– Supplemental Material –
Relieving Long-tailed Instance Segmentation via Pairwise Class Balance

Yin-Yin He[1]*, Peizhen Zhang[2]*†, Xiu-Shen Wei[3,1], Xiangyu Zhang[2], Jian Sun[2]

[1]State Key Laboratory for Novel Software Technology, Nanjing University
[2]MEGVII Technology
[3]School of Computer Science and Engineering, Nanjing University of Science and Technology

heyy@lamda.nju.edu.cn, weixs.gm@gmail.com
{zhangpeizhen, zhangxiangyu, sunjian}@megvii.com

A. Details in implementation

Implentation of MS calibration. We found the original MS calibration mentioned in the main paper did not work well. The deep reason is: If instances of rare class i are always predicted as frequent class j with very high confidence (i.e., $M_{i,j} \approx 1$ and $M_{i,i} \approx 0$), then plenty of instances will be miscalibrated into class i ($s_i \approx 0$). To avoid this, we did two slight modifications to the original MS calibration in our experiments. Firstly, instead of using $s_i = M_{i,i}$, we adopted $s_i = \sum_{k=1}^{C} M_{k,i}$ to soften the distribution. Secondly, if s_i was still close to 0, we did not predict class i in this case. As shown in Table 1, the modifications do help improve the performance of MS calibration on each split.

Table 1. The performance of MS calibration before and after applied two modifications. Experiments are conducted on LVIS v0.5 using Mask R-CNN with ResNet-50-FPN and RFS [4] sampler.

Modification		AP	AP$_r$	AP$_c$	AP$_f$	AP^b
#1	#2					
	✓	20.0	13.5	19.1	23.7	19.9
✓		26.5	20.0	27.7	27.5	26.3
✓	✓	**27.0**	**20.6**	**28.2**	**28.0**	**27.0**

Applied to EQL v2 [11] and Seesaw [12]. PCB regularization can be easily applied to EQL v2 [11] and Seesaw loss [12] as they only change the loss weight or model activation. For each k-th proposal of label y at current iteration, the original equalization loss v2 can be formulated as follows:

$$L_{EQL\,v2}(k) = -\sum_{i=1}^{C} w_i \left[y_i \log \hat{p}_i + (1 - y_i) \log(1 - \hat{p}_i) \right] , \tag{1}$$

where $\hat{p}_i = 1/\left(1 + e^{-z_i^{fg}}\right)$ is the post-sigmoid probability for class i, y_i indicates whether the proposal belongs to class i, and w_i is a weight calculated from gradient perspective. When applied to EQL v2, our PCB regularization becomes

$$L_{PCB} = -\sum_{i=1}^{C} w_i \left[\hat{M}_{i,y}^{t} \log \hat{p}_i + (1 - \hat{M}_{i,y}^{t}) \log(1 - \hat{p}_i) \right] . \tag{2}$$

And the total classification loss function for the proposal is

$$L_{cls}(k) = \alpha L_{PCB}(k) + (1 - \alpha) L_{EQL\,v2} . \tag{3}$$

Similarly, we define the Seesaw variant of PCB regularization. The original Seesaw loss terms as

$$L_{seesaw}(k) = -\sum_{i=1}^{C} y_i \log \hat{p}_i ,$$
$$with\, \hat{p}_i = \frac{\exp(z_i^{fg})}{\sum_{j \neq i}^{C} \mathcal{S}_{ij} \exp(z_j^{fg}) + \exp(z_i^{fg})} . \tag{4}$$

The $\mathcal{S}_{ij}$ is composed by a mitigation factor and a compensation factor. So, the PCB regularization can be written as

$$L_{PCB}(k) = -\sum_{i=1}^{C} \hat{M}_{i,y}^{t} \log \hat{p}_i ,$$
$$with\, \hat{p}_i = \frac{\exp(z_i^{fg})}{\sum_{j \neq i}^{C} \mathcal{S}_{ij} \exp(z_j^{fg}) + \exp(z_i^{fg})} . \tag{5}$$

We combine them to get

$$L_{cls}(k) = \alpha L_{PCB}(k) + (1 - \alpha) L_{seesaw} . \tag{6}$$

Regression loss calculation. In practice, we calculate the regression loss only in the last recurrent step, rather than in

Table 2. Analysis on the influence of calculating the regression loss in each step or only last step. Experiments are conducted on LVIS v0.5.

Method	Regression	AP	AP$_r$	AP$_c$	AP$_f$	AP^b
Softmax	each	**25.1**	11.1	**25.9**	**29.6**	**25.3**
	last	**25.1**	**12.6**	25.5	29.5	25.2
RFS	each	27.5	20.4	**28.2**	29.4	27.8
	last	**27.7**	**21.8**	28.0	**29.7**	**28.2**

each step. So the overall objective becomes:

$$L = \left[\sum_{r=1}^{R} w_r L_{cls}^{r} \right] + L_{loc}^{R} + L_{mask} \, . \tag{7}$$

It shows similar performance to the later, while obtains higher AP$_r$. The results of comparison are shown in Tab. 2. While the overall mask AP and box AP are comparable for the two manners, there is a constant improvement in performance of rare classes (over 1 AP$_r$) for the former.

