Seesaw Loss for Long-Tailed Instance Segmentation Supplementary Materials

Jiaqi Wang[1] Wenwei Zhang[2] Yuhang Zang[2] Yuhang Cao[1] Jiangmiao Pang[5] Tao Gong[6]
Kai Chen[3,4] Ziwei Liu[2] Chen Change Loy[2] Dahua Lin[1]

[1]SenseTime-CUHK Joint Lab, The Chinese University of Hong Kong
[2]S-Lab, Nanyang Technological University [3] SenseTime Research
[4] Shanghai AI Laboratory [5]Zhejiang University [6] University of Science and Technology of China
{wj017,cy020,dhlin}@ie.cuhk.edu.hk {wenwei001,zang0012,ziwei.liu,ccloy}@ntu.edu.sg
{pangjiangmiao,gongtao950513}@gmail.com chenkai@sensetime.com

1. Analysis of $\mathcal{S}_{ij}$ in Seesaw Loss

In this work, we propose Seesaw Loss to dynamically re-balance gradients of positive and negative samples for each category. Specifically, Seesaw Loss mitigates the overwhelming gradients of negative samples imposed by a head class i on a tail class j via decreasing the value of $\mathcal{S}_{ij}$ in the following formula,

$$\frac{\partial L_{seesaw}(\mathbf{z})}{\partial z_j} = \mathcal{S}_{ij}\frac{e^{z_j}}{e^{z_i}}\widehat{\sigma}_i, \quad \text{with } \widehat{\sigma}_i = \frac{e^{z_i}}{\sum_{j\neq i}^{C}\mathcal{S}_{ij}e^{z_j} + e^{z_i}}. \tag{A1}$$

To further analyze the effects of adjusting the value of $\mathcal{S}_{ij}$, we calculate the partial derivative of Eqn A1 with respect to $\mathcal{S}_{ij}$ as

$$\frac{\partial(\mathcal{S}_{ij}\frac{e^{z_j}}{e^{z_i}}\widehat{\sigma}_i)}{\partial \mathcal{S}_{ij}} = \frac{e^{z_j}(\sum_{k\neq i,j}^{C}\mathcal{S}_{ik}e^{z_k} + e^{z_i})}{(\sum_{j\neq i}^{C}\mathcal{S}_{ij}e^{z_j} + e^{z_i})^2} > 0. \tag{A2}$$

The value of the partial derivative in Eqn A2 is always positive. This indicates that the gradients of negative samples imposed by class i on class j will be reduced as the value of $\mathcal{S}_{ij}$ decreases.

2. How Seesaw Loss works

Via re-balancing gradients of positive and negative samples, Mask R-CNN [7] w/ Seesaw Loss significantly outperforms Mask R-CNN [7] w/ Cross-Entropy Loss on LVIS [6] dataset. Here, we conduct a quantitative analysis of the effectiveness of Seesaw Loss on re-balancing the gradients of positive and negative samples for each category. Specifically, we adopt Mask R-CNN [7] with ResNet-101 [8] backbone and FPN [11] as instance segmentation framework.

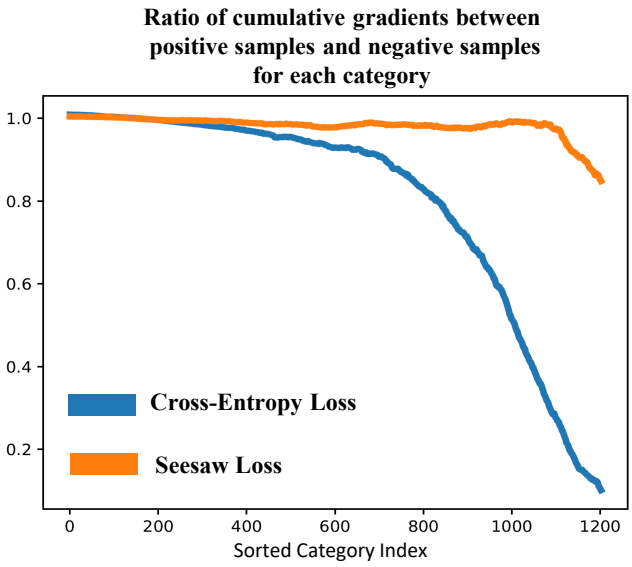

Figure A1: The distribution of the ratio of cumulative gradients between positive and negative samples for each category with Cross-Entropy Loss and Seesaw Loss, respectively. The categories are sorted in descending order with respect to their instance numbers. In contrast to Cross-Entropy Loss, Seesaw Loss effectively re-balances the gradients of positive and negative samples.

