Reconciling Object-level and Global-level Objectives for Long-Tail Detection Supplementary Material

Shaoyu Zhang[1,2], Chen Chen[1,2*], Silong Peng[1,2,3*]
[1]Institute of Automation, Chinese Academy of Sciences, Beijing, China
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China
[3]Beijing Visystem Co. Ltd, Beijing, China
{zhangshaoyu2019, chen.chen, silong.peng}@ia.ac.cn

A. Deduction of GAP Loss and Optimization

In this section, we present the detailed deduction of the generalized average precision (GAP) loss and its optimization.

A.1. GAP Loss

As the global-level ranking objective, GAP loss aims to rectify ranking errors by maximizing AP over all categories. In a mini-batch $\hat{\mathcal{B}}$, it could be represented as:

$$L = 1 - AP_{\hat{\mathcal{B}}} = \frac{1}{|\hat{C}|} \sum_{m \in \hat{C}} \frac{1}{|\hat{\mathcal{P}}(m)|} \sum_{i \in \hat{\mathcal{P}}(m)} \sum_{j \in \hat{\mathcal{N}}(m)} \frac{H(i,j)}{R(i)}, \tag{1}$$

where $\hat{C}$ is the set of categories present in $\hat{\mathcal{B}}$, and $\hat{\mathcal{P}}(\cdot)$ is the set of positive samples from each specific category in $\hat{\mathcal{B}}$. By definition, $\hat{\mathcal{N}}(m)$ includes not only the negative set $\hat{\mathcal{N}}$, but also the positive set of other categories, *i.e.*, $\hat{\mathcal{N}}(m) = \hat{\mathcal{N}} \cup \hat{\mathcal{P}}(n), \forall n \neq m$. This causes conflicts in optimization of different categories and confuses the learning objective. Therefore, when optimizing ranking orders for category m, we exclude the positive samples of other categories from $\hat{\mathcal{N}}(m)$ and only use $\hat{\mathcal{N}}$ as negative sample set. In addition, as analyzed in Sec. 3.3, the $|\hat{\mathcal{P}}(m)|$ in each batch could not truly reflect the category statistic of the entire dataset, which still leads to significantly larger gradients for frequent categories than rare categories during training. Therefore, the global statistic ω is introduced to approximatively maximize AP over the entire dataset. According to these modifications, GAP loss is rewritten as:

$$L_{gap} = \frac{1}{|\hat{C}|} \sum_{m \in \hat{C}} \frac{\omega_m}{|\hat{\mathcal{P}}(m)|} \sum_{i \in \hat{\mathcal{P}}(m)} \sum_{j \in \hat{\mathcal{N}}} L_{ij}, \tag{2}$$

where $L_{ij} = H(i,j)/R(i)$ is the pairwise ranking loss between sample i and j.

A.2. Optimization of GAP Loss

Since L_{ij} is non-differentiable, the popular gradient descent method is not applicable to the optimization of GAP loss. Assuming $x_{ij} = s_j - s_i$, the key to the problem is to calculate $\frac{\partial L_{ij}}{\partial x_{ij}}$. Similar to [1], we adopt the error-driven update [3] method to obtain the gradients. Given the input x_{ij}, the update is simply derived from the difference between target loss value and current loss value

$$\Delta x_{ij} = L^*_{ij} - L_{ij}, \tag{3}$$

where $L^*_{ij} = 0$ in the case of proper ranking between i and j. Thus, the derivative w.r.t. s_i and s_j is:

$$\frac{\partial L_{ij}}{\partial s_i} = -\Delta x_{ij} \frac{\partial x_{ij}}{\partial s_i} = -L_{ij}, \tag{4}$$

$$\frac{\partial L_{ij}}{\partial s_j} = -\Delta x_{ij} \frac{\partial x_{ij}}{\partial s_j} = L_{ij}. \tag{5}$$

