

# Adaptive Class Suppression Loss for Long-Tail Object Detection

Tong Wang<sup>1,2</sup>, Yousong Zhu<sup>1,3</sup>, Chaoyang Zhao<sup>1</sup>, Wei Zeng<sup>4,5</sup>, Jinqiao Wang<sup>1,2,6</sup>, Ming Tang<sup>1</sup>

<sup>1</sup> National Laboratory of Pattern Recognition, Institute of Automation,  
Chinese Academy of Sciences, Beijing, China

<sup>2</sup> School of Artificial Intelligence, University of Chinese Academy of Sciences,  
Beijing, China

<sup>3</sup> ObjectEye Inc., Beijing, China

<sup>4</sup> Peking University, Beijing, China

<sup>5</sup> Peng Cheng Laboratory, Shenzhen, China

<sup>6</sup> NEXWISE Co., Ltd., Guangzhou, China

{tong.wang, yousong.zhu, chaoyang.zhao, jqwang, tangm}@nlpr.ia.ac.cn  
weizeng@pku.edu.cn

## Abstract

To address the problem of long-tail distribution for the large vocabulary object detection task, existing methods usually divide the whole categories into several groups and treat each group with different strategies. These methods bring the following two problems. One is the training inconsistency between adjacent categories of similar sizes, and the other is that the learned model is lack of discrimination for tail categories which are semantically similar to some of the head categories. In this paper, we devise a novel Adaptive Class Suppression Loss (ACSL) to effectively tackle the above problems and improve the detection performance of tail categories. Specifically, we introduce a statistic-free perspective to analyze the long-tail distribution, breaking the limitation of manual grouping. According to this perspective, our ACSL adjusts the suppression gradients for each sample of each class adaptively, ensuring the training consistency and boosting the discrimination for rare categories. Extensive experiments on long-tail datasets LVIS and Open Images show that the our ACSL achieves 5.18% and 5.2% improvements with ResNet50-FPN, and sets a new state of the art. Code and models are available at <https://github.com/CASIA-IVA-Lab/ACSL>.

## 1. Introduction

With the advent of deep Convolutional Neural Network, researchers have achieved significant progress on object detection task. Many efforts have been paid to refresh the

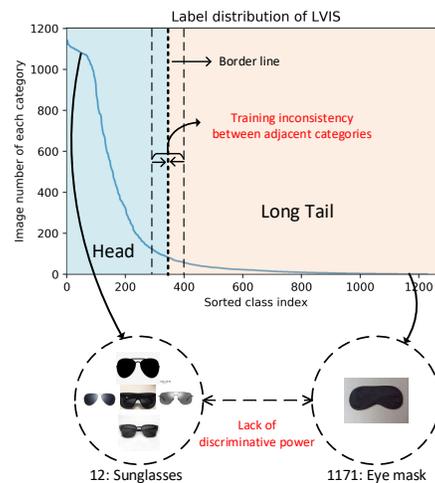


Figure 1: The label distribution of LVIS [11] dataset. The x-axis represents the sorted category index of LVIS. The y-axis is the image number of each category.

record on classic benchmarks like PASCAL VOC [9] and MS COCO [21]. However, these benchmarks usually have limited quantities of classes and exhibit relatively balanced category distribution. Whereas, in real-world scenarios, data usually comes with a long-tail distribution. A few head classes (frequent classes) contribute most of the training samples, while the huge number of tail classes (rare classes) are under-represented in data. Such extremely imbalanced class distribution proposes new challenges for researchers.

An intuitive solution is to re-balance the data distribution by re-sampling technique [12, 7, 28, 23]. By over-sampling

the tail or under-sampling the head classes, a less imbalanced distribution could be artificially generated. Nevertheless, over-sampling usually brings undesirable over-fitting issues on tail classes. And under-sampling may miss valuable information of head classes. The effectiveness of such techniques is limited when the dataset is extremely imbalanced. To improve the performance of tail classes while avoiding the over-fitting issue, Tan *et al.* [29] devises Equalization Loss, which argues that the poor performance of tail classes originates from the over-suppression of samples from head classes. Since tail classes only contain few samples, they receive much more negative gradients than positive ones during training, thus they are consistently in a state of being suppressed in most of the training time. In order to prevent tail classifiers from being over-suppressed, Equalization Loss proposes to ignore all negative gradients from head classes. Balanced Group Softmax (BAGS) [18] puts categories with similar numbers of training instances into the same group and computes group-wise softmax cross-entropy loss respectively. BAGS achieves relative balance within each group, thus can effectively ameliorate the domination of the head classes over tail classes.

The above methods can efficaciously reduce the suppression on tail classifiers. However, they need to partition the categories into several groups based on their category frequency prior. Such hard division between head and tail classes brings two problems, namely training inconsistency between adjacent categories and lack of discriminative power for rare categories. As shown in Figure 1, when two categories with similar instance statistics are divided into two different groups, there exists a huge gap between their training strategies. Such training inconsistency may deteriorate the network’s performance. In general, it is sub-optimal to distinguish the head and tail classes by the absolute number of samples. In addition, it frequently happens that two categories with high appearance similarity hold totally different sample frequency for datasets with large category vocabulary. For instance, category “sunglasses” and “eye mask” belong to the head and tail classes, respectively. To prevent the tail classifier “eye mask” from being over-suppressed, the negative gradients from category “sunglasses” is ignored by the classifier of “eye mask”. Nevertheless, this can also reduce the differentiability between these semantically similar cases in feature space, the classifier of “eye mask” has a high chance to misclassify sample “sunglasses” as “eye mask”. For those categories which belong to different groups but have high appearance similarity, it is hard for the network to learn a discriminative feature representation.

