Long-tail Detection with Effective Class-Margins

Jang Hyun Cho[1] and Philipp Krähenbühl[1]

The University of Texas at Austin, Austin TX 78712, USA
janghyuncho7@utexas.edu, philkr@cs.utexas.edu
https://github.com/janghyuncho/ECM-Loss

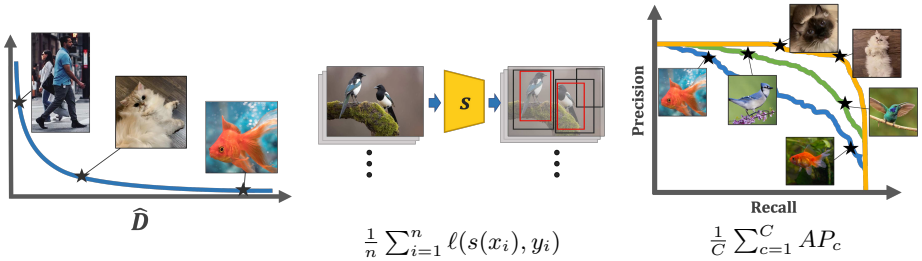
Abstract. Large-scale object detection and instance segmentation face a severe data imbalance. The finer-grained object classes become, the less frequent they appear in our datasets. However, at test-time, we expect a detector that performs well for all classes and not just the most frequent ones. In this paper, we provide a theoretical understanding of the long-trail detection problem. We show how the commonly used mean average precision evaluation metric on an unknown test set is bound by a margin-based binary classification error on a long-tailed object detection training set. We optimize margin-based binary classification error with a novel surrogate objective called **Effective Class-Margin Loss** (ECM). The ECM loss is simple, theoretically well-motivated, and outperforms other heuristic counterparts on LVIS v1 benchmark over a wide range of architecture and detectors. Code is available at `https://github.com/janghyuncho/ECM-Loss`.

Keywords: object detection, long-tail object detection, long-tail instance segmentation, margin bound, loss function

1 Introduction

The state-of-the-art performance of common object detectors has more than tripled over the past 5 years. However, much of this progress is measured on just 80 common object categories [27]. These categories cover only a small portion of our visual experiences. They are nicely balanced and hide much of the complexities of large-scale object detection. In a natural setting, objects follow a long-tail distribution, and artificially balancing them is hard [16]. Many recent large-scale detection approaches instead balance the training loss [39, 44, 51] or its gradient [24, 40] to emulate a balanced training setup. Despite the steady progress over the past few years, these methods largely rely on heuristics or experimental discovery. Consequently, they are often based on intuition, require extensive hyper-parameter tuning, and include a large bag-of-tricks.

In this paper, we take a statistical approach to the problem. The core issue in long-tail recognition is that training and evaluation metrics do not line up, see Figure 1. At test time, we expect the detector to do well on all classes, not just a select few. This is reflected in the common evaluation metric: mean-average-precision (mAP) [12,16,22,27,38]. At training time, we ideally learn from



(a) Data distribution (b) Training objective (c) Evaluation objective

Fig. 1: In long-trail detection, training objectives (b) do not align with evaluation objectives (c). During training, we optimize an empirical objective on a long-tail data distribution (a). However at test time, we expect a detector that performs well on all classes. In this paper, we connect the detection objective (c) on an unknown test set to an empirical training objective (b) on a long-tail real-world data distribution (a) through the margin-bound theory [1, 4, 19, 21].

all available data using a cross-entropy [5, 17, 36] or related loss [23, 26, 33, 49]. Here, we draw a theoretical connection between the balanced evaluation metric, mAP, and margin-based binary classification. We show that mAP is bound from above and below by a pairwise ranking error, which in turn reduces to binary classification. We address the class imbalance through the theory of margin-bounds [1, 4, 19, 21], and reduce detector training to a margin-based binary classification problem.

Putting it all together, margin-based binary classification provides a closed-form solution for the ideal margin for each object category. This margin depends only on the number of positive and negative annotations for each object category. At training time, we relax the margin-based binary classification problem to binary cross entropy on a surrogate objective. We call this surrogate loss **Effective Class-Margin Loss** (ECM). This relaxation converts margins into weights on the loss function. Our ECM loss is completely *hyperparameter-free* and applicable to a large number of detectors and backbones.

We evaluate the ECM loss on LVIS v1 and OpenImages. It outperforms state-of-the-art large-scale detection approaches across various frameworks and backbones. The ECM loss naturally extends to one-stage detectors [42, 49, 50, 54].

