *boy* are of head classes. Thereat, one can conclude that the cross-entropy model barely misclassifies the samples of head classes as the minority, and meanwhile, the situation of samples misclassification among tail classes hardly happens. As evidenced in Table 1, tailed samples are easily misclassified as head classes with high appearance similarity. Figure 3 in the Appendix A displays some images with confusable visual features on Cifar100-LT training set.

Table 1: The statistics of prediction error on Cifar-100-LT test set. For the first and second columns, the values in brackets denote the number of samples on Cifar100-LT training set. *Error* is the percentage of samples which are misclassified into a certain category. For the fourth column, the numbers in brackets represent the decreased proportion of the misclassified samples.

Label	Prediction	Error for CE(%)	Error for SWRM(%)
woman (5)	baby (455)	18	14 (-4)
	boy (299)	24	18 (-6)
whale (6)	dolphin (123)	56	38 (-18)
train (7)	bus (273)	40	19 (-21)
tram (11)	bus (273)	55	48 (-7)
shrew (15)	beaver (415)	26	10 (-16)
shark (16)	dolphin (123)	36	28 (-8)
hedgehog (26)	bear (434)	18	12 (-6)
	beaver (415)	19	12 (-7)
plate (29)	bowl (314)	47	24 (-23)
pickup (33)	bus (273)	55	31 (-24)

In order to alleviate the issue of misprediction, we propose a Similarity Window Reweighting and Margins (SWRM) algorithm. We firstly pretrain a cross-entropy model on the long-tailed dataset to model the category similarity. Afterward, a sliding window is adopted upon the modeling result to constrain the impact scope of similarity. On the one hand, we devise weight coefficients for different classes based on the similarity window, this is Similarity Window Reweighting (SWR) algorithm. Unlike previous reweighting methods (Cui et al., 2019; Zhang et al., 2022) that only reweight samples according to the discrepancy of sample numbers, SWR takes such discrepancy as well as the impact of category similarity into consideration. Therefore, the weights computed by SWR are more reasonable. On the other hand, we enforce margin in logit space, which is named Similarity Window Margin (SWM), and the margin is measured by category similarity inside the sliding window. SWM imposes more penalty to the classification error of confusing classes, aiming to reduce the confusion between head- and tail classes. In this way, the meticulous distinguishing ability of the model can be enhanced.

When coping with different long-tailed datasets, it is inevitable for SWRM to pretrain a cross-entropy model. It is time-consuming and expensive. To address this problem, we introduce a learnable similarity vector to replace the similarity modeling result produced by the pretrained model. Compared with the entire pretrained model, the similarity vector only contains C (the number of classes in the dataset) trainable parameters, which spends less computation cost. Similarly, the weights and margins of different classes are computed based on the similarity vector and the sliding window. We name this method Dynamic Similarity Window Reweighting and Margins (DySWRM) algorithm.

Contributions. (1) We conduct in-depth analysis of the difficulty for long-tailed recognition and find what causes poor performance of long-tailed model is that the model is prone to erroneously classify tailed samples as head classes with easily confusable visual features. (2) e design two algorithms named SWRM and DySWRM dedicated to improving such misclassification problem, in which DySWRM is more robust and generalized than SWRM. (3) SWR takes the difference of category frequency and the impact of category similarity into account simultaneously, so that it can produce more reasonable weight coefficients. SWM penalizes errors on confusable categories and makes them more recognizable, which is conducive to discriminative feature learning. (4) Experimental results on three long-tailed benchmark datasets verify that our proposal is effective and superior to the SOTA reweighting and logit adjustment methods.

2 Related work

In long-tailed image recognition, due to the severe class imbalance and scarce samples in tail classes, the model learning is extremely skewed, which leads to difficulty for the model to distinguish similar visual features between head- and tail classes. One of the recent mainstream approaches to tackle this problem is class re-balancing, which includes resampling, logit adjustment method and reweighting. These methods balance the model learning by data engineering (Chawla et al., 2002; Drummond et al., 2003; Kang et al., 2020; Ren et al., 2020), logit margin and loss modification, respectively.

Logit adjustment method modifies the logit output of the model via class prior knowledge, so as to achieve the goal of learning re-balancing. Menon et al. (2021) propose to calibrate the logits by using an offset related to the estimates of class prior during sample prediction. Hong et al. (2021) regard long-tailed recognition as the problem of label distribution shift and propose a post-compensation strategy to adjust the logits by using the test data distribution in the inference phase. Tan et al. (2020) argue that tail classes are over-suppressed by the overwhelming discouraging gradients of head classes, which is not conducive to the learning of the minority. Hence, they elaborately design Equalization Loss, which can reduce the suppression of discouraging gradients via ignoring the logits of head classes when computing the prediction score. Wang et al. (2021b) carry on more thorough analysis, they believe that the discouraging gradients of easily confusable categories conduce to the discriminative ability of the model, thereat, those discouraging gradients should be preserved. Jitkrittum et al. (2022) point out that since the embedding of tail classes is diffuse, logit margin cannot guarantee the correctness of tailed samples prediction. Therefore, in addition to enforcing logit margin, they also regularize the feature distribution in the embedding space, which is called embedding margin. Different from these methods calibrating logits either by prior knowledge or by sample gradients, our SWM modifies logits with the help of category similarity.