Therefore, in this paper, we present a novel Adaptive Class Suppression loss (ACSL) to address the above two problems. The design philosophy is simple and straightforward: *We assume all categories are from “tail” group,*

*and adaptively generate suppression gradients for each category according to their current learning status.* To be specific, we propose to treat all object categories as scarce categories, regardless of the statistics of per-category instances, thus eliminating the dilemma of manually defining the head and tail. Furthermore, in order to alleviate the insufficient learning and representation, we introduce an Adaptive Class Suppression Loss to adaptively balance the negative gradients between different categories, which effectively boosts the discriminative ability for tail classifiers. On the one hand, the proposed method can exterminate several heuristics and hyper-parameters of data distribution. On the other hand, it can also avoid the problem caused by over-sampling and under-sampling, and ensure the training consistency for all classes and sufficient learning of rare or similar categories. Finally, it yields reliable and significant improvements in detection performance on large-scale benchmarks like LVIS and Open Images.

To sum up, this work makes the following three contributions:

1. We propose a new statistic-free perspective to understand the long-tail distribution, thus significantly avoiding the dilemma of manual hard division.
2. We present a novel adaptive class suppression loss (ACSL) that can effectively prevent the training inconsistency of adjacent categories and improve the discriminative power of rare categories.
3. We conduct comprehensive experiments on long-tail object detection datasets LVIS and Open Images. ACSL achieves 5.18% and 5.2% improvements with ResNet50-FPN on LVIS and OpenImages respectively, which validates its effectiveness.

## 2. Related Works

**General Object Detection** Current CNN-based object detectors can be divided into anchor-based and anchor-free detectors based on whether they depend on the anchor heuristic. Classic anchor-based detectors consist of one-stage and two-stage approaches. Two-stage detectors [27, 10, 6, 1, 25, 13, 19, 24] first generate coarse object candidates by a Region Proposal Network (RPN). And then the region features of these proposals will be extracted for accurate classification and bounding box regression. One-stage detectors [22, 37, 20, 3, 36, 35] have a more concise structure. They make predictions on multiple feature maps directly without proposal generation process, thus enjoying higher computational efficiency.

Anchor-free pipelines abandon the anchor mechanism. They first locate several predefined or self-learned keypoints and then group them to final detections. CornerNet [16, 17], ExtremeNet [38] and CenterNet [8] *etc.* represent one object as several predefined keypoints and detect

by predicting these keypoints. RepPoints [34] proposes a finer representation of objects as a set of sample points and learns to arrange them in a manner that bounds the spatial extent of an object. FCOS [30] solves object detection in a per-pixel prediction fashion. It predicts four distances for each location and filters those unreliable detections with novel centerness scores. They all perform well on balanced datasets. However, directly applying them to long-tail datasets achieves inferior performance due to the imbalanced data distribution. Thus, our intention is to improve the detectors’ performance on long-tail datasets.

**Long-tail Recognition** Re-sampling strategy is a typical technique for imbalanced datasets. Class-aware sampling [28] and Repeat factor sampling [11] share the same design philosophy that they aim to balance the data distribution by adopting different sampling frequencies for different categories. Re-weighting [33, 14] is another widely used method which works by assigning weights for different training samples to emphasize the training of tail samples. Cui *et al.* [5] further proposes Class balanced loss which first calculates the effective number of samples for each category and uses the effective numbers to re-balance the loss.

In addition, some works design specialized loss functions or training strategies to tackle the imbalanced issue. Equalization Loss [29] ignores the negative gradients from head samples to prevent the tail classifiers from being over-suppressed. Cao *et al.* [2] proposes label-distribution-aware margin loss motivated by minimizing a margin-based generalization bound. Balanced Group Softmax [18] puts categories with similar sample numbers to one group and applies softmax within each group. Balanced Meta-Softmax [26] devises an unbiased extension of softmax to accommodate the label distribution shift between training and testing. SimCal [31] proposes a simple calibration framework to more effectively alleviate classification head bias with a bi-level class balanced sampling approach. Kang *et al.* [15] design a decoupling training schema, which first learns the representations and classifier jointly, then obtains a balanced classifier by re-training the classifier with class-balanced sampling.

The above methods can relieve the imbalanced training to a great extent. However, they all depend on the class distribution which brings inconvenience when we apply them to new long-tail datasets. In this paper, we devise a more general loss function which does not rely on the category frequency prior and can also handle the long-tail datasets well.

### 3. Adaptive Class Suppression Loss

In this section, we start by analyzing the limitations of group-based methods. Then, we introduce our proposed

Table 1: Experiments on LVIS with different groups.

Groups	$mAP$	$AP_r$	$AP_c$	$AP_f$
(0,5)[5,∞)	22.74	6.83	22.14	29.83
(0,50)[50,∞)	25.30	15.11	24.99	29.77
(0,500)[500,∞)	25.66	13.19	25.98	30.25
(0,5000)[5000,∞)	23.89	8.27	23.87	30.16

Adaptive Class Suppression Loss in detail and summary its advantages compared with previous methods.