The Cross-Entropy Loss and Seesaw Loss are integrated into the framework and trained with random sampler by 2x schedule. We accumulate the gradients of positive and negative samples on predicted logit z_i of each category i during the whole training procedure.

Figure A1 shows the distribution of the ratio of cumulative gradients between positive and negative samples for each category in Mask R-CNN [7] with Cross-Entropy Loss and Seesaw Loss, respectively. With Cross-Entropy Loss, tail classes obtain heavily imbalanced gradients of positive

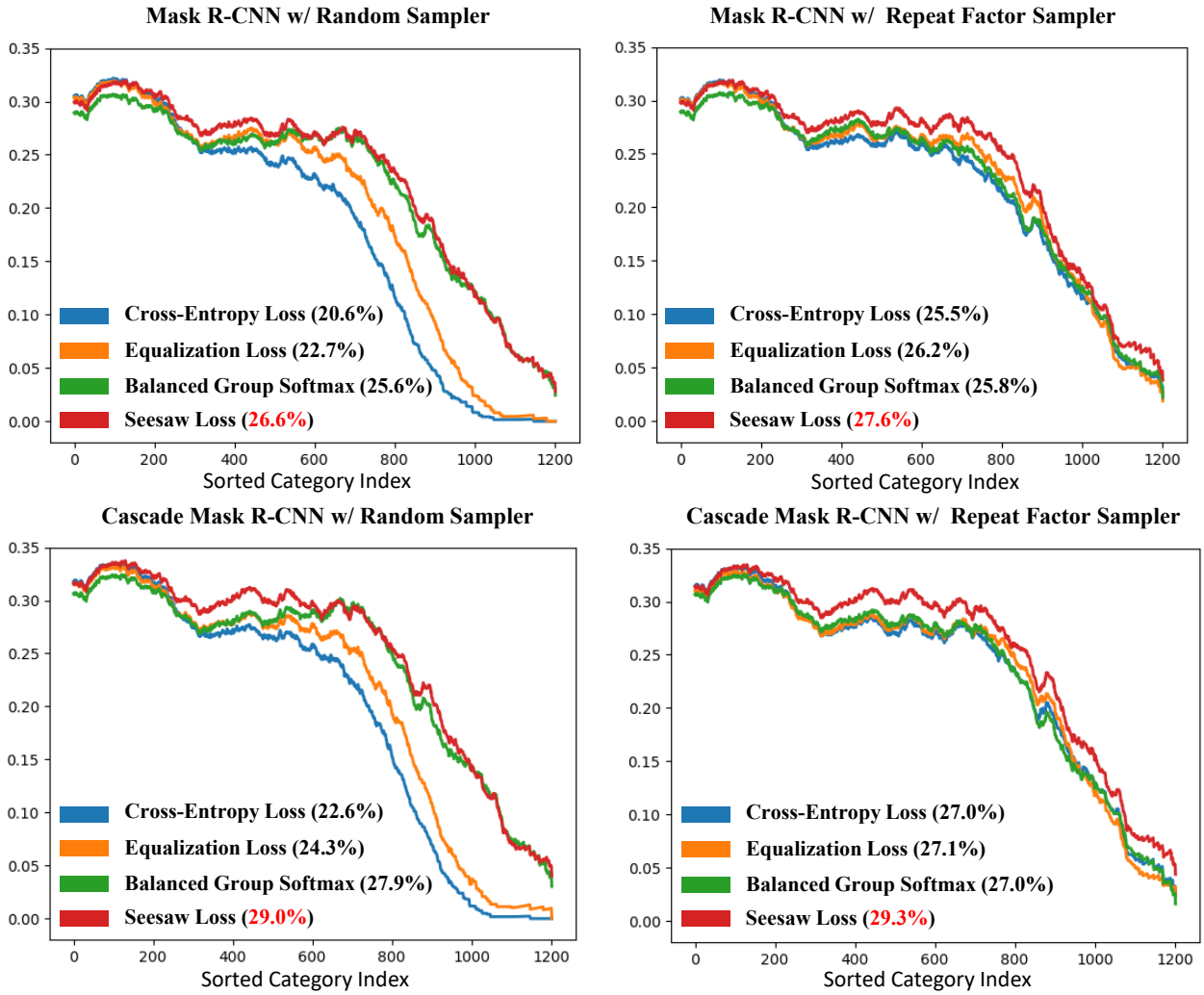

Figure A2: Per-category performance (AP) comparison between different methods in Table 1 of the main text. Norm Mask is not adopted for a fair comparison.