Reweighting is to reweight each sample during the training phase so that the model will pay more attention to the learning of tail classes. In general, reweighting allocates different loss contribution (Wang et al., 2017; Lin et al., 2017; Smith, 2022; Sinha et al., 2022) or decision margin (Cao et al., 2019) to different classes according to the sample frequency (Wang et al., 2017; Cao et al., 2019), sample difficulty (Lin et al., 2017; Smith, 2022) and class difficulty Sinha et al. (2022), respectively. The works (Cui et al., 2019; Zhang et al., 2022) decide the weighting coefficients of different classes based on the effective number of samples in the training set, rather than the exact number of samples. Unlike using the sample number to design weights, Park et al. (2021) reweight samples according to their influence on the decision surface, so that a more generalized decision boundary can be trained. Instead of fixing the weight coefficients, Shu et al. (2019) update them during model parameter optimization by using a handful of unbiased meta-data. Although these methods can pose considerable performance gains, they unexpectedly have some shortcomings in practice. For example, CB Loss (Cui et al., 2019) calculates the weights only according to the category frequency, and meta-weight-net (Shu et al., 2019) needs additional unbiased data for optimizing the weights. Our SWR not only considers the difference of category frequency, but also takes the category similarity into account, to design the weights for each class, so that the calculated weights are more rational. By introducing a learnable similarity vector, DySWRM can also update the weights during the training process without any additional data for learning.

3 MOTHODOLOGY

3.1 PRELIMINARY

Giving a long-tailed training set $\mathbb{D} = \{(x_i, y_i)\,|i \in \{1, 2, ..., N\}, y_i \in \{1, 2, ..., C\}\}$, where N and C are the total number of samples and classes, respectively, x_i denotes a sample and y_i is its corresponding label. When feeding x_i to the network, we can get its logits $z_i \in R^C$.

In practice, the cross-entropy model is prone to misclassify tailed samples as head classes that have easily confusing visual appearance, which results in extremely low classification accuracy on tail classes and the overall dataset. Therefore, we propose Similarity Window Reweighting and Margins (SWRM) algorithm to alleviate this problem, in which SWRM mainly includes two steps: category similarity modeling, weights and margins computing.

3.2 CATEGORY SIMILARITY MODELING

The first step of SWRM is category similarity modeling. Firstly, we train an initial model on to extract all image features. The loss function used to supervise model training is the conventional cross-entropy, which can be expressed as:

$$L_{ce}(x_i, c) = -log(\frac{e^{z_{i,c}}}{\sum_{j=1}^{C} e^{z_{i,j}}}) = log(1 + \sum_{j \neq c} e^{z_{i,j} - z_{i,c}}) \tag{1}$$

where $z_{i,j}$ is the j-th component of z_i, $j \in \{1, 2, ..., C\}$. Since there is more than one sample in a class, we leverage the average features (i.e., prototypes (Snell et al., 2017)) of each class to evaluate the category similarity. Specifically, we use a similarity assessment strategy to measure the similarity of the all the average features. This process aims to achieve the goal that the farther the two average features are, the lower similarity they hold. To this end, we leverage t-SNE (Van der Maaten & Hinton, 2008) to project the C average features that represent C categories into one-dimensional space, which will produce a projection vector. Due to the property of t-SNE that it reduces the feature dimensions according to the feature similarity, in the one-dimensional space, the distance of two classes with high similarity will be smaller than the ones that are rarely similar. The projection vector will be taken as the result of similarity modeling. Upon projection vector, a sliding window is adopted to constrain the scope of similarity impact.

3.3 Similarity window reweighting

In order to diminish the error of tailed sample predictions, the model should pay more attention to tail classes with high probability of being classified incorrectly. To be specific, in the training phase, we should assign larger weights to those classes, while assigning smaller ones to the head classes with confusable appearance. To this end, we design weight coefficients for different classes with the help of similarity window.

Through similarity modeling in Section 3.2, high similarity exists in the categories inside the similarity window. In other words, if a tail class and a head class are highly similar, they will appear in the same window. Since the effect of the categories with low similarity is quite subtle and can be ignored, we only leverage the impact of the categories locating in the same window to compute the weight coefficients. Such impact is called local impact and is measured by the ratio of sample number. For one category c, the larger the ratio is, the fewer samples it has compared with other categories. During training, the model will under-learn this category. Thereat, category c should obtain a larger weight. In addition to the local impact, we also exert the global impact of the entire datatset on category c. The global impact reflects the imbalance degree of the long-tailed dataset and it is measured by the maximum number of samples n_{max} and the number of samples n_c in category c. Formally, the weight of category c can be computed by:

$$w_c = log(1 + \frac{n_{max}}{n_c} * \frac{1}{W} \sum_{i=c-\frac{W}{2}}^{i=c+\frac{W}{2}} \frac{n_i}{n_c}) \tag{2}$$

where W is the window size and $W \in \{2, 3, ..., C\}$. For the computation of weight coefficients, previous works (Cui et al., 2019; Zhang et al., 2022), only leverage the difference of category frequency. Our SWR, however, gives consideration to such difference and the impact of category similarity concurrently, and is capable to allocate more rational weights to different classes, the evidence and in-depth analysis about this will be presented in Section 4.4 and Figure 2. Through Eq. 2, for one tail class that is easily confusable to a head class, it will obtain a large weight to increase its loss contribution, while that of the majority will be decreased. Hence, the model will give more focus on tail classes and effectively learn their characteristics. In this way, minority classes can be well-represented and the misclassification problem of them can be mitigated.