### 3.1. Limitations of Group-based Methods

To pursue a balanced training for tail classifiers, several works propose to divide the whole categories into several groups based on the category frequency prior and adopt different training strategies for different groups. For instance, Equalization Loss [29] ignores the negative gradients from samples of head categories to prevent the tail classifiers from being dominated by negative signals. Balanced Group Softmax [18] first puts categories with similar sample numbers into the same group and applies softmax cross-entropy loss within each group.

The group-based methods show their ability to improve the performance of tail classifiers. For these methods, how to properly divide all categories into different groups is of vital importance. We empirically find that the quality of grouping directly impacts the performance. We choose BAGS (one of the state-of-the-art group-based methods for long-tail object detection) with ResNet50-FPN as an example to illustrate the importance of proper grouping in LVIS dataset. For clarity, we divide all categories into 2 groups rather than the default 4 groups. We utilize different group partition strategies to train the detectors and report the  $mAP$ ,  $AP_r$  ( $r$  for rare categories),  $AP_c$  ( $c$  for common) and  $AP_f$  ( $f$  for frequent) in Table 1. As we can see from this table, the dividing line need to be settled in a proper range (50~500) to achieve satisfactory performance. Setting it to a too large or small number all deteriorate the performance.

Since the proper grouping is the precondition of good performance, it is necessary to divide groups based on the data distribution when applying group-based methods to other long-tail datasets. Nevertheless, different datasets usually have various data distributions. A good group partition for one dataset may be suboptimal for other datasets. As shown in Figure 2, models can obtain the best performance when the dividing line is set to 500 in LVIS. However, it is not a good partition for Open Images dataset since it has fewer categories and each category has more samples. Finding a proper group partition strategy for a new dataset could be laborious and time-consuming, which will limit the application of group-based methods in real-world scenarios. We naturally ask a question: can we devise a more general method that can be directly applied to other long-

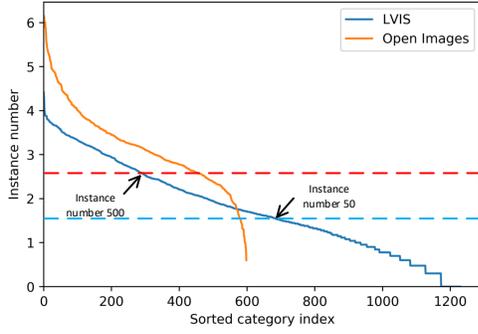


Figure 2: The data distribution of LVIS and Open Images dataset. The x-axis represents the sorted category index. Y-axis is the base-10 logarithm of the instance number.

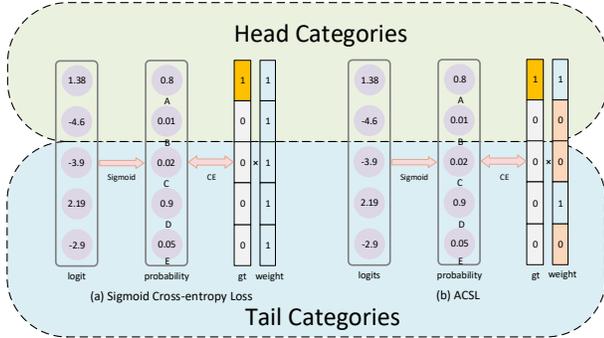


Figure 3: An illustration of Sigmoid Cross-entropy Loss and our proposed ACSL. The top two classes belong to head categories and the bottom three classes belong to tail categories. For ACSL, the hyper-parameter  $\xi$  is 0.7.

tail datasets without extra heuristics? To accomplish this goal, we propose a novel Adaptive Class Suppression Loss which does not need to perform group partition nor depend on the category frequency prior. It can be easily combined with different datasets seamlessly.

### 3.2. Formulation of ACSL

Based on the above analysis, we believe a satisfactory loss function for long-tail datasets should have the following two properties:

- (i) *The tail classifiers should be protected from being over-suppressed by the overwhelming head samples. In the meantime, negative gradients for easily confused categories should be preserved for discriminative learning.*
- (ii) *The loss function should better not depend on the label distribution prior so that it can be seamlessly applied to diverse datasets without re-calculating the new category distribution statistics.*

Giving a sample  $x_s$  belonging to the head category  $k$ , its label is a one-hot vector  $Y$ , in which  $y_k$  equals to 1 and  $y_i (i \neq k)$  is set to 0.  $z_i$  represents the network output logit of category  $i$ . By applying the sigmoid function to the logit  $z_i$  (Equation (1)), we can obtain  $p_i$ , the probability of current sample belonging to category  $i$ . The binary cross-entropy loss is formulated as Equation (2). Then we can derive the gradient of the loss function with respect to logit  $z_i$  as in Equation (4). For tail category  $i \neq k$ , BCE generates negative suppression gradients to force the classifier  $i$  to output low confidence. Such suppression gradients is beneficial to some extent. However, the excessive suppression gradients derived from the head categories will seriously hinder the positive activation of tail categories.

$$p_i = \frac{1}{1 + e^{-z_i}} \quad (1)$$

$$L_{BCE}(x_s) = - \sum_{i=1}^C \log(\hat{p}_i) \quad (2)$$

where,

$$\hat{p}_i = \begin{cases} p_i, & \text{if } i = k \\ 1 - p_i, & \text{if } i \neq k \end{cases} \quad (3)$$

$$\frac{\partial L_{BCE}}{\partial z_i} = \begin{cases} p_i - 1, & \text{if } i = k \\ p_i, & \text{if } i \neq k \end{cases} \quad (4)$$