Given a score s_k, the gradient propagated from L_{gap} is

$$g^k_{gap} = \frac{1}{|\hat{\mathcal{C}}|} \sum_{m \in \hat{C}} \frac{\omega_m}{|\hat{\mathcal{P}}(m)|} \sum_{i \in \hat{\mathcal{P}}(m)} \sum_{j \in \hat{\mathcal{N}}} \frac{\partial L_{ij}}{\partial s_k} \tag{6}$$

$$= \begin{cases} -\frac{1}{|\hat{\mathcal{C}}|} \frac{\omega_m}{|\hat{\mathcal{P}}(m)|} \sum_{j \in \hat{\mathcal{N}}} L_{kj}, & k \in \hat{\mathcal{P}}(m) \\ \frac{1}{|\hat{\mathcal{C}}|} \sum_{m \in \hat{C}} \frac{\omega_m}{|\hat{\mathcal{P}}(m)|} \sum_{i \in \hat{\mathcal{P}}(m)} L_{ik}, & k \in \hat{\mathcal{N}}. \end{cases} \tag{7}$$

This manually set gradient is further back propagated by the chain rule to update parameters of the whole network.

B. More Ablation Studies

In addition to the ablation studies in Sec. 4.2, here we conduct more comprehensive studies on our ROG. The experiments are conducted on the Mask R-CNN with ResNet-50-FPN. All the models are trained with 1x schedule. Unless specified, Sigmoid CE loss is adopted as the classification loss in ROG.

Table A1. Analysis on different values of hyper-parameters γ and λ_{gap}.

Sampler	γ	λ_{gap}	AP^{bbox}	AP	AP_r	AP_c	AP_f
Random	1.0	1.0	**24.7**	**23.9**	**14.1**	**23.7**	28.0
		0.1	20.5	19.6	2.8	18.2	**28.7**
RFS	0.5	1.0	23.8	23.5	13.4	23.5	27.9
		0.5	24.7	24.2	14.0	24.1	28.9
		0.1	**26.5**	**25.8**	**19.2**	**25.3**	**29.3**
	1.0	1.0	23.9	23.2	12.5	23.4	27.7
		0.5	25.5	24.9	15.5	24.9	29.0
		0.1	25.9	25.1	18.2	24.6	28.7

Table A2. Detailed ablation study on the designs in GAP loss. *Cat. Spec.* means the category-specific manner in defining positive and negative samples. The global statistic ω is defined in Eq. 9. *Online* means the online statistic manner for calculating category size. $Excl.FG$ means excluding other foreground positive samples from the negative set of category m.

Cat. Spec.	ω	Online	Excl. FG	AP^{bbox}	AP	AP_r	AP_c	AP_f
✗	✗	✗	✗	14.6	15.8	2.9	13.8	23.7
✓	✗	✗	✗	19.2	19.6	7.5	19.2	25.3
✓	✓	✗	✗	23.8	23.5	14.1	23.7	28.0
✓	✓	✓	✗	24.1	23.4	**14.8**	22.5	28.1
✓	✓	✗	✓	24.1	**23.9**	14.5	**23.8**	28.2
✓	✓	✓	✓	**24.7**	**23.9**	**14.8**	23.0	**28.8**

B.1. Hyper-parameters

We study the two main parameters in ROG, *i.e.*, γ for calculating ω and λ_{gap} for controlling the weight of GAP loss. The results are reported in Table A1. For the models trained without RFS, λ_{gap} has an important influence on the performance. Compared with $\lambda_{gap} = 0.1$, a value of 1.0 boots the weight of GAP loss and improves the AP^{bbox} and AP by 4.2 and 4.3, respectively. It also highlights the significance of GAP loss for adjusting ranking orders across objects. Interestingly, we find that the models trained with RFS prefer a smaller λ_{gap}, possibly because RFS mediately helps to improve the scores of rare objects and leads to less ranking errors, to a certain extent. We further study the effect of γ. Intuitively, a larger value of γ brings stronger global statistic and vice versa, *e.g.*, $\gamma = 0$ means no global statistic is injected into GAP loss. Experimentally, $\gamma = 0.5$ and $\lambda_{gap} = 0.1$ leads to better results with the RFS sampler.