3.4 Similarity window margin

In the process of model parameter updating, since the scarcity of tail classes, the model will under-represent them. In addition, some majority classes have visual appearance that are similar to the minority, which hinders the fine-grained feature learning of tail classes. Such negative effect coming from the majority is called discouraging gradients in (Tan et al., 2020; Wang et al., 2021a). To get rid of these undesirably discouraging gradients, EQL (Tan et al., 2020) introduces a weight term to encourage the model to ignore them. However, Wang et al. (2021b) believe that the discouraging gradients from the categories with similar visual features should be retained, aiming to enhance the discriminative capability of the model. Differing from these approaches that modify the logits according to the gradients of samples, we argue that calibrating the logits via taking category similarity as "encouraging gradients" (actually it should be named positive effect) is conducive to eliminate the confusion between head- and tail classes, so that promote accurate classification, we present evidence in Section 4.4. Therefore, in order to prompt the model to learn discriminative feature representation, we propose Similarity Window Margins (SWM) algorithm.

SWM computes margins for different classes inside the similarity window to adjust the prediction logits. For category c, its allocated margin is computed as:

$$m_c = log(\frac{1}{W}\sum_{i=c-\frac{W}{2}}^{i=c+\frac{W}{2}}\frac{1}{sim(i,c)^{\beta}}) \tag{3}$$

where $sim(i, c)$ denotes the similarity between category i and category c, β is a hyper-parameter that controls the contribution of similarity to the margin. The larger the β is, the more sensitive the margin is to the change of similarity. SWM enforces logit margin with category similarity and penalises errors on similar classes more strongly. Therefore, it can prompt the model to learn fine-grained characteristics for the categories with confusable visual appearance. The intention of SWM is to increase the identifiability of confusing categories and improve the recognition of tail classes. Surprisingly, we experimentally find that SWM is also conducive to the classification of head classes (see Section 4.4 and Table 5).

3.5 SWRM and DySWRM

The conventional cross-entropy is defined as Eq. 1. Just as our analysis, due to the extreme class imbalance and limited samples of tail classes, the cross-entropy model is incompetent to depict the real feature distribution of the minority, which leads to the incorrect prediction problem. In this paper, we alleviate this issue with SWRM algorithm.

Specifically, after computing the weights and margins for each class by Eq. 2 and Eq. 3, respectively, we can define SWRM:

$$L_{SWRM}(x_i, c) = -w_c log(\frac{e^{z_{i,c}+m_c}}{e^{z_{i,c}+m_c} + \sum_{j\neq c} e^{z_{i,j}}}) = w_c log(1 + \sum_{j\neq c} e^{-m_c+z_{i,j}-z_{i,c}}) \tag{4}$$

SWRM allocates larger weights to the easily misclassified tail classes and smaller ones to head classes with confusable visual appearance, so that the model can pay more attention to those tail classes and promote their classification. Besides, it enforces margin in logit space, which can diminish the confusion and conduces to meticulous representation learning. SWRM Loss has a strikingly homogeneous form to (Cao et al., 2019; Menon et al., 2021; Jitkrittum et al., 2022). However, it is the category similarity that is used to enforce logit margin for SWRM, rather than label frequency. In other words, SWRM encourages large relative margin among similar categories.

However, the weights and margins computed by SWRM are fixed in the training phase. When dealing with different long-tailed datasets, SWRM needs to pretrain a cross-entropy model on the datasets in advance for category similarity modeling. Undoubtedly, it brings additional computation cost. In addition, the performance of SWRM algorithm severely relies on the similarity modeling result, we present evidence in Section 4.4 and Figure 1. In order to train a more robust and generalized model, we further propose Dynamic Similarity Window Reweighting and Margins (DySWRM) algorithm. To be specific, inspired by (Hu et al., 2021) that the authors devise a confidence bank dedicated to distinguishing the under-represented classes, we introduce a learnable similarity vector to replace the similarity modeling result, i.e., the projection vector mentioned in Section 3.2. The similarity vector only contains C optimizable parameters, which brings negligible computation cost compared with pretraining the entire model. In the training phase, the similarity vector combined with sliding window is used to help DySWRM calculate the weights and margins. Through similarity vector, DySWRM is capable to seamlessly apply to different long-tailed datasets without pretraining model. The weights, margins and objective function of DySWRM are computed by Eq. 2, Eq. 3 and Eq. 4, respectively.

Table 2: Experimental results on Cifar100-LT with ResNet-32. † denotes results are reproduced by us with released code. ‡ means that the results are copied from (Menon et al., 2021).

	Imbalance factor		
Method	100	50	10
CE	38.32	43.85	55.71
Focal Loss (Lin et al., 2017)	38.41	44.32	55.78
CB Loss (Cui et al., 2019)	39.6	45.32	57.99
LDAM-DRW (Cao et al., 2019)	42.04	47.30	58.71
Logit Adjustment (Menon et al., 2021)	43.89	-	-
LADE (Hong et al., 2021)	45.4	50.5	61.7
TDE (Tang et al., 2020)	44.1	50.3	59.6
EQL (Tan et al., 2020)‡	42.74	-	-
IB-Focal (Park et al., 2021)	42.06	47.49	58.20
Seesaw (Wang et al., 2021a)	40.87	-	57.83
GHM-CWAP (Zhang et al., 2022)	41.59	-	57.81
CFL (Smith, 2022)†	42.71	48.66	60.87
CDB-W-CE Sinha et al. (2022)	42.59	-	58.74
ELM (Jitkrittum et al., 2022)	45.77	-	-
SWRM (ours)	**46.86**	51.70	62.19
DySWRM (ours)	**46.82**	**51.92**	**62.28**

4 Experiments

4.1 Datasets

Following (Cui et al., 2019; Liu et al., 2019), we construct three long-tailed datasets: Cifar100-LT, ImageNet-LT and Places-LT. The construction of Cifar100-LT is consistent with (Cui et al., 2019). In the training set, each category has $n_c = n_{max} \times \mu^{-\frac{c}{C}}$ samples, where C is the number of classes, n_{max} denotes the maximum number of samples owned by a category, μ is the imbalance factor, and it is set to 100, 50 and 10. ImageNet-LT and Places-LT are sampled from the balanced versions (Russakovsky et al., 2015) and (Zhou et al., 2017) following the Pareto distribution with the power value of α=6. In ImageNet-LT, there are 115.8k samples and the number of samples per category is between 5-1280. Places-LT contains 62.5k samples from 365 categories with a maximum of 4980 samples and a minimum of 5 samples per category.