Therefore, we devise a novel Adaptive Class Suppression Loss to protect the training of tail classifiers and adaptively choose which categories should be suppressed based on the learning status. As shown in Equation (5), we multiply a binary weight term  $w_i$  to the loss term  $-\log(\hat{p}_i)$  for category  $i$ . For category  $k$ ,  $w_i$  is set to 1 since the current sample belongs to category  $k$ . For other categories  $i (i \neq k)$ ,  $w_i$  controls whether the network applies suppression on category  $i$ . Here, we utilize the output confidence  $p_i$  as a signal to determine whether to suppress category  $i$ . If  $p_i$  is larger than the predefined threshold  $\xi$ , that means the network is confused between category  $i$  and  $k$ . Hence, we set  $w_i$  to 1 to perform discriminative learning. Otherwise,  $w_i$  will be set to 0 to avoid numerous unnecessary negative suppression. Rather than depending on the category distribution statistics, our proposed ACSL only relies on the network output confidences which saves the efforts of finding optimal category statistics related hyper-parameters when switching to a new dataset. The formulation is defined as follows:

$$L_{ACSL}(x_s) = - \sum_{i=1}^C w_i \log(\hat{p}_i) \quad (5)$$

where,

$$w_i = \begin{cases} 1, & \text{if } i = k \\ 1, & \text{if } i \neq k \text{ and } p_i \geq \xi \\ 0, & \text{if } i \neq k \text{ and } p_i < \xi \end{cases} \quad (6)$$

The gradient of the loss function with respect to  $z_i$  can be derived as Equation (7).

$$\frac{\partial L_{ACSL}}{\partial z_i} = \begin{cases} p_i - 1, & \text{if } i = k \\ w_i p_i, & \text{if } i \neq k \end{cases} \quad (7)$$

As shown in Figure 3, we give a simple illustration to better understand how ACSL works. For sigmoid cross-entropy loss (Figure 3(a)), it does not consider the imbalanced class distribution, and directly generate negative gradients for all other categories (the weight vector is filled with 1), thus leading to the tail categories receiving a large number of suppression information continuously, which severely reduces the discrimination ability of tail classifiers. For the proposed ACSL, the network can adaptively generate suppression gradients for tail categories. As show in Figure 3(b), the network simultaneously yields high confidence 0.9 for category ‘‘D’’, which means the category ‘‘A’’ is semantically similar to the category ‘‘D’’. Therefore, it is necessary to generate negative suppression for category ‘‘D’’, but not for the remaining tail categories with low confidences.

### 3.3. Advantages over Previous Methods

Compared with previous methods, the proposed ACSL has the following three characters that makes it more suitable for the long-tail datasets:

*ACSL takes the network learning status into consideration.* Previous methods [7, 28, 29, 18] only consider the label distribution, while ignore the network learning status. For instance, state-of-the-art Equalization Loss [29] calculates a binary weight for each category based on the sample numbers. Once the dataset is given, the weight will be determined and will not be changed during the whole training process. ACSL adaptively choose which categories to suppress based on the output confidences, which takes the network learning status into consideration, resulting in a more efficient learning process.

*ACSL works in a more fine-grained sample level.* Previous methods perform an identical operation on samples from the same category. For example, Class balanced loss [7] assigns the same weights for samples from one category. Equalization loss [29] generates the same weight masks for two samples if they all belong to the same category. These methods all ignore the diversity of variant samples. In contrast, ACSL calculates the category weights for a sample based on its output confidences. The generated category weights for samples from the same category might

Table 2: Experimental results of ACSL with different  $\xi$ .

	$\xi$	$mAP$	$AP_r$	$AP_c$	$AP_f$
baseline (1x)	–	21.18	4.30	20.09	29.28
baseline (2x)	–	22.28	7.38	22.34	28.17
ACSL	0.01	23.53	11.48	22.73	29.35
	0.1	25.11	16.04	24.72	29.22
	0.3	25.72	17.65	25.45	29.27
	0.5	26.08	18.61	25.85	29.36
	0.7	<b>26.36</b>	<b>18.64</b>	<b>26.41</b>	29.37
	0.9	25.99	17.25	26.0	<b>29.46</b>

Table 3: Results with larger backbones ResNet101, ResNeXt-101-64x4d and stronger detector Cascade R-CNN.

Models	Method	$mAP$	$AP_r$	$AP_c$	$AP_f$
Faster R101	baseline	22.36	3.14	21.82	<b>30.72</b>
	Ours	<b>27.49</b>	<b>19.25</b>	<b>27.60</b>	30.65
Faster X101	baseline	24.70	5.97	24.64	<b>32.26</b>
	Ours	<b>28.93</b>	<b>21.78</b>	<b>28.98</b>	31.72
Cascade R101	baseline	25.14	3.96	24.55	<b>34.35</b>
	Ours	<b>29.71</b>	<b>21.72</b>	<b>29.43</b>	33.26
Cascade X101	baseline	27.14	4.36	27.32	<b>36.03</b>
	Ours	<b>31.47</b>	<b>23.39</b>	<b>31.50</b>	34.66

be different. Thus, ACSL can control the training of each classifier in a more accurate way.

*ACSL does not depend on the class distribution.* The label distribution is indispensable for previous methods. They need to know the class distribution in advance to design sampling strategy [28, 11], determine the weights for samples [7] or divide categories into groups [18, 29]. This is inefficient since the strategies must be re-designed based on the new data distribution when we want to apply these methods to new long-tail datasets. In comparison, ACSL does not require the category frequency prior, which means it can be applied to new long-tail datasets seamlessly. We empirically find that ACSL works well on LVIS and Open Images under the same hyper-parameter setting.