2 Related works

Object detection is largely divided into one-stage and two-stage pipelines. In one-stage object detection [26, 33–35, 54], classification and localization are simultaneously predicted densely on each coordinate of feature map representation. Hence, one-stage detection faces extreme foreground-background imbalance aside from cross-category imbalance. These issues are addressed either

by carefully engineered loss function such as the Focal Loss [26, 54], or sampling heuristics like ATSS [50]. Two-stage object detection [3, 15, 36] mitigates the foreground-background imbalance using a category-agnostic classifier in the first-stage. However, neither type of detection pipelines handles cross-category imbalance. We show that our ECM loss trains well with both types of detectors.

Long-tail detection. Learning under severely long-tailed distribution is challenging. There are two broad categories of approaches: data-based and loss-based. Data-based approaches include external datasets [48], extensive data augmentation with larger backbones [14], or optimized data-sampling strategies [16, 45, 46]. Loss-based approaches [24, 39–41, 44, 51, 52] modify or re-weights the classification loss used to train detectors. They perform this re-weighting either *implicitly* or *explicitly*. The Equalization Loss [40] ignores the negative gradient for rare classes. It builds on the intuition that rare classes are "discouraged" by all the negative gradients of other classes (and background samples). Balanced Group Softmax (BaGS) [24] divides classes into several groups according to their frequency in the training set. BaGS then applies a cross-entropy with softmax only within each group. This implicitly controls the negative gradient to the rare classes from frequent classes and backgrounds. The federated loss [52] only provides negative gradients to classes that appear in an image. This implicitly reduces the impact of the negative gradient to the rare classes. The Equalization Loss v2 [39] directly balances the ratio of cumulative positive and negative gradients per class. The Seesaw Loss [44] similarly uses the class frequency to directly reduce the weight of negative gradients for rare classes. In addition, it compensates for the diminished gradient from misclassifications by scaling up by the ratio of the predicted probability of a class and that of the ground truth class. These methods share the common premise that overwhelming negative gradients will influence the training dynamics of the detector and result in a biased classifier. While this makes intuitive sense, there is little analytical or theoretical justification for particular re-weighting schemes. This paper provides a theoretical link between commonly used mean average precision on a test set and a **weighted binary cross entropy** loss on an unbalanced training set. We provide an optimal weighting scheme that bounds the expected test mAP.

Learning with class-margins. Margin-based learning has been widely used in face recognition [11, 28, 43] and classification under imbalanced data [4]. In fact, assigning proper margins has a long history in bounds to generalization error [1, 2, 19, 21]. Cao et al. [4] showed the effectiveness of analytically derived margins in imbalanced classification. In *separable* two-class setting (i.e., training error can converge to 0), closed form class-margins follow from a simple constrained optimization. Many recent heuristics in long-tail detection use this setting as the basis of re-weighted losses [24, 40, 44]. We take a slightly different approach. We show that the margin-based classification theory applies to detection by first establishing a connection between mean average precision (mAP) and a pairwise ranking error. This ranking error is bound from above by a margin-based classification loss. This theoretical connection then provides us with a set

of weights for a surrogate loss on a *training set* that optimizes the mAP on an *unknown test set*.

Optimizing average precision. Several works optimize the average precision metric directly. Rolínek et al. [37] address non-differentiability and introduced black-box differentiation. Chen et al. [7] propose an AP Loss which applies an error-driven update mechanism for the non-differentiable part of the computation graph. Oksuz et al. [31] took a similar approach to optimize for Localization-Recall-Precision (LRP) [30]. In contrast, we reduce average precision to ranking and then margin-based binary classification, which allows us to use tools from learning theory to bound and minimize the generalization error of our detector.

3 Preliminary

Test Error Bound. Classical learning theory bounds the test error in terms of training error and some complexity measures. Much work builds on the Lipschitz bound of Bartlett and Mendelson [2]. For any Lipschitz continuous loss function ℓ, it relates the expected error $\mathcal{L}(f) = E[\ell(f(x), y)]$ and empirical error $\hat{\mathcal{L}}(f) = \frac{1}{n}\sum_{i=1}^{n} \ell(f(x_i), y_i)$ over a dataset of size n with inputs x_i and labels y_i. In detection, we commonly refer to the expected error as a test error over an unknown test set or distribution, and empirical error as a training error. Training and testing data usually follow the same underlying distribution, but different samples.