4.2 Implementation details

Evaluation metric. Top-1 accuracy (%) is used to evaluate the recognition performance of the model on different datasets. Besides, as in (Liu et al., 2019), we split the long-tailed datasets into three subsets. They are Many-shot (categories with more than 100 samples), Medium-shot (categories with more than 20 samples but less than 100 ones) and Few-shot (categories with less than 20 samples), respectively.

Parameter setup. For Cifar100-LT, we train the ResNet-32 (He et al., 2016) for 200 epochs and use an initial learning rate of 0.1, which warms up to 0.1 in the first 5 epochs and is scaled down after 140 and 180 epochs. The window size W and the hyper-parameter β of SWRM are set to 15 and 1, respectively. Similar to LDAM-DRW (Cao et al., 2019), we leverage a two-stage training strategy and let SWRM work at the last 60 epochs. On ImageNet-LT, we train the ResNet-10 (He et al., 2016) for 90 epochs. The learning rate warms up to 0.1 and is decayed at epochs 60 and 80 by 0.1. Considering that there are a large number of categories on ImageNet-LT, we set W and β to 30 and 10. SWRM works at the last 10 epochs. The pretrained ResNet-152(Liu et al., 2019; Kang et al., 2020) is used for Places-LT. We train it for 30 epochs and use an initial learning rate of 0.01 and decay it at epochs 10 and 20 by 0.1. Since the class imbalance is more serious on Places-LT, we set the value of W and β to 40 and 10, respectively. At the last 10 epochs, SWRM is used to supervise the model training.

Unless otherwise stated, we use a batch size of 128 and the model is optimized by SGD with momentum 0.9, weight decay 0.0001. In all experiments, the experimental setup of DySWRM is the same as SWRM. Besides, for the value of W and β, we provide more experimental analysis in Appendix B.

4.3 Comparison with previous methods

Experiments on Cifar100-LT. Table 2 displays the comparative experimental results of different approaches using three imbalance factors on Cifar100-LT. From the table we can observe that compared with CE, all approaches have improved their recognition performance to varying degrees. Although our methods perform slightly different, they deliver the best results in all situations, which indicates that our methods are effective ways to deal with long-tailed image recognition.

Experiments on ImageNet-LT. Table 3 shows that our proposal outperforms existing SOTA reweighting and logit adjustment methods by a large gap, which is attributed to SWR and SWM. On the one hand, SWR assigns larger weight coefficients to tail classes that are easily misclassified and meanwhile assigns smaller ones to head classes with high appearance similarity, so that the model can pay more attention to tail classes. On the other hand, SWM adjusts logits with the help of category similarity, which is conducive to making the confusing categories more recognizable and improves the sample prediction.

Table 3: Experimental results on ImageNet-LT with ResNet-10.

Method	Many	Medium	Few	Overall
CE	49.4	13.7	2.4	23.9
Focal Loss (Lin et al., 2017)	36.4	29.9	16.0	30.5
Lifted Loss (Oh Song et al., 2016)	35.8	30.4	17.9	30.8
Range Loss (Zhang et al., 2017)	35.8	30.3	17.6	30.7
CB Loss (Cui et al., 2019)	43.1	32.9	**24.0**	35.8
LDAM-DRW (Cao et al., 2019)	45.3	34.1	19.3	36.3
EQL (Tan et al., 2020)	49.4	32.3	14.5	36.4
CDB-W-CE (Sinha et al., 2022)	-	-	-	38.50
SWRM (ours)	57.39	**38.87**	15.58	**42.71**
DySWRM (ours)	**57.60**	38.49	15.82	42.66

Table 4: Experimental results on Places-LT with ResNet-152.

Method	Many	Medium	Few	Overall
CE	**45.9**	22.4	0.36	27.2
Focal Loss (Lin et al., 2017)	41.4	34.8	22.4	34.6
Lifted Loss (Oh Song et al., 2016)	41.1	35.4	**24.0**	35.2
Range Loss (Zhang et al., 2017)	41.4	35.4	23.2	35.1
SWRM (ours)	44.47	37.50	20.19	**36.36**
DySWRM (ours)	44.62	**37.62**	18.63	36.15

Experiments on Places-LT. In Table 4, we present the comparisons of our proposal and previous methods on Places-LT. Compared with CE, the accuracy of our methods slightly drops on Many-shot. We suspect that with a larger window size on Places-LT, the weight coefficients that SWR assigns to head classes with confusable visual features are too small, which makes the model over-ignore the learning of head classes. Nevertheless, our methods deliver significant performance gains on Medium-shot and Few-shot, especially on Medium-shot, they surpass CE by 67.41% (for SWRM) and 67.94% (for DySWRM), which is mainly attributed to SWR. Therefore, our proposed methods consistently outperform previous approaches on the overall accuracy.