and negative samples during training. The overwhelming gradients of negative samples lead to a biased learning process for the classifier, which results in the low classification accuracy on tail classes. On the contrary, Seesaw Loss effectively re-balances the gradients of positive and negative samples across different categories. Consequently, Mask R-CNN with Seesaw Loss achieves significant improvements on instance segmentation performance as shown in Figure 1 and Table 1 in the main text.

3. Per-category Performance Comparison

In addition to the performance reported in Table 1 of the main text, we further show the per-category performance (AP) to verify the superiority of Seesaw Loss compared to other loss functions. As shown in Figure A2, compared to other loss functions (*i.e.*, Cross Entropy Loss, Equalization Loss [15], and Balanced Group Softmax [10]), Seesaw Loss consistently achieves strong performance across categories with different frequency on different frameworks (*i.e.* Mask R-CNN [7], Cascade Mask R-CNN [1]) and samplers (*i.e.*, random sampler, repeat factor sampler [6]).

4. LVIS Challenge 2020

Here we present the approach used in the entry of team **MMDet** in the LVIS Challenge 2020. In our entry, we adopt Seesaw Loss for long-tailed instance segmentation as described in the main text. Seesaw Loss improves the strong baseline by 6.9% AP on LVIS v1 *val* split. Furthermore, we propose HTC-Lite, a light-weight version of Hybrid Task Cascade (HTC) [3] which replaces the semantic segmentation branch with a global context encoder. With a single model and without using external data and annotations

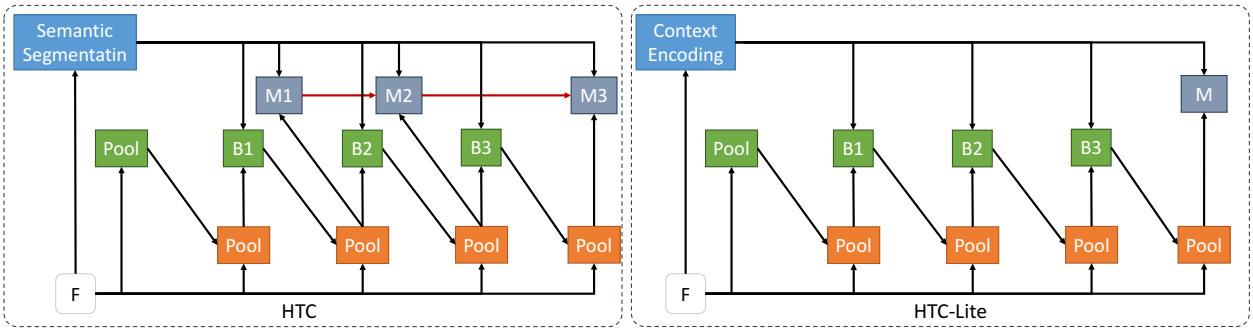

Figure A3: The comparison of HTC and HTC-Lite.

Table A1: Step by step results of our entry on LVIS v1 [6] *val* split.

Modification	Schedule	AP	AP_r	AP_c	AP_f	AP^{box}
Mask R-CNN	2x	18.7	1.0	16.1	29.4	20.1
+ SyncBN	2x	18.9 (+0.2)	0.7	16.0	30.3	20.2 (+0.1)
+ CARAFE Upsample	2x	19.4 (+0.5)	0.7	16.5	30.9	20.4 (+0.2)
+ HTC-Lite	2x	21.9 (+2.5)	1.1	19.8	33.5	23.6 (+3.2)
+ TSD	2x	23.5 (+1.6)	2.3	22.3	34.0	25.5 (+1.9)
+ Mask scoring	2x	23.9 (+0.4)	2.8	22.4	35.0	25.6 (+0.1)
+ Training-time augmentaion	45e	26.5 (+2.6)	3.6	25.7	37.4	28.1 (+2.5)
+ Stronger neck	45e	27.0 (+0.5)	3.5	25.8	38.6	29.1 (+1.0)
+ Stronger backbone	45e	29.9 (+2.9)	4.2	29.4	41.8	32.1 (+3.0)
+ **Seesaw Loss**	45e	36.8 (**+6.9**)	25.5	35.6	42.9	39.8 (**+7.7**)
+ Dual Head Classification	1x	37.3 (+0.5)	26.4	36.3	43.1	40.6 (+0.8)
+ Test-time augmentation	-	**38.8** (+1.5)	**26.4**	**38.3**	**44.9**	**41.5** (+0.9)