Theorem 1 (Class-margin bound [4, 19]). *Suppose $\ell(x) = 1_{[x<0]}$ a zero-one error and $\ell_\gamma = 1_{[x<\gamma]}$ a margin error with a non-negative margin γ. Similarly, $\mathcal{L}_y(f) = E[\ell(f(x)_y)]$ and $\mathcal{L}_{\gamma,y}(f) = E[\ell_\gamma(f(x)_y)]$. Then, for each class-conditional data distribution $P(X|Y = c)$, for all $f \in \mathcal{F}$ and class-margin $\gamma_c > 0$, with probability $1 - \delta_c$, the class-conditional test error for class $c \in C$ can be bounded from above as following:*

$$\mathcal{L}_c(f) \leq \hat{\mathcal{L}}_{\gamma_c,c}(f) + \frac{4}{\gamma_c}\mathcal{R}_n(\mathcal{F}) + \sqrt{\frac{\log(2/\delta_c)}{n_c}} + \epsilon(n_c, \gamma_c, \delta_c)$$

where $\mathcal{R}_n(\mathcal{F})$ is the Rademacher complexity of a function class $\mathcal{F}$ which is typically bounded by $\sqrt{\frac{C(\mathcal{F})}{n}}$ for some complexity measure of C [1, 4, 19].

Kakade et al. [19] prove the above theorem for binary classification, and Cao et al. [4] extend it to multi-class classification under a long-tail. Their proof follows Lipschitz bounds of Bartlett and Mendelson [2]. Theorem 1 will be our main tool to bound the generalization error of a detector.

Detection Metrics. Object detection measures the performance of a model through average precision along with different recall values. Let $\mathrm{TP}(t)$, $\mathrm{FP}(t)$, and $\mathrm{FN}(t)$ be the true positive, false positive, and false negative detections for a score threshold t. Let $\mathrm{Pc}(t)$ be the precision and $\mathrm{Rc}(t)$ be the recall. Average

precision AP then integrates precision over equally spaced recall thresholds $t \in T$:

$$\mathrm{Pc}(t) = \frac{\mathrm{TP}(t)}{\mathrm{TP}(t) + \mathrm{FP}(t)} \quad \mathrm{Rc}(t) = \frac{\mathrm{TP}(t)}{\mathrm{TP}(t) + \mathrm{FN}(t)} \quad \mathrm{AP} = \frac{1}{|T|} \sum_{t \in T} \mathrm{Pc}(t). \quad (1)$$

Generally, true positives, false positives, and false negatives follow an assignment procedure that enumerates all annotated objects. If a ground truth object has a close-by prediction with a score $s > t$, it counts as a positive. Here closeness is measured by overlap. If there is no close-by prediction, it is a false negative. Any remaining predictions with score $s > t$ count towards false positives. All above metrics are defined on finite sets and do not directly extend general distributions.

In this paper, we base our derivations on a probabilistic version of average precision. For every class $c \in C$, let D_c be the distribution of positive samples, and $D_{\neg c}$ be the distribution of negative samples. These positives and negatives may use an overlap metric to ground truth annotations. Let $P(c)$ and $P(\neg c)$ be the prior probabilities on labels of class c or not c. $P(c)$ is proportional to the number of annotated examples of class c at training time. Let $s_c(x) \in [0, 1]$ be the score of a detector for input x. The probability of a detector s_c to produce a true positive of class c with threshold t is $tp_c(t) = P(c)P_{x \sim D_c}(s_c(x) > t)$, false positive $fp_c(t) = P(\neg c)P_{x \sim D_{\neg c}}(s_c(x) > t)$, and false negative $fn_c(t) = P(c)P_{x \sim D_c}(s_c(x) \leq t)$. This leads to a probabilistic recall and precision

$$r_c(t) = \frac{tp_c(t)}{tp_c(t) + fn_c(t)} = \frac{tp_c(t)}{P(c)} = P_{x \sim D_c}(s_c(x) > t), \quad (2)$$

$$p_c(t) = \frac{tp_c(t)}{tp_c(t) + fp_c(t)} = \frac{r_c(t)}{r_c(t) + \alpha_c P_{x \sim D_{\neg c}}(s_c(x) > t)}. \quad (3)$$

Here, $\alpha_c = \frac{P(\neg c)}{P(c)}$ corrects for the different frequency of foreground and background samples for class c. By definition $1 - r_c(t)$ is a cumulative distribution function $r_c(1) = 0$, $r_c(0) = 1$ and $r_c(t) \geq r_c(t + \delta)$ for $\delta > 0$. Without loss of generality, we assume that the recall is strictly monotonous $r_c(t) > r_c(t + \delta)$[1]. For a strictly monotonous recall $r_c(t)$, the quantile function is the inverse $r_c^{-1}(\beta)$. Average precision then integrates over these quantiles.

Definition 1 (Probabilistic average precision).

$$ap_c = \int_0^1 p_c(r_c^{-1}(\beta))d\beta = \int_0^1 \frac{\beta}{\beta + \alpha P_{x \sim D_{\neg c}}(s_c(x) > r_c^{-1}(\beta))} d\beta$$

There are two core differences between regular AP and probabilistic AP: 1) The probabilistic formulation scores a nearly exhaustive list of candidate objects, similar to one-stage detectors or the second stage of two-stage detectors.