4.4 Ablation study

Influence of SWR and SWM. As two independent approaches, SWR and SWM can be used separately to promote the optimization of the model. In order to study how they influence the model, we carry on experiments on ImageNet-LT and the results are showed in Table 5. Baseline is the model trained by cross-entropy (CE). One can see that the Baseline performs poorly on tail classes and the overall dataset. It is attributed to the fact that the CE model under-represents tail classes because of the paucity of samples, which results in the issue that tailed samples are misclassified as head classes with high appearance similarity. SWM, however, applies such category similarity to enforce margin in logit space, prompting the model to learn discriminative feature representation aiming to eliminate the confusion. Thus, SWM improves the classification of tail classes significantly. It demonstrates that the category similarity is conducive to the accurate classification of samples. Surprisingly, SWM performs the best on Many-shot. It is understandable because head classes have abundant samples to underpin the model to well-represent them, and meanwhile, they enjoy the benefit of category similarity to calibrate the logits. Since SWR will allocate bigger weights to tail classes and smaller ones to head classes with confusable appearance, its performance of Many-shot is worse than SWM. Nevertheless, by balancing the model learning, SWR improves the recognition of tail classes significantly, so that its overall performance surpasses SWM by a large gap. When combining SWR and SWM, the performance of both SWRM and DySWRM is greatly improved. This is because both our methods share the goal of alleviating the misprediction problem of tailed samples.

Table 5: Experimental results of investigating the influence of SWR and SWM on ImageNet-LT.

Method	Many	Medium	Few	Overall
Baseline (CE)	49.4	13.7	2.4	23.9
SWM	**59.07**	32.34	10.17	39.56
SWR	55.61	37.89	15.68	41.58
SWRM	57.39	**38.87**	15.58	**42.71**
DySWRM	57.60	38.49	**15.82**	42.66

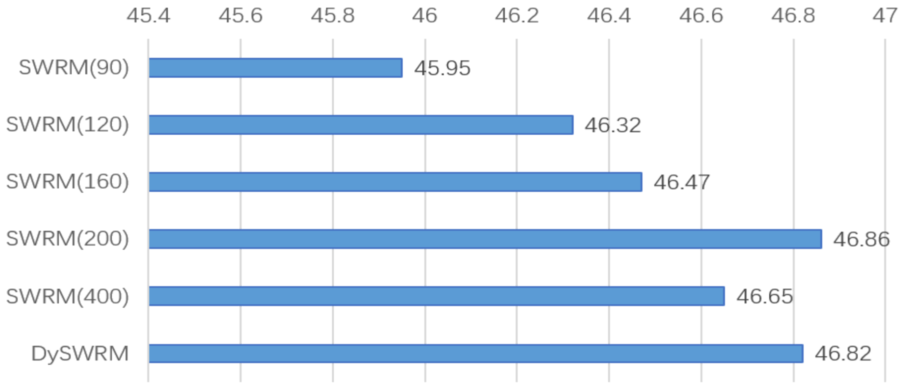
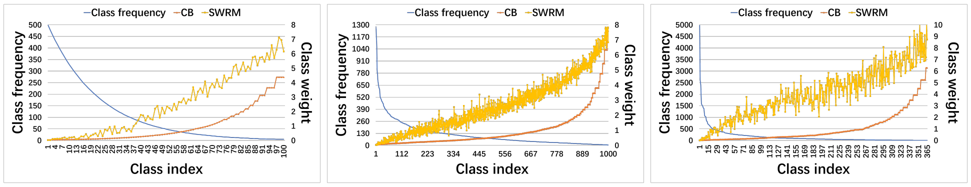


Figure 1: The influence of similarity modeling on model performance. Experiments are conducted on Cifar100-LT with μ=100. The numbers in brackets mean the pretraining epochs for the CE model. For example, SWRM (90) means that the CE model is pretrained with 90 epochs.

Influence of similarity modeling on model performance. Figure 1 illustrates the influence of the similarity modeling results on the model performance. In this figure, the CE model is pretrained for 90, 120, 160, 200 and 400 epochs, respectively, then it will be used to model the category similarity. From the figure we see that as the pretraining epoch increases, the classification accuracy of SWRM also rises. It is because the CE model with a longer pretraining period can better depict the feature distribution, which results in high-quality similarity modeling result. However, a too long pretraining time will lead to overfitting of the CE model, which damages the modeling of category similarity and further affects the classification of SWRM. Therefore, we can draw a conclusion that the recognition performance of SWRM depends on the similarity modeling result. This limits the practicability and applicability of SWRM, i.e., when applying SWRM to different long-tailed datasets, it is inevitable to pretrain a model to evaluate the category similarity, which spends additional computation cost. To tackle this problem, we devise DySWRM that enforces on a learnable similarity vector. DySWRM is capable to work seamlessly on different long-tailed datasets without pretraining. Thereat, DySWRM poses better generality and practicability than SWRM. Besides, the performance of DySWRM is comparable to that of SWRM.

Weight distribution. In order to intuitively understand the property of SWRM, we visualize the weight distribution calculated by SWRM and CB Loss (Cui et al., 2019) on three datasets. The visualization results are presented in Figure 2. One can see that CB Loss allocates equal weight coefficients (where the curve is gentle) to the classes with the same number of samples, which is not sound enough. Imagine that there are two classes having identical number of samples in the dataset. The one is quite different from all other categories in visual features, so it is easy for the model to represent this class. However, easily confusing visual appearance emerges between the other class and some categories, which results in the difficulty for the model to distinguish this class. In this case, we should assign a larger weight to the class that is difficult to learn, rather than assigning equal weights to these two classes. Therefore, the way that allocating weights to different classes only according to category frequency cannot reflect the discrepancy of category similarity in visual features. Our SWRM takes the difference of category frequency and the impact of category similarity into account simultaneously, so that the weight distribution is more reasonable. From the figure we can see that even if two classes have the same number of samples, the weights allocated to them may be different.