## 4. Experiments on LVIS

### 4.1. Dataset and Setup

To validate the effectiveness of the proposed ACSL, we conduct comprehensive experiments on the long-tail Large Vocabulary Instance Segmentation (LVIS) dataset [11]. In this work, we use the LVIS-v0.5, which contains 1230 categories with both bounding box and instance mask annotations. We train the models with 57k *train* images and report the accuracy on 5k *val* images. LVIS divides all categories into 3 groups based on the number of images that

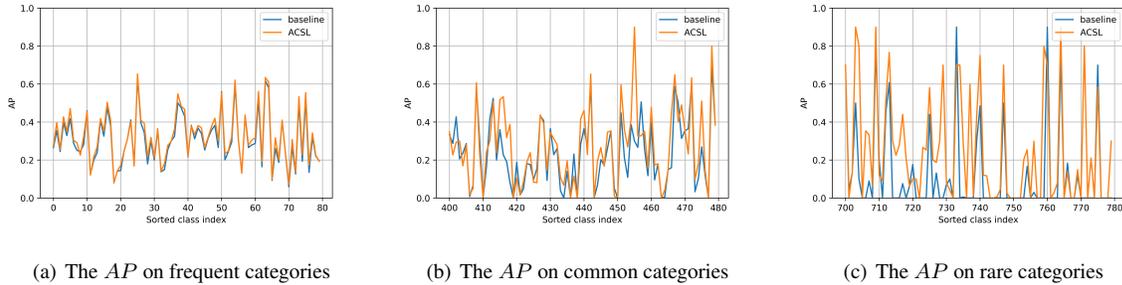


Figure 4: The  $AP$  of baseline and ACSL on frequent, common and rare categories, respectively. Both models are trained with ResNet50-FPN backbone. The x-axis is the sorted class index. The y-axis means the precision.

contain those categories: frequent ( $>100$  images), common (11-100 images) and rare ( $<10$  images). We use the official evaluation API<sup>1</sup> to evaluate all the models. Besides the official metrics  $mAP$ , the  $AP_r$  ( $AP$  for rare classes),  $AP_c$  ( $AP$  for common classes) and  $AP_f$  ( $AP$  for frequent classes) are also reported.

## 4.2. Implementation Details

We choose the classic two-stage detector Faster R-CNN with FPN structure as the baseline. The training image is resized such that the shorter edge is 800 pixels and the longer edge is no more than 1333. When training, we use 8 GPUs with a total batch size 16. The optimizer is stochastic gradient descent (SGD) with momentum 0.9 and weight decay 0.0001. The initial learning rate is 0.02 with 500 iterations' warm up. For 1x training schedule, learning rate decays to 0.002 and 0.0002 at epoch 8 and 11, respectively. The training stops at the end of epoch 12. When testing, we first apply Non-Maximum Suppression with IoU threshold 0.5 to remove duplicates. Then, the top 300 detections will be selected as the final results. Other hyper-parameter settings like anchor scales and anchor ratios follow the default settings in MMDetection [4].

When we conduct experiments with ACSL or other specialized loss functions designed for long-tail datasets, we focus on the classification subnet of Faster R-CNN and replace the softmax cross-entropy loss with these specialized loss functions. Inspired by [15], we decouple the training of feature representation and the classifier. We train a naive Faster R-CNN detector in the first stage and fine-tune the classifier in the second stage. In object detection, the background samples are negative samples for all categories. We adopt a simple strategy to chase a relative balance for each classifier. Since the ratio between the image numbers of rare, common and frequent categories is approximately 1:10:100, we randomly choose 1%, 10% background samples for rare and common categories, respectively.

## 4.3. Ablation Study

We take the Faster R-CNN with ResNet50-FPN backbone as the baseline model. As we first train a naive Faster R-CNN detector for 12 epochs in the first stage and then fine-tune the classifier with ACSL for another 12 epochs, the total training epoch is 24. We also report the results of the baseline model trained with 24 epochs for a fair comparison. As the results in Table 2, the baseline (1x) model has a relatively high precision on frequent categories but an extremely low accuracy on rare categories, merely 4.3%. More training iterations bring benefits to rare and common categories and lift the overall  $mAP$  by 1.1%. Even so, the performance of rare categories is still unsatisfactory, which shows the bottleneck of the baseline model lies in the tail classes.

ACSL introduces a hyper-parameter  $\xi$  to define the easily confused categories. It is a trade-off between relieving over-suppression on tail classes and chasing discriminative learning. A small  $\xi$  means that most of the categories will be suppressed, which will suppress too much on tail categories. However, for an extremely large  $\xi$ , the network will only suppress categories with extremely high confidences while ignore most of the other categories, thus will weaken the classifier's discriminative power. To explore how  $\xi$  influences the performance, we conduct experiments with several different values and report the results in Table 2. As shown in this table, setting  $\xi$  to 0.01 improves the precision but the improvements are limited. As  $\xi$  grows larger, the  $mAP$  and  $AP_r$  also increase which proves that properly increasing  $\xi$  relieves the suppression on tail classifiers. When  $\xi$  increases to 0.7, the model achieves the best performance. The overall  $mAP$  is 26.36%, surpassing the baseline model by considerable 5.18%. Nevertheless, continuing to increase its value deteriorates the performance since it will weaken the discriminative power of the classifier. We empirically find  $\xi = 0.7$  works best under current setting.