Table A2: Comparison of different cascading instance segmentation frameworks on LVIS v1 [6] dataset with repeat factor sampler and 1x training schedule. HTC w/o semantic indicates HTC without adopting the semantic segmentation branch since semantic segmentation annotations are not available on LVIS v1 dataset.

Method	AP	AP_r	AP_c	AP_f	AP^{box}	fps
Cascade Mask R-CNN [1]	24.3	13.7	23.8	29.6	27.2	0.1
HTC w/o semantic [3]	24.8	14.5	24.1	30.2	27.0	0.1
HTC-Lite	25.5	15.0	25.4	30.3	28.0	2.8

except for standard ImageNet-1k classification dataset for backbone pre-training, our entry achieves **38.92% AP** on the *test-dev* split of the LVIS v1 benchmark.

4.1. HTC-Lite

We propose HTC-Lite, a light-weight version of Hybrid Task Cascade (HTC) [3], to accelerate the training and inference speed while maintaining good performance. As shown in Figure A3, the modifications are in two folds: replacing the semantic segmentation branch with a global context encoding branch and reducing mask heads.

Context Encoding Branch. Since semantic segmentation annotations are not available for LVIS [6] dataset, we replace the semantic segmentation branch with a global context encoder [20] which works as a multi-label classification branch trained by a binary cross-entropy loss. The context encoder applies convolution layers and a global average pooling on the input feature map to obtain a feature vector. And an auxiliary fully connected (fc) layer is applied on the feature vector to predict the categories existing in the current image. By this approach, this feature vector encodes the global context information of the image. Then it is added to the RoI features used by box heads and mask heads to enrich their semantic information.

Reduced Mask Heads. To further reduce the cost of instance segmentation, HTC-Lite only keeps the mask head in the last stage, which also spares the original interleaved information passing.

In Table A2, we compare the performance and inference speed on LVIS v1 [6] dataset of HTC-Lite with two mainstream cascading instance segmentation frameworks, *i.e.*, Cascade Mask R-CNN and HTC. The ResNet-50 with FPN backbone, repeat factor sampler and 1x training schedule are adopted in these methods. The semantic segmentation branch in HTC [3] is removed since semantic segmentation annotations are not available on LVIS v1 [6] dataset. We evaluate the inference speed for each framework with a single Tesla V100 GPU. The experimental results show that HTC-Lite is not only much more efficient than its counterparts but also outperforms them.

4.2. Step by Step Results

We report the step-by-step results of our entry in LVIS Challenge 2020 as shown in Table A1.

Baseline. The baseline model is Mask R-CNN [7] using ResNet-50-FPN [11], trained with multi-scale training and random data sampler by 2x schedule [4].

SyncBN. We use SyncBN [12, 13] in the backbone and heads.

CARAFE Upsample. CARAFE [16] is used for upsampling in the mask head.

HTC-Lite. We use HTC-Lite as described in Section 4.1 in supplementary materials.

TSD. TSD [14] is used to replace the box heads in all three stages in HTC-Lite.

Mask Scoring. We further use the mask IoU head [9] to improve mask results.

Training Time Augmentation. We train the model with stronger augmentations with 45 epochs. The learning rate is decreased by 0.1 at 30 and 40 epochs. We randomly resize the image with its longer edge in a range of 768 to 1792 pixels. And then, we randomly crop the image to the size of 1280×1280 after adopting instaboost augmentation [5].

Stronger Neck. We replace the neck architecture with an enhanced version of Feature Pyramid Grids (FPG) [2]. The enhanced FPG uses deformable convolution v2 (DCNv2) [22] after feature upsampling, and a downsampler version of CARAFE [16, 17] for feature downsampling.

Stronger Backbone. We use ResNeSt-200 [21] with DCNv2 [22] as the backbone.