(a) Cifar100-LT (b) ImagenNet-LT (c) Places-LT

Figure 2: The class weight distribution of different methods on (a) Cifar100-LT (μ=100), (b) ImageNet-LT, (c) Places-LT. Our SWRM assigns more reasonable weights to different classes.

Do SWRM achieve the goal? In order to investigate whether SWRM has achieved its goal, i.e., alleviating the problem that in long-tailed recognition, the CE model tends to wrongly predict tailed samples as head classes with confusing visual appearance, we conduct experiments on Cifar100-LT with an imbalance factor of 100. The experimental results are shown in Table 1. From the table we see that SWRM is capable to efficaciously reduce the occurrence of such misclassification. In other words, SWRM realizes its goal. We visualize the confusion matrix of SWRM in Appendix C.

5 Conclusion

In this paper, a SWRM algorithm is proposed to alleviate the problem that the samples of tail classes are prone to be classified incorrectly as head classes with confusable visual features. Specifically, we pretrain a cross-entropy model to model the category similarity and introduce a sliding window upon the modeling result to constrain the impact scope of similarity. Based on the similarity window, we compute weights and margins for different classes according to SWR and SWM algorithms, respectively. In SWR, the difference of category frequency and the impact of category similarity are considered concurrently. Therefore, the weights calculated by SWR are more reasonable. For SWM, it enjoys the benefit of category similarity to eliminate the confusion of head- and tail classes, which is conducive to the model's dicriminative ability. Besides, in order to learn a more robust and generalized model, and also to reduce the computation cost, we introduce a learnable similarity vector to substitute the similarity modeling result. The method that SWR and SWM work on such similarity vector and the sliding window is named DySWRM. Wide-ranging experiments prove the effectiveness and competitiveness of our methods. Meanwhile, our proposal outperforms the SOTA reweighting and logit adjustment methods.

REFERENCES

Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in neural information processing systems*, 32, 2019.

Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9268–9277, 2019.

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.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

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*, pp. 6626–6636, 2021.

Hanzhe Hu, Fangyun Wei, Han Hu, Qiwei Ye, Jinshi Cui, and Liwei Wang. Semi-supervised semantic segmentation via adaptive equalization learning. *Advances in Neural Information Processing Systems*, 34:22106–22118, 2021.

Wittawat Jitkrittum, Aditya Krishna Menon, Ankit Singh Rawat, and Sanjiv Kumar. Elm: Embedding and logit margins for long-tail learning. *arXiv preprint arXiv:2204.13208*, 2022.

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*, 2020.

Gaoyuan Liang, Haoran Mo, Ying Qiao, Chuxin Wang, and Jing-Yan Wang. Paying deep attention to both neighbors and multiple tasks. In *International Conference on Intelligent Computing*, pp. 140–149. Springer, 2020.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.

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

Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. In *9th International Conference on Learning Representations*. OpenReview.net, 2021.

Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4004–4012, 2016.

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*, pp. 735–744, 2021.

Jiawei Ren, Cunjun Yu, Xiao Ma, Haiyu Zhao, Shuai Yi, et al. Balanced meta-softmax for long-tailed visual recognition. *Advances in neural information processing systems*, 33:4175–4186, 2020.

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 Li Fei-Fei. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.*, 115(3):211–252, 2015.

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. *Advances in neural information processing systems*, 32, 2019.

Saptarshi Sinha, Hiroki Ohashi, and Katsuyuki Nakamura. Class-difficulty based methods for long-tailed visual recognition. *International Journal of Computer Vision*, 130(10):2517–2531, 2022.

Leslie N Smith. Cyclical focal loss. *arXiv preprint arXiv:2202.08978*, 2022.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017.

Jingru Tan, Changbao Wang, Buyu Li, Quanquan Li, Wanli Ouyang, Changqing Yin, and Junjie Yan. Equalization loss for long-tailed object recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11662–11671, 2020.

Kaihua Tang, Jianqiang Huang, and Hanwang Zhang. Long-tailed classification by keeping the good and removing the bad momentum causal effect. *Advances in Neural Information Processing Systems*, 33:1513–1524, 2020.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

Jiaqi Wang, Wenwei Zhang, Yuhang Zang, Yuhang Cao, Jiangmiao Pang, Tao Gong, Kai Chen, Ziwei Liu, Chen Change Loy, and Dahua Lin. Seesaw loss for long-tailed instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9695–9704, 2021a.

Tong Wang, Yousong Zhu, Chaoyang Zhao, Wei Zeng, Jinqiao Wang, and Ming Tang. Adaptive class suppression loss for long-tail object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3103–3112, 2021b.

Yiru Wang, Weihao Gan, Jie Yang, Wei Wu, and Junjie Yan. Dynamic curriculum learning for imbalanced data classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 5017–5026, 2019.

Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Learning to model the tail. *Advances in neural information processing systems*, 30, 2017.

Renhui Zhang, Tiancheng Lin, Rui Zhang, and Yi Xu. Solving the long-tailed problem via intra- and inter-category balance. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 2355–2359, 2022.

Xiao Zhang, Zhiyuan Fang, Yandong Wen, Zhifeng Li, and Yu Qiao. Range loss for deep face recognition with long-tailed training data. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5409–5418, 2017.

Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017.