[1] For any detector s_c with a non-strict monotonous recall, there is a nearly identical detector s'_c with strictly monotonous recall: $s'_c(x) = s_c(x)$ with chance $1 - \varepsilon$ and uniform at random $s'_c(x) \in U[0, 1]$ with chance ε for any small value $\varepsilon > 0$.

It does not consider bounding box regression. 2) Regular AP penalizes duplicate detections as false positives, probabilistic AP does not. This means that at training time, positives and negatives are strictly defined for probabilistic AP, which makes a proper analysis possible. At test time, non-maxima-suppression removes most duplicate detections without major issues.

In the next section, we show how this probabilistic AP relates to a pairwise ranking error on detections.

Definition 2 (Pairwise Ranking Error).

$$R_c = P_{x,x' \sim D_c \times D_{\neg c}}\big(s_c(x) < s_c(x')\big) = E_{x' \sim D_{\neg c}}\left[1 - r_c(s_c(x'))\right]$$

The pairwise ranking error measures how frequently negative samples x' rank above positives x. The second equality is derived in supplement.

While it is possible to optimize the ranking error empirically, it is hard to bound the empirical error. We instead bound R_c by a margin-based 0-1 classification problem and use Theorem 1.

4 Effective Class-Margins

We aim to train an object detector that performs well for all object classes. This is best expressed by maximizing mean average precision over all classes equally: $mAP = \frac{1}{|C|}\sum_{c \in C} ap_c$. Equivalently, we aim to minimize the detection error

$$\mathcal{L}^{\mathrm{Det}} = 1 - mAP = \frac{1}{|C|}\sum_{c \in C} \underbrace{(1 - ap_c)}_{\mathcal{L}_c^{\mathrm{Det}}} \tag{4}$$

Optimizing the detection error or mAP directly is hard [7, 31, 32, 37]. First, ap_c involves a computation over the entire distribution of detections and does not easily factorize over individual samples. Second, our goal is to optimize the expected detection error. However, at training time, we only have access to an empirical estimate over our training set $\hat{D}$.

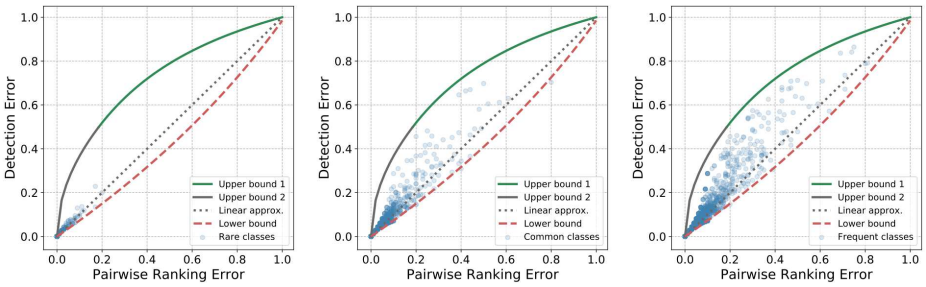
Despite these complexities, it is possible to optimize the expected detection error. The core idea follows a series of bounds for each training class c:

$$\mathcal{L}_c^{\mathrm{Det}} \lesssim m_c R_c \lesssim m_c \hat{\mathcal{L}}_{\gamma_c^{\pm},c},$$

where $\lesssim$ refers to inequalities up to a constant. In Section 4.1, we bound the detection error $\mathcal{L}_c^{\mathrm{Det}}$ by a weighted version of the ranking error R_c. In Section 4.2, we directly optimize an empirical upper bound $\hat{\mathcal{L}}_{\gamma_c^{\pm},c}$ to the weighted ranking error using class-margin-bounds in Theorem 1. Finally, in Section 4.3 we present a differentiable loss function to optimize the class-margin-bound.

4.1 Detection Error Bound

There is a strong correlation between the detection error $\mathcal{L}_c^{\mathrm{Det}}$ and the ranking objective R_c. For example, a perfect detector, that scores all positives above



(a) Rare classes (b) Common classes (c) Frequent classes

Fig. 2: Visualization of the upper and lower bound of Detection Error with respect to Pairwise Ranking Error. The solid and dotted lines are the theoretical bounds discussed in Theorem 2. We show that **actual detection errors** strictly follow the derived bounds. We evaluate multiple checkpoints of the same detector on rare, common, and frequent classes of lvis v1. Each point represents a checkpoint's performance on one class. We compute AP and ranking errors over the training set which has low errors especially for rare classes. In practice the upper and lower bounds are tight and the linear approximation fits well.

negatives, achieves both a ranking and detection error of zero. A detector that scores all negatives higher than positives has a ranking error of 1 and a detection error close to 1. For other error values the the ranking error R_c bounds the ap_c from both above and below, as shown in Figure 2 and Theorem 2.