Training details. Following [12], we implement our method with mmdetection [1]. Mask R-CNN [5] with ResNet-50-FPN and ResNet-101-FPN [6, 8] is adopted as our baseline model. We utilize the standard 2× schedule for LVIS of both versions. The models are trained using SGD with 0.9 momentum and 0.0001 weight decay for 24 epochs. With batch size of 16 on 8 GPUs, the initial learning rate is set to 0.02 and is decreased by 0.1 after 16 and 22 epochs, respectively. The training data augmentations include scale jittering (640-800) and horizontal flipping. For evaluation, we set the maximum number of detections per image to 300 and the minimum score threshold to 0.0001, as [4].

B. Analysis on α w/o iterative learning paradigm

As discussed in Sec. 3.3 of the main paper, the performance will deteriorate soon with the increase of α if the iterative learning paradigm is not applied. Tab. 3 shows an example. As α increases, AP$_r$ gets improved until $\alpha = 0.6$, while AP$_c$ and AP$_f$ decline soon after $\alpha > 0.2$. So the PCB regularization hurts the fundamental classification, and the flexibility of debiasing is limited. By applying iterative learning paradigm, which guarantees the fundamental classification, such worry gets relieved. The room for debiasing is increased.

C. Comparing with refinement module in CrowdDet [2]

We notice that the refinement module (RM) in [2] is similar to our iterative learning paradigm. There are two main differences, RM concatenates the predictions and features rather than element-wise operation, and it utilizes features of the penultimate layer. We also provide the results of adopting RM as the learning paradigm, which are summarized in Tab. 4. While RM achieves promising AP$_r$ compared to PCB regularization, it hurts the performance of common classes and frequent classes much, so the overall AP drops. Different from RM, our proposed iterative learning paradigm guarantees the performance of common and frequent classes.

Table 3. Analysis on the influence of different PCB regularization coefficient α without iterative learning paradigm. Experiments are conducted with RFS on LVIS v0.5.

α	AP	AP$_r$	AP$_c$	AP$_f$	AP^b
0.0	25.6	16.0	26.4	28.5	25.6
0.2	**26.7**	18.9	**27.5**	**28.8**	**27.1**
0.4	26.5	18.3	27.4	28.7	26.7
0.6	26.2	**19.7**	26.4	28.4	26.6
0.8	26.0	18.5	26.7	28.1	26.5
1.0	25.1	18.7	25.9	26.5	25.3

Table 4. Comparison with refine module (RM in short) in CrowdDet [2]. Experiments are conducted with RFS on LVIS v0.5. PCB regularization is applied.

Paradigm	AP	AP$_r$	AP$_c$	AP$_f$	AP^b
N/A	26.7	18.9	27.5	28.8	27.1
RM	26.4	20.8	26.5	28.5	26.9
Iterative	**27.7**	**21.8**	**28.0**	**29.7**	**28.2**

D. Extension to long-tailed classification.

We also extend our PCB to long-tailed classification to testify its generalization ability. The commonly used ImageNet-LT [9] dataset is adopted in our experiment, and we use ResNeXt-50 [13] as the backbone network. Models are trained for 90 epochs with batch size 512. The intial learning rate is set to 0.2 and the first 5 epochs are trained with linear warm-up learning rate schedule [3]. The learning rate is deacyed at 60th and 80th epoch by 0.1. For the implementation of PCB, we ignore the MLP_{loc} and set the dimension of hidden layers in MLP_{cls} to 256 for simplicity. For a feature vector from the backbone, it will go through the same classifier for two times, and the last prediction is used for evaluation. γ is set to 0.9999. We train PCB in a decoupled manner [7], so the PCB regularizer is only applied in the fine-tune phase.

Two methods are utilized as baseline, CE and BSCE [10]. The results are in Tab. 5. Equipped with PCB (α is set to 0.8 and 0.15 respectively), the performance gain is significant and consistent, the accuracy of few split is raised almost 10% even on the strong baseline. The results fully demonstrate the generalization ability of our PCB.

Table 5. Accuracy on ImageNet-LT with a ResNeXt-50 backbone.

Method	PCB	Many	Medium	Few	Overall
CE	✗	**67.76**	38.89	7.44	45.73
	✓	61.68	**49.10**	**21.97**	**50.26**
BSCE [10]	✗	**62.65**	48.75	25.44	50.94
	✓	61.72	**49.66**	**34.85**	**52.29**

References

[1] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 2

[2] Xuangeng Chu, Anlin Zheng, Xiangyu Zhang, and Jian Sun. Detection in crowded scenes: One proposal, multiple predictions. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 12214–12223, 2020. 2

[3] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017. 2

[4] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5356–5364, 2019. 1, 2

[5] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Int. Conf. Comput. Vis.*, pages 2961–2969, 2017. 2

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 770–778, 2016. 2

[7] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. *arXiv preprint arXiv:1910.09217*, 2019. 2

[8] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2117–2125, 2017. 2

[9] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2537–2546, 2019. 2

[10] Jiawei Ren, Cunjun Yu, Shunan Sheng, Xiao Ma, Haiyu Zhao, Shuai Yi, and Hongsheng Li. Balanced meta-softmax for long-tailed visual recognition. *arXiv preprint arXiv:2007.10740*, 2020. 2, 3

[11] Jingru Tan, Xin Lu, Gang Zhang, Changqing Yin, and Quanquan Li. Equalization loss v2: A new gradient balance approach for long-tailed object detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1685–1694, 2021. 1

[12] 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 *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 9695–9704, 2021. 1, 2

[13] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1492–1500, 2017. 2