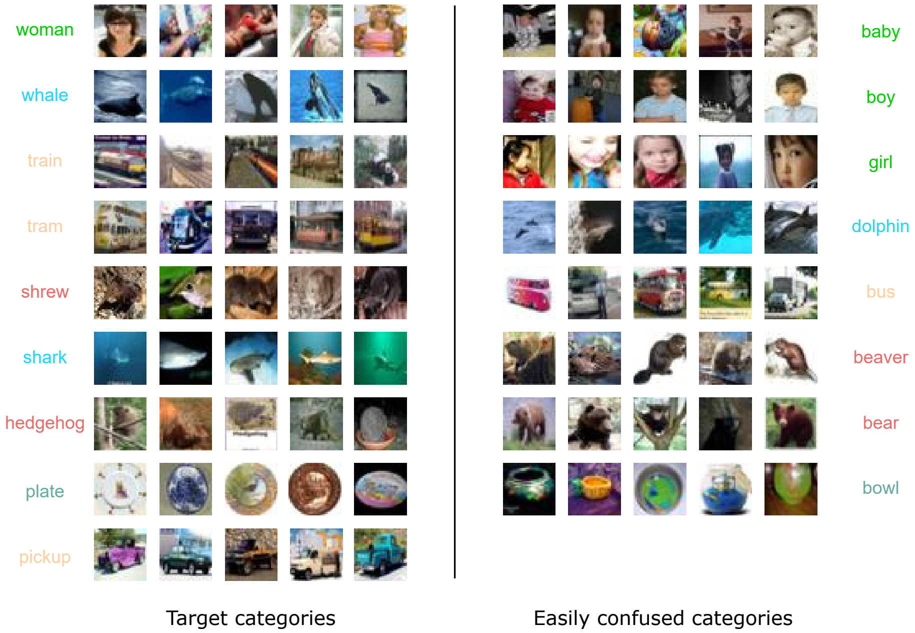


Figure 3: Some images with confusing visual appearance on Cifar100-LT training set. The images on the left are of target categories, which belong to tail classes, while those on the right are of easily confusable categories that usually come from head classes. As shown in Table 1, these images on the left are prone to be misclassified as the confused classes with high appearance similarity represented on the right.

A Some confusable categories

We present a part of images on Cifar100-LT training set. These images are from different categories and have similar visual appearance. For example, the class *woman* is similar to *baby*, *whale* is similar to *dolphin*. Due to the extreme class imbalance and the paucity of tailed samples, the model is incompetent to represent the feature embedding of minority classes. In addition of the overwhelming discouraging gradients (Tan et al., 2020; Wang et al., 2021a;b) coming from the majority, the model learning is highly biased to head classes, and the minority is over-suppressed. It leads to the problem that tailed samples are easily misclassified as head classes. In this paper, we propose SWRM and DySWRM algorithms to alleviate this problem.

B Further analysis of hyper-parameter

B.1 Window size W

Analysis of window size W. In SWRM and DySWRM, the sliding window is used to constrain the impact scope of category similarity. In order to explore the sensitivity of SWRM to the window size W, we conduct experiments on Cifar100-LT with an imbalance factor of 100. The experimental results are presented in Figure 4. In this experiment, the value of hyper-parameter β is fixed to 1, and we increase W from 2 to 100. As illustrated in this figure, before the window size increases to 15, the larger W is, the more categories with confusable visual features affect the calculation of weights and margins, and the higher overall accuracy can be obtained. Meanwhile, the performance of the model on the three subsets has been improved to varying degrees. It fully demonstrates that the difference of category frequency and the impact of category similarity have a substantive influence on the model performance. When W gradually increases from 15 to 60, the model places more emphasis on tail classes, yet head classes get less attention. Thus, the classification accuracy on Many-shot drops slightly. Despite all this, the performance of the model on Medium-shot and Few-shot improves to varying degrees, which leads to negligible changes on the overall accuracy.

However, when W continues to increase, the obvious performance degradation has emerged on Many-shot, Medium-shot and the overall dataset. In the similarity window, two categories with large distance are quite different from each other, which weakens the impact of category similarity. That is the reason why a too big W brings inferior performance gains. Through the above analysis, we can conclude: the window size W has a great effect on the model performance, an appropriate one can make the model perform better.

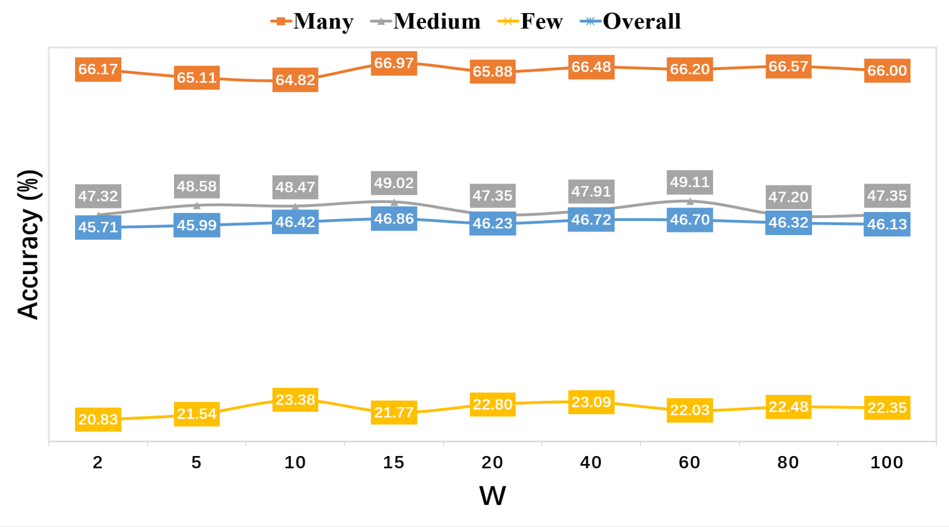


Figure 4: The effect of window size W. The experiments are conducted on Cifar100-LT with μ=100 and top-1 accuracy (%) is reported. Suitable window size can make the model perform well.