B.2. Designs in GAP Loss

Regarding the designs in GAP loss, compared with Table 3 of the main text, we additionally make the comparisons between whether or not to define positive-negative sample pairs in a category-specific manner, and also report AP on each category group.

In Table A2, the first line is the result from category-agnostic AP loss, *i.e.*, the loss is averaged on all positive samples regardless of the categories which they belong to. Since the majority of positive samples belongs to frequent categories, the gradients for optimizing ranking of rare categories are negligible. This leads to poor performance on AP_r. By defining positive set $\hat{\mathcal{P}}(m)$ and negative set $\hat{\mathcal{N}}(m)$ w.r.t. each specific category m in a batch, AP_r is improved from 2.9 to 7.5. However, considering the entire dataset, the gradients are still unbalanced, as rare categories appear only in a few batches and the statistic in a batch fails to represent the global distribution of entire dataset. Therefore, we introduce the statistical information of entire dataset via the global indicator ω. This significantly improves AP_r by 6.6 points. As analyzed in Sec. 4.2, the results also prove the effectiveness of online statistic for total category size and excluding positive samples of other foreground categories when optimizing for category m. The overall design of our GAP loss leads to superior and more balanced performance on the rare, common and frequent categories.

B.3. Weight Norms of Classifier

To investigate whether ROG balances the training across categories, we visualize the weight norms of classifier under various settings. For the classification loss in ROG, we choose the state-of-the-art ECM loss [2] and Seesaw loss [4]. Following their implementations, the normalized mask prediction (Norm Mask) is adopted with the RFS sampler. As shown in Figure A1, compared with ECM and Seesaw, the weight norms are more balanced under our ROG framework. This phenomenon is consistent across different losses (*e.g.*, ECM and Seesaw), samplers (*e.g.*, random and RFS) and other tricks (*e.g.*, Norm Mask), validating the effectiveness of ROG for balancing the classifier during training.

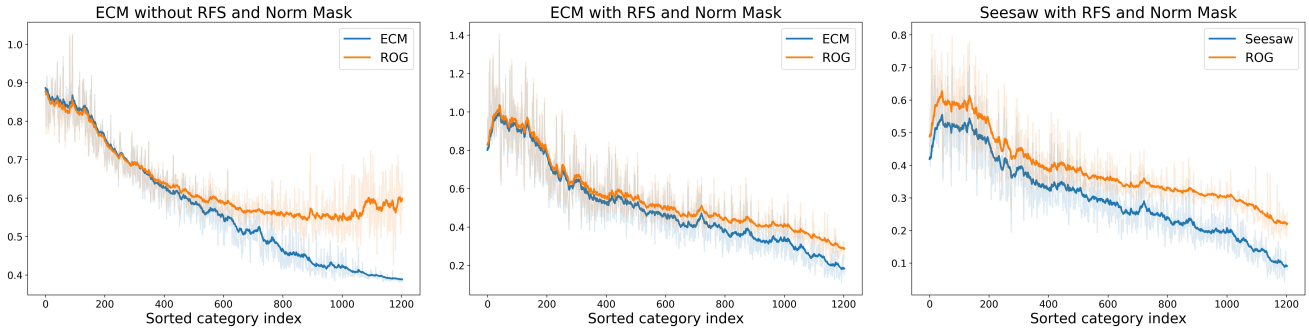

Figure A1. Visualization on weight norms of classifier.

References

[1] Kean Chen, Weiyao Lin, Jianguo Li, John See, Ji Wang, and Junni Zou. Ap-loss for accurate one-stage object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(11):3782–3798, 2020.

[2] Jang Hyun Cho and Philipp Krähenbühl. Long-tail detection with effective class-margins. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VIII*, pages 698–714. Springer, 2022.

[3] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[4] 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*, pages 9695–9704, 2021.