Seesaw Loss. We apply the proposed Seesaw Loss to classification branches of the TSD box head, in all cascading stages. Furthermore, we remove the original progressive constraint (PC) loss on classification branches in TSD.

Dual Head Classification. Inspired by [19, 18], we adopt a dual-head classification policy to further boost the performance. Specifically, after obtaining the model with Seesaw Loss trained by a random sampler, we freeze all components in the original model. Then we finetune a new classification branch for each cascading stage on the fixed model using repeat factor sampler [6] by 1x schedule. During inference, the classification scores of original classification branches and the scores of new classification branches are averaged to get the final scores.

Test Time Augmentation. We adopt multi-scale testing with horizontal flipping. Specifically, image scales are 1200, 1400, 1600, 1800, and 2000 pixels.

Final Performance on Test-dev. After adding the above-mentioned components step by step, we finally achieve **38.8% AP** on the *val* split and **38.92% AP** on the *test-dev* split.

References

[1] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: High quality object detection and instance segmentation. *arXiv preprint arXiv:1906.09756*, 2019. 2, 3

[2] Kai Chen, Yuhang Cao, Chen Change Loy, Dahua Lin, and Christoph Feichtenhofer. Feature pyramid grids. 2020. 4

[3] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. Hybrid task cascade for instance segmentation. In *CVPR*, 2019. 2, 3

[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. 4

[5] Hao-Shu Fang, Jianhua Sun, Runzhong Wang, Minghao Gou, Yong-Lu Li, and Cewu Lu. Instaboost: Boosting instance segmentation via probability map guided copy-pasting. In *ICCV*, 2019. 4

[6] Agrim Gupta, Piotr Dollar, and Ross Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *CVPR*, 2019. 1, 2, 3, 4

[7] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask R-CNN. *ICCV*, 2017. 1, 2, 4

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1

[9] Zhaojin Huang, Lichao Huang, Yongchao Gong, Chang Huang, and Xinggang Wang. Mask Scoring R-CNN. In *CVPR*, 2019. 4

[10] 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 *CVPR*, 2020. 2

[11] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 1, 4

[12] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *CVPR*, 2018. 4

[13] Chao Peng, Tete Xiao, Zeming Li, Yuning Jiang, Xiangyu Zhang, Kai Jia, Gang Yu, and Jian Sun. MegDet: A large mini-batch object detector. *CVPR*, 2018. 4

[14] Guanglu Song, Yu Liu, and Xiaogang Wang. Revisiting the sibling head in object detector. *CVPR*, 2020. 4

[15] Jingru Tan, Changbao Wang, Buyu Li, Quanquan Li, Wanli Ouyang, Changqing Yin, and Junjie Yan. Equalization loss for long-tailed object recognition. In *CVPR*, 2020. 2

[16] Jiaqi Wang, Kai Chen, Rui Xu, Ziwei Liu, Chen Change Loy, and Dahua Lin. CARAFE: Content-Aware ReAssembly of FEatures. In *ICCV*, 2019. 4

[17] Jiaqi Wang, Kai Chen, Rui Xu, Ziwei Liu, Chen Change Loy, and Dahua Lin. Carafe++: Unified content-aware reassembly of features, 2020. 4

[18] Tao Wang, Yu Li, Bingyi Kang, Junnan Li, Jun Hao Liew, Sheng Tang, Steven Hoi, and Jiashi Feng. Classification calibration for long-tail instance segmentation. *arXiv preprint arXiv:1910.13081*, 2019. 4

[19] Tao Wang, Yu Li, Bingyi Kang, Junnan Li, Jun Hao Liew, Sheng Tang, Steven C. H. Hoi, and Jiashi Feng. The devil is in classification: A simple framework for long-tail instance segmentation. In *ECCV*, 2020. 4

[20] Hang Zhang, Kristin Dana, Jianping Shi, Zhongyue Zhang, Xiaogang Wang, Ambrish Tyagi, and Amit Agrawal. Context encoding for semantic segmentation. In *CVPR*, 2018. 3

[21] Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Zhi Zhang, Haibin Lin, Yue Sun, Tong He, Jonas Muller, R. Manmatha, Mu Li, and Alexander Smola. ResNeSt: Split-attention networks. *arXiv preprint arXiv:2004.08955*, 2020. 4

[22] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable ConvNets V2: More deformable, better results. In *CVPR*, 2019. 4