Theorem 2 (AP - Pairwise Ranking Bound). *For a class c with negative-to-positive ratio $\alpha_c = \frac{P(\neg c)}{P(c)}$, the ranking error R_c bounds the probabilistic average precision ap_c from above and below:*

$$\alpha_c \log\left(\frac{1+\alpha_c}{1+\alpha_c - R_c}\right) \leq \mathcal{L}_c^{\text{Det}} \leq \min\left(\sqrt{\frac{2}{3}\alpha_c R_c}, 1 - \frac{8}{9}\frac{1}{1+2\alpha_c R_c}\right).$$

We provide a full proof in supplement and sketch out the proof strategy here. We derive both bounds using a constrained variational problem. For any detector s_c, data distributions D_c and $D_{\neg c}$, the average precision has the form

$$ap_c = \int_0^1 \frac{\beta}{\beta + \alpha g(\beta)} d\beta, \tag{5}$$

where $g(\beta) = P_{x' \sim D_{\neg c}}\left(s_c(x') > r_c^{-1}(\beta)\right) = P_{x' \sim D_{\neg c}}\left(r_c(s_c(x')) < \beta\right)$, since the recall is a strictly monotonously decreasing function. At the same time the ranking loss reduces to

$$R_c = \int_0^1 g(\beta) d\beta = E_{x' \sim D_{\neg c}}\left[1 - r_c(s_c(x'))\right]. \tag{6}$$

For a fixed ranking error $R_c = \kappa$, we find a function $0 \leq g(\beta) \leq 1$ that minimizes or maximizes the detection error $\mathcal{L}_c^{\text{Det}}$ through variational optimization. See supplement for more details. See Figure 2 for a visualization of the bounds.

Theorem 2 clearly establishes the connection between the ranking and detection. Unfortunately, the exact upper bound is hard to further simplify. We instead chose a linear approximation $\mathcal{L}_c^{\text{Det}} \approx m_c R_c$, for $\alpha_c \log\left(\frac{1+\alpha_c}{\alpha_c}\right) \leq m_c \leq \frac{\frac{1}{9}+2\alpha_c}{1+2\alpha_c}$. Figure 2 visualizes this linear approximation. The linear approximation even bounds the detection error from above $\mathcal{L}_c^{\text{Det}} \leq m_c R_c + o$ with an appropriate offset o. We denote this as $\mathcal{L}_c^{\text{Det}} \lesssim m_c R_c$.

In the next section, we show how this ranking loss is bound from above with a margin-based classification problem, which we minimize in 4.3.

4.2 Ranking bounds

To connect the ranking loss to the generalization error, we first reduce ranking to binary classification.

Theorem 3 (Binary error bound). *The ranking loss is bound from above by*

$$R_c \leq P_{x\sim D_c}\big(s_c(x) \leq t\big) + P_{x\sim D_{\neg c}}\big(t < s_c(x)\big),$$

for an arbitrary threshold t.

Proof. For any indicator $1_{[a<b]} \leq 1_{[a<t]} + 1_{[t\leq b]}$. Let's first rewrite ranking as expectations over indicator functions:

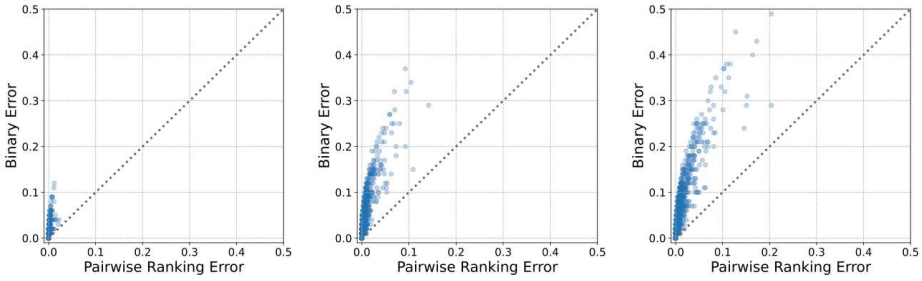
$$\begin{aligned} R_c &= E_{x\sim D_c}\left[E_{x'\sim D_{\neg c}}\left[1_{[s_c(x)<s_c(x')]}\right]\right] \\ &\leq E_{x\sim D_c}\left[E_{x'\sim D_{\neg c}}\left[1_{[s_c(x)\leq t]} + 1_{[t<s_c(x')]}\right]\right] \\ &= E_{x\sim D_c}\left[1_{[s_c(x)\leq t]}\right] + E_{x'\sim D_{\neg c}}\left[1_{[t<s_c(x')]}\right]. \end{aligned}$$

The last line uses the linearity of expectation. □

Figure 3 visualizes this upper bound. While any threshold t leads to an upper bound to ranking. We would like to optimize for the tightest upper bound t. We do this by folding t into the optimization. In a deep network, this simply means optimizing for a bias term of the detector score $s_c(x)$. For the remainder of the exposition, we assume t is part of s_c and use a detection threshold of $\frac{1}{2}$.