## 4.4. Generalization on Stronger Models

To verify the generalization of our method, we further conduct experiments on stronger backbones and de-

<sup>1</sup> <https://github.com/lvis-dataset/lvis-api>

Table 4: Comparison with state-of-the-art methods on LVIS-v0.5 *val* dataset. **Bold** numbers denote the best results among all models. “ms” means multi-scale testing.

Methods	backbone	$mAP$	$AP_r$	$AP_c$	$AP_f$	$AP@0.5$	$AP@0.75$	$AP_s$	$AP_m$	$AP_l$
Focal Loss [20]	ResNet-50	21.95	10.49	22.42	25.93	35.15	23.91	18.66	28.59	31.46
CBL [5]		23.9	11.4	23.8	27.3	—	—	—	—	—
LDAM [2]		24.5	14.6	25.3	26.3	—	—	—	—	—
RFS [11]		24.9	14.4	24.5	29.5	41.6	25.8	19.8	30.6	37.2
LWS [15]		24.1	14.4	24.4	26.8	—	—	—	—	—
SimCal [31]		23.4	16.4	22.5	27.2	—	—	—	—	—
EQL [29]	ResNet-50	25.06	11.92	25.98	29.14	40.14	27.30	20.08	31.50	38.67
	ResNet-101	26.05	11.45	27.14	30.51	41.30	27.83	20.35	33.73	40.75
	ResNeXt-101-64x4d	28.04	15.03	29.14	31.87	44.06	30.07	22.19	34.52	42.97
BAGS [18]	ResNet-50	25.96	17.65	25.75	29.54	43.58	27.15	20.26	32.81	40.10
	ResNet-101	26.39	16.80	25.82	30.93	43.44	27.63	20.29	34.39	41.07
	ResNeXt-101-64x4d	27.83	18.78	27.32	32.07	45.83	28.99	21.92	35.65	43.11
ACSL (Ours)	ResNet-50	26.36	18.64	26.41	29.37	42.38	28.63	20.43	33.11	40.21
	ResNet-101	27.49	19.25	27.60	30.65	43.45	29.69	21.11	34.96	42.00
	ResNeXt-101-64x4d	28.93	<b>21.78</b>	28.98	31.72	45.54	31.19	22.16	35.81	43.43
	ResNet-50 (ms)	27.24	17.86	27.42	30.76	44.46	28.54	20.96	34.40	41.68
	ResNet-101 (ms)	28.23	17.42	28.40	32.32	44.73	30.13	21.86	35.43	44.06
	ResNeXt-101-64x4d (ms)	<b>29.47</b>	20.30	<b>29.45</b>	<b>33.15</b>	<b>46.82</b>	<b>31.55</b>	<b>22.52</b>	<b>37.32</b>	<b>45.51</b>

tectors. We replace the ResNet50 backbone with larger ResNet101 and ResNeXt-101-64x4d. The results are summarized in Table 3. Experimental results reveal that ACSL can still achieve competitive results on larger backbones. With ResNet101 backbone, ACSL outperforms the baseline model by 5.13%  $mAP$ . And we observe that ACSL brings significant performance gains for rare categories with various backbones (16.11%  $AP_r$  improvements for ResNet101, for instance), which demonstrates the advantage of ACSL when tackling with long-tail datasets. The advantage of ACSL still exists when we use a larger backbone ResNeXt-101. The  $mAP$  of ACSL is 4.23% higher than the baseline model for ResNeXt-101. Moreover, the utilization of ACSL is not limited to a certain type of detector. It can be easily combined with other detectors such as Cascade R-CNN. For Cascade R-CNN, we replace the softmax cross-entropy loss with ACSL on all 3 heads and fine-tune the classifiers’ weights in the second training stage. With ResNet-101 Cascade R-CNN detector, ACSL achieves 29.71%  $mAP$ , surpassing baseline model by 4.57%  $mAP$ . When using larger backbone ResNeXt-101, the overall  $mAP$  can be further pushed to 31.47%, outperforming the baseline by a significant 4.33%. It is worth noticing that ACSL obtains the best performance on the overall  $mAP$ ,  $AP_r$  and  $AP_c$ , which further proves ACSL’s ability to tackle long-tail datasets.

#### 4.5. Performance Analysis

In order to have a more intuitive sense of how ACSL influences the network’s performance, we visualize the performance of baseline and ACSL on different categories in

Figure 4. Figure 4(a) shows the  $AP$  on frequent categories. The two curves almost overlap with each other which demonstrates that ACSL does not harm the training of head classifiers. For common categories (Figure 4(b)), ACSL begins to show its advantages over baseline model. The precision curve of ACSL almost covers that of the baseline model, which shows that ACSL brings moderate improvements for common categories. As the decreasing of categories’ sample numbers, the advantages of ACSL becomes more significant. As showed in Figure 4(c), ACSL outperforms baseline by a landslide. The integral area of the orange curve (ACSL) is much larger than that of the blue one (baseline). These three figures indicate that ACSL is able to improve the performance of tail categories without sacrificing the precision of head classes.