B.2 Hyper-parameter β

The hyper-parameter β affects the value of margins. Figure 5 studies the sensitivity of SWRM to β. These experiments are conducted on Cifar100-LT with an imbalance factor of 100. We fix the window size to 15 and vary β from 0 to 10. When the value of β is 0, SWRM neglects the impact of category similarity on the margins calculation, so the model delivers the lowest overall accuracy. As we increase the value of β, the performance of SWRM initially improves, however after a certain point (β=1) it starts to drop. When β=1, the model keeps a balance between the learning of head- and tail classes, so that SWRM delivers the highest overall performance. However, such a balance is broken as we continue to increase β, which results in performance degradation of the model. Therefore, we conclude that SWRM is sensitive to the hyper-parameter β, and an appropriate one is conducive to balancing the model learning.

C The confusion matrix of prediction results

As a complementary work to Table 1, we visualize the confusion matrices of cross-entropy and our SWRM. The comparative results are presented in Figure 6. It should be noted that in confusion matrix, the more dark points on the diagonal, the better the model performs. From the comparison of the fading color on the diagonal elements of both confusion matrices, SWRM delivers better recognition on tail classes. It indicates that a part of the samples in tail classes have been correctly predicted. And Table 1 further demonstrates that SWRM alleviates the problem that tailed samples are prone to be mispredicted as head classes.

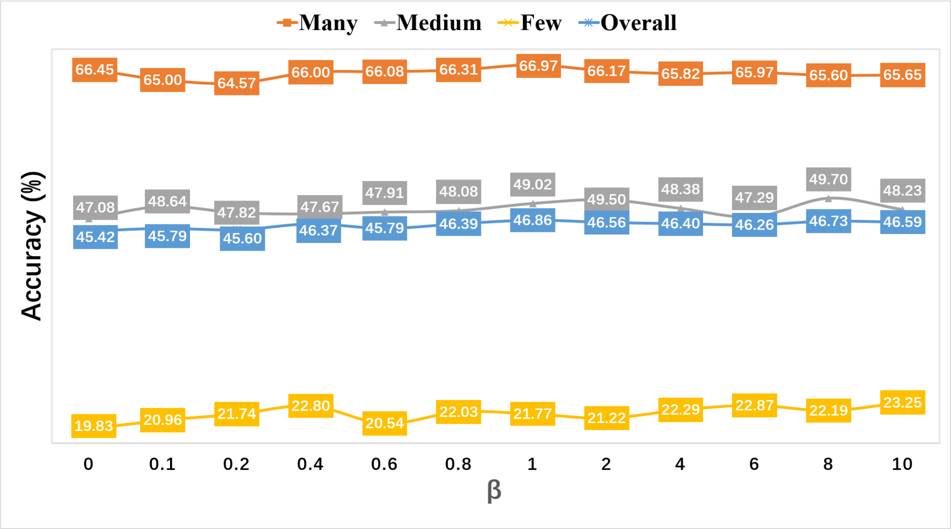
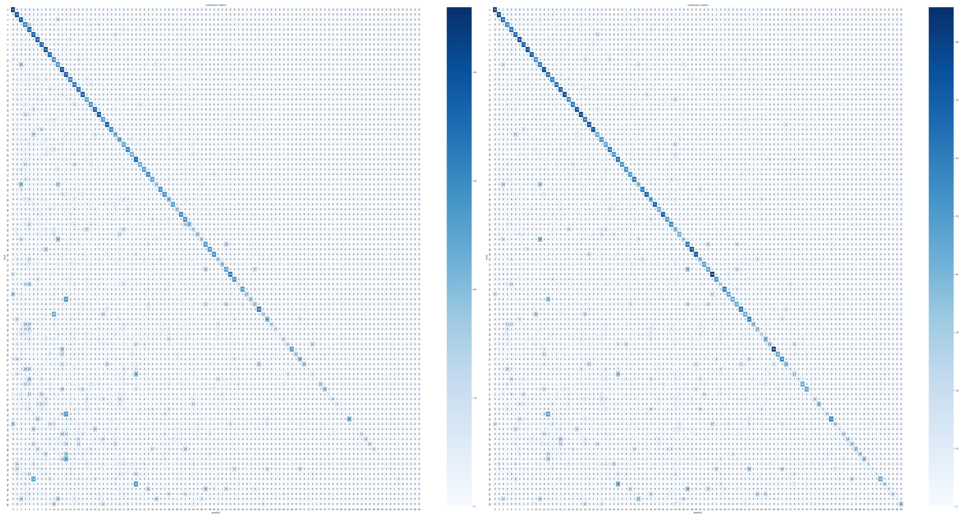


Figure 5: The effect of hyper-parameter β. The experiments are conducted on Cifar100-LT with μ=100 and on top-1 accuracy (%) is reported. Suitable value of β can keep a balance between the learning of head- and tail classes.



(a) CE

(b) SWRM

Figure 6: The confusion matrix of prediction results on Cifar100-LT. (a) The results of cross-entropy. (b) The results of SWRM. In confusion matrix, the darker the color is, the higher confidence the model has in the prediction results. In general, the more dark points on the diagonal line represents the better the model performing.