Next, lets us use Theorem 1 to bound the classification error, and thus the detection objective, by an empirical bound

$$\mathcal{L}_c^{\text{Det}} \lesssim m_c\left(\hat{\mathcal{L}}_{\gamma_c^+,c} + \hat{\mathcal{L}}_{\gamma_c^-,\neg c} + \frac{2}{\gamma_c^+}\sqrt{\frac{C(\mathcal{F})}{n_c}} + \frac{2}{\gamma_c^-}\sqrt{\frac{C(\mathcal{F})}{n_{\neg c}}} + \epsilon(n_c) + \epsilon(n_{\neg c})\right), \quad (7)$$



(a) Rare classes (b) Common classes (c) Frequent classes

Fig. 3: Visualization of the upper bound of Pairwise Ranking Error with respect to a binary classification error under an optimal threshold t. The blue dots correspond to actual binary errors of a detector. We evaluate multiple checkpoints of the same detector on rare, common, and frequent classes of lvis v1. Each point represents a checkpoint's performance in one class. We compute classification and ranking errors over the training set which has low errors, especially for rare classes.

where ϵ is a small constant that depends on the number of training samples n_c and $n_{\neg c}$. Here, we use empirical foreground $\hat{\mathcal{L}}_{\gamma_c^+,c} = \frac{1}{n}\sum_{i=1}^{n} 1_{[y_i=c]} 1_{[s_c(\hat{x}_i)\leq\gamma_c^+]}$ and background $\hat{\mathcal{L}}_{\gamma_c^-,\neg c} = \frac{1}{n}\sum_{i=1}^{n} 1_{[y_i\neq c]} 1_{[s_{\neg c}(\hat{x}_i)\leq\gamma_c^-]}$ classification errors for detector s_c. γ_c^+ and γ_c^- are positive and negative margins respectively.

Under a separability assumption, the tightest margins take the form

$$\gamma_c^+ = \frac{n_{\neg c}^{1/4}}{n_c^{1/4} + n_{\neg c}^{1/4}} \qquad \gamma_c^- = \frac{n_c^{1/4}}{n_c^{1/4} + n_{\neg c}^{1/4}}. \tag{8}$$

See Cao et al. [4] or the supplement for a derivation of these margins.

We have now arrived at an upper bound of the detection error $\mathcal{L}^{\text{Det}} = \frac{1}{|C|}\sum_{c\in C}\mathcal{L}_c^{\text{Det}}$ using an empirical margin-based classifier for each class c. This margin-based objective takes the generalization error and any potential class imbalance into account.

In the next section, we derive a continuous loss function for this binary objective and optimize it in a deep-network-based object detection system. Note that standard detector training is already classification-based, and our objective only introduces a margin and weight for each class.

4.3 Effective Class-Margin Loss

Our goal is to minimize the empirical margin-based error

$$\hat{\mathcal{L}}_{\gamma_c^\pm,c} = \hat{\mathcal{L}}_{\gamma_c^+,c} + \hat{\mathcal{L}}_{\gamma_c^-,\neg c} = \frac{1}{n}\sum_{i=1}^{n}\left(1_{[y_i=c]}1_{[s_c(\hat{x}_i)\leq\gamma_c^+]} + 1_{[y_i\neq c]}1_{[s_{\neg c}(\hat{x}_i)\leq\gamma_c^-]}\right) \tag{9}$$

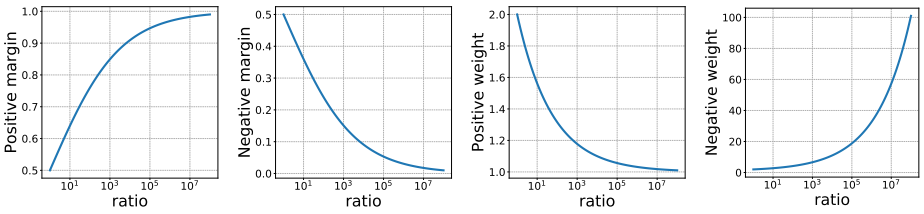


Fig. 4: Visualization of the positive and negative margins and weights as a function of the negative-to-positive ratio α_c.

for a scoring function $s_c(x) \in [0, 1]$. A natural choice of scoring function is a sigmoid $s_c(x) = \frac{\exp(f(x))}{\exp(f(x))+\exp(-f(x))}$.