#### 4.6. Comparison with State of the Arts

In this section, we compare the performance of the proposed ACSL with other state-of-the-art methods and report the results in Table 4. All the models adopt Faster R-CNN with FPN structure. As shown in the table, without any bells and whistles, our single ResNet-50 model achieves 26.36%  $mAP$ . It surpasses other competitive methods, including Equalization Loss (25.06%  $mAP$ ) and BAGS (25.96%  $mAP$ ). And we do observe that the performance gain comes from the tail classes (rare and common categories). Among all the single ResNet-50 models, ACSL obtains the best performance on rare (18.64%) and common categories (26.41%). In the meantime, it does not sacrifice the precision of head categories in exchange for the im-

Table 5: Experiments on Open Images with different backbones.

Backbone	Methods	<i>AP</i>
ResNet50-FPN	baseline	55.1
	ours	<b>60.3</b>
ResNet101-FPN	baseline	56.3
	ours	<b>61.6</b>
ResNet152-FPN	baseline	57.4
	ours	<b>62.8</b>

provements on tail classes. The observations above exhibit the ability of ACSL to tackle the class imbalance problem in long-tail datasets.

The advantages of ACSL becomes more significant when it is combined with larger backbones. With ResNet-101 backbone, ACSL achieves 27.49% *mAP*, outperforming Equalization Loss and BAGS by 1.44% and 1.1% *mAP*, respectively. For stronger backbone ResNeXt-101-64x4d, ACSL obtains a 28.93% *mAP*, surpassing the counterparts of Equalization Loss (28.04% *mAP*) and BAGS (27.83% *mAP*). When applying multi-scale testing strategy, the performance can be further improved. Finally, our best model achieves 29.47% *mAP*.

## 5. Experiments on Open Images

Open Images V5 is the largest existing object detection dataset which contains a total of 16M bounding boxes for 600 object classes on 1.9M images. For such a large scale dataset, it is hard to maintain a relatively balanced class distribution. In fact, it suffers from severe class imbalance. For instance, the most frequent category “Man” contains 378077 images, which is 126 thousand times of the rarest category “Paper cutter” (3 images). The whole dataset consists of three parts: *train*, *validation* and *test*. All models are trained with 1.7M *train* images and evaluated on *validation* dataset. When performing evaluation, we use the official evaluation code for Open Images which evaluates only at IoU threshold 0.5. To reduce the training time on such a large dataset, we use the large batch training framework LargeDet [32] to train our models.

To testify the effectiveness of ACSL, we conduct experiments with various backbones including ResNet50, ResNet101 and ResNet152. The hyper-parameter  $\xi$  of ACSL is set to 0.7, the same value as in LVIS. Since objects in Open Images have multiple labels, we train the models under multiple label setting. The experimental results are summarized in Table 5. As shown in this table, the models trained with ACSL outperform the baseline models by a large margin. For ResNet50-FPN detector, ACSL achieves 60.3% *AP*, 5.2% higher than the baseline. For larger backbone ResNet152, our model obtains 62.8% *AP*, surpassing the baseline by 5.4% *AP*. To validate that the perfor-

Table 6: The detailed precision on some of the tail categories of Open Images.

	Spa	Scr	Fac	Cas	Hor
img num	38	46	49	53	54
baseline	35.0	46.6	17.8	19.9	8.3
ACSL	<b>41.6(+6.6)</b>	<b>55.6(+9.0)</b>	<b>80.9(+63.1)</b>	<b>47.5(+27.6)</b>	<b>16.6(+8.3)</b>
	Slo	Obo	Squ	Bin	Ser
img num	103	93	97	109	106
baseline	25.0	22.2	29.1	42.7	40.2
ACSL	<b>45.0(+20)</b>	<b>83.3(+61.1)</b>	<b>50.3(+21.2)</b>	<b>61.5(+18.8)</b>	<b>73.2(+33)</b>

Table 7: Comparison with other methods on Open Images. All models are trained with ResNet50-FPN backbone and evaluated on 500 categories.

Method	<i>AP</i>
Class Aware Sampling [28]	56.50
Equalization Loss [29]	57.83
Ours	<b>61.70</b>

mance gains mainly come from the rare categories, we extract the precisions of some rare categories and list them in Table 6. We can observe that ACSL achieves remarkable performance improvements for rare categories. For rare category “Face powder” (“Fac” for short), the performance gain is 63.1% *AP*. As shown in this table, ACSL brings considerable improvements to other rare categories, leading to higher overall precision.

We also compare our method with other methods on Open Images dataset. The results are summarized in Table 7. From Table 7, we can observe the performance advantage of our method with the same backbone. And the effectiveness of our method can be adequately validated.

## 6. Conclusion

In this work, we present a novel adaptive class suppression loss (ACSL) for long-tail datasets. ACSL is able to prevent the classifiers of tail categories from being over-suppressed by the samples from head categories. In the meantime, it preserves the discriminative power between easily confused categories. Equipped with ACSL, the models can achieve higher precision, especially on tail classes. Experimental results on long-tail datasets LVIS and Open Images validate its effectiveness. We will also apply the proposed ACSL to the long-tail image classification datasets in the future work.

**Acknowledgement:** This work was supported by the Research and Development Projects in the Key Areas of Guangdong Province (No.2019B010153001) and National Natural Science Foundation of China under Grants No.61772527, No.61976210, No.61876086, No.62002357, No.61806200, No.62002356, No.51975044 and No.61633002.