However, there is no natural equivalent to a margin-based binary cross-entropy (BCE) loss. Regular binary cross-entropy optimizes a margin $s_c(x) = s_{\neg c}(x) = \frac{1}{2}$ at $f(x) = 0$, which does not conform to our margin-based loss. We instead want to move this decision boundary to $s_c(x) = \gamma_c^+$ and $s_{\neg c}(x) = \gamma_c^-$.

We achieve this with a surrogate **Effective Class-Margin Loss**:

$$\mathcal{L}_c^{\text{ECM}} = -\frac{1}{n}\sum_{i=1}^{n} m_c \left(1_{[y=c]} \log(\hat{s}_c(x)) + 1_{[y\neq c]} \log(1 - \hat{s}_c(x))\right). \tag{10}$$

The ECM loss optimizes a binary cross entropy on a surrogate scoring function

$$\hat{s}_c(x) = \frac{w_c^+ e^{f(x)_c}}{w_c^+ e^{f(x)_c} + w_c^- e^{-f(x)_c}}, \qquad w_c^\pm = (\gamma_c^\pm)^{-1}. \tag{11}$$

This surrogate scoring function has the same properties as a sigmoid $\hat{s}_c \in [0, 1]$ and $\hat{s}_{\neg c}(x) = 1 - \hat{s}_c(x)$. However, its decision boundary $\hat{s}_c(x) = \hat{s}_{\neg c}(x)$ lies at $f(x) = \frac{1}{2}(\log w_c^- - \log w_c^+)$. In the original sigmoid scoring function s, this decision boundary corresponds to $s_c(x) = \gamma_c^+$ and $s_{\neg c}(x) = \gamma_c^-$. Hence, the **Effective Class-Margin Loss** minimizes the binary classification error under the margins specified by our empirical objective (9). In Figure 4, we visualize the relationship between the negative-to-positive ratio α_c and the positive and negative class margins/weights.

We use this ECM loss as a plug-in replacement to the standard binary cross entropy or softmax cross entropy used in object detection.

5 Experiments

5.1 Experimental Settings

Datasets. We evaluate our method on LVIS v1.0 [16] and OpenImages datasets. LVIS v1.0 is large-scale object detection and instance segmentation dataset. It

Table 1: LVIS v1 validation set results. We compare different methods on various frameworks and backbones on 2× schedule. For Swin-B backbones [29], we use ImageNet-21k pretrained weight as initialization. We used the results of the original papers if available, and reproduced them from the official code otherwise.

Framework	Backbone	Method	mAP_{segm}	AP_r	AP_c	AP_f	mAP_{bbox}
Mask R-CNN	ResNet-50	CE Loss	22.7	10.6	21.8	29.1	23.3
		Federated Loss [52]	26.0	18.7	24.8	30.6	26.7
		Seesaw Loss [44]	26.7	18.0	26.5	**32.4**	27.3
		LOCE [13]	26.6	18.5	26.2	30.7	27.4
		ECM Loss	**27.4**	**19.7**	**27.0**	31.1	**27.9**
Mask R-CNN	ResNet-101	CE Loss	25.5	16.6	24.5	30.6	26.6
		EQL v1 [40]	26.2	17.0	26.2	30.2	27.6
		BAGS [24]	25.8	16.5	25.7	30.1	26.5
		EQL v2 [39]	27.2	20.6	25.9	31.4	27.9
		Federated Loss [52]	27.9	20.9	26.8	32.3	28.8
		Seesaw Loss [44]	28.1	20.0	**28.0**	31.8	28.9
		LOCE [13]	28.0	19.5	27.8	32.0	29.0
		ECM Loss	**28.7**	**21.9**	27.9	**32.3**	**29.4**
Cascade Mask R-CNN	ResNet-101	CE Loss	27.0	16.6	26.7	32.0	30.3
		EQL v1 [40]	27.1	17.0	27.2	31.4	30.4
		De-confound TDE [41]	27.1	16.0	26.9	32.1	30.0
		BAGS [24]	27.0	16.9	26.9	31.7	30.2
		Federated Loss [24]	28.6	20.3	27.5	33.4	31.8
		DisAlign [51]	28.9	18.0	29.3	33.3	32.7
		Seesaw Loss [44]	30.1	**21.4**	30.0	33.9	32.8
		ECM Loss	**30.6**	19.7	**30.7**	**35.0**	**33.4**
Cascade Mask R-CNN	Swin-B	Seesaw Loss [44]	38.7	**34.3**	39.6	39.6	42.8
		ECM Loss	**39.7**	33.5	**40.6**	**41.4**	**43.6**