## References

- [1] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6154–6162, 2018. [2](#)
- [2] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arachiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *Advances in Neural Information Processing Systems*, volume 32, pages 1567–1578. Curran Associates, Inc., 2019. [3](#), [7](#)
- [3] Kean Chen, Jianguo Li, Weiyao Lin, John See, Ji Wang, Lingyu Duan, Zhibo Chen, Changwei He, and Junni Zou. Towards accurate one-stage object detection with ap-loss. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5119–5127, 2019. [2](#)
- [4] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. [6](#)
- [5] 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 Conference on Computer Vision and Pattern Recognition*, pages 9268–9277, 2019. [3](#), [7](#)
- [6] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 379–387, 2016. [2](#)
- [7] 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, pages 1–8. Citeseer, 2003. [1](#), [5](#)
- [8] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6569–6578, 2019. [2](#)
- [9] Mark Everingham, Luc Gool, Christopher K. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010. [1](#)
- [10] Ross Girshick. Fast r-cnn. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015. [2](#)
- [11] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5356–5364, 2019. [1](#), [3](#), [5](#), [7](#)
- [12] Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. Borderline-smote: a new over-sampling method in imbalanced data sets learning. In *ICIC'05 Proceedings of the 2005 international conference on Advances in Intelligent Computing - Volume Part I*, pages 878–887, 2005. [1](#)
- [13] Yihui He, Chenchen Zhu, Jianren Wang, Marios Savvides, and Xiangyu Zhang. Bounding box regression with uncertainty for accurate object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. [2](#)
- [14] Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. Learning deep representation for imbalanced classification. In *Computer Vision and Pattern Recognition*, pages 5375–5384, 2016. [3](#)
- [15] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition, 2020. [3](#), [6](#), [7](#)
- [16] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. *International Journal of Computer Vision*, 128(3):642–656, 2020. [2](#)
- [17] Hei Law, Yun Teng, Olga Russakovsky, and Jia Deng. Cornernet-lite: Efficient keypoint based object detection. *arXiv preprint arXiv:1904.08900*, 2019. [2](#)
- [18] Yu Li, Tao Wang, Bingyi Kang, Sheng Tang, Chunfeng Wang, Jintao Li, and Jiashi Feng. Overcoming classifier imbalance for long-tail object detection with balanced group softmax. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. [2](#), [3](#), [5](#), [7](#)
- [19] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. [2](#)
- [20] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):318–327, 2020. [2](#), [7](#)
- [21] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollr, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755, 2014. [1](#)
- [22] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. *European conference on computer vision*, pages 21–37, 2016. [2](#)
- [23] 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 *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 181–196, 2018. [1](#)
- [24] Jiangmiao Pang, Kai Chen, Jianping Shi, Huajun Feng, Wanli Ouyang, and Dahua Lin. Libra r-cnn: Towards balanced learning for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. [2](#)
- [25] Zheng Qin, Zeming Li, Zhaoning Zhang, Yiping Bao, Gang Yu, Yuxing Peng, and Jian Sun. Thundernet: Towards real-time generic object detection. *arXiv preprint arXiv:1903.11752*, 2019. [2](#)

- [26] Jiawei Ren, Cunjun Yu, Shunan Sheng, Xiao Ma, Haiyu Zhao, Shuai Yi, and Hongsheng Li. Balanced meta-softmax for long-tailed visual recognition, 2020. [3](#)
- [27] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2017. [2](#)
- [28] Li Shen, Zhouchen Lin, and Qingming Huang. Relay back-propagation for effective learning of deep convolutional neural networks. In *European conference on computer vision*, pages 467–482. Springer, 2016. [1](#), [3](#), [5](#), [8](#)
- [29] Jingru Tan, Changbao Wang, Buyu Li, Quanquan Li, Wanli Ouyang, Changqing Yin, and Junjie Yan. Equalization loss for long-tailed object recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. [2](#), [3](#), [5](#), [7](#), [8](#)
- [30] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9627–9636, 2019. [3](#)
- [31] Tao Wang, Yu Li, Bingyi Kang, Junnan Li, Junhao Liew, Sheng Tang, Steven Hoi, and Jiashi Feng. The devil is in classification: A simple framework for long-tail instance segmentation. 2020. [3](#), [7](#)
- [32] Tong Wang, Yousong Zhu, Chaoyang Zhao, Wei Zeng, Yaowei Wang, Jinqiao Wang, and Ming Tang. Large batch optimization for object detection: Training coco in 12minutes. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 481–496, Cham, 2020. Springer International Publishing. [8](#)
- [33] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Learning to model the tail. In *Advances in Neural Information Processing Systems*, pages 7029–7039, 2017. [3](#)
- [34] Ze Yang, Shaohui Liu, Han Hu, Liwei Wang, and Stephen Lin. Reppoints: Point set representation for object detection. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9657–9666, 2019. [3](#)
- [35] Shifeng Zhang, Longyin Wen, Xiao Bian, Zhen Lei, and Stan Z. Li. Single-shot refinement neural network for object detection. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4203–4212, 2018. [2](#)
- [36] Xiaosong Zhang, Fang Wan, Chang Liu, Rongrong Ji, and Qixiang Ye. Freeanchor: Learning to match anchors for visual object detection. In *Advances in Neural Information Processing Systems*, pages 147–155, 2019. [2](#)
- [37] Peng Zhou, Bingbing Ni, Cong Geng, Jianguo Hu, and Yi Xu. Scale-transferrable object detection. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 528–537, 2018. [2](#)
- [38] Xingyi Zhou, Jiacheng Zhuo, and Philipp Krahenbuhl. Bottom-up object detection by grouping extreme and center points. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 850–859, 2019. [2](#)