includes 1203 object categories that follow the extreme long-tail distribution. Object categories in LVIS dataset are divided into three groups by frequency: *frequent*, *common*, and *rare*. Categories that appear in less than 10 images are considered rare, more than 10 but less than 100 are common, and others are frequent. There are about 1.3 M instances in the dataset over 120k images (100k train, 20k validation split). The OpenImages Object Detection dataset contains 500 object categories over 1.7 M images in a similar long-tail.

Evaluation. We evaluate all models using both the conventional mAP evaluation metric and the newly proposed mAP_{fixed} metric [9]. mAP measures the mean average precision over IoU thresholds from 0.5 to 0.95 [27] over 300 detections per image. mAP_{fixed} [9] has no limit for detections per image, but instead limits the number of detections per class over the entire dataset to 10k. Due to memory limitations, we also limit detections per image to 800 per class. We further evaluate the boundary IoU $mAP_{boundary}$ [8], the evaluation metric in this year's LVIS Challenge. For OpenImages, we follow the evaluation protocol of Zhou *et al.* [53] and measure mAP@0.5.

Table 2: Comparison on LVIS v1 validation set. Models are trained with Mask R-CNN with ResNet-50 backbone on 1x schedule. Numbers with * use a different implementation [9]. $\text{mAP}^{\text{fixed}}_{\text{boundary}}$ and $\text{mAP}^{\text{fixed}}_{\text{bbox}}$ refer to the new LVIS Challenge evaluation metrics [8, 9].

Method	mAP_{segm}	AP_{r}	AP_{c}	AP_{f}	mAP_{bbox}	$\text{mAP}^{\text{fixed}}_{\text{boundary}}$	$\text{mAP}^{\text{fixed}}_{\text{bbox}}$
RFS+CE Loss	21.7	9.5	21.1	27.7	22.2	18.3	25.7
LWS [20]	17.0	2.0	13.5	27.4	17.5	-	-
cRT [20]	22.1	11.9	20.2	29.0	22.2	-	-
BAGS [24]	23.1	13.1	22.5	28.2	23.7	-	26.2*
EQL v2 [39]	23.9	12.5	22.7	30.4	24.0	20.3	25.9
Federated Loss [52]	23.9	15.8	23.3	30.7	24.9	-	26.3*
Seesaw Loss [44]	25.2	16.4	24.4	**30.8**	25.4	19.8	26.5
ECM Loss	**26.3**	**19.5**	**26.0**	29.8	**26.7**	**21.4**	**27.4**

5.2 Implementation Details.

Our implementation is based on Detectron2 [47] and MMDetection [6], two most popular open-source libraries for object detection tasks. We train both Mask R-CNN [17] and Cascade Mask R-CNN [3] with various backbones: ResNet-50 and ResNet-101 [18] with Feature Pyramid Network [25], and the Swin Transformer [29]. We use a number of popular one-stage detectors: FCOS [42], ATSS [50] and VarifocalNet [49]. We largely follow the standard COCO and LVIS setup and hyperparameters for all models. For OpenImages, we follow the setup of Zhou *et al.* [53]. More details are in the supplementary material.

ECM Loss. Our ECM Loss is a plug-in replacement to the sigmoid function used in most detectors. Notably, ECM Loss does not require any hyper-parameter. We use the training set from each dataset to measure $\alpha_c, n_c, n_{\neg c}$ of each class.

5.3 Experimental Results

Table 1 compares our approach on frameworks and backbones using a standard $2\times$ training schedule. We compare different long-tail loss functions under different experimental setups. With Mask R-CNN on a ResNet-50 backbone, our ECM Loss outperforms all alternative losses by 0.7 mAP_{segm} and 0.5 mAP_{bbox}. With Mask R-CNN on a ResNet-101 backbone, our ECM loss outperforms alternatives with a 0.6 mAP_{segm} and 0.4 mAP_{bbox}. The results also hold up in the more advanced Cascade R-CNN framework [3] with ResNet-101 and Swin-B backends. Here the gains are 0.5 mAP_{segm} and 0.6 mAP_{bbox} for ResNet-101, and 1 mAP_{segm} and 0.8 mAP_{bbox} for Swin-B. The overall gains over a simple cross-entropy baseline are 3-5 mAP throughout all settings. The consistent improvement in accuracy throughout all settings highlights the empirical efficacy of our method, in addition to the grounding in learning theory.

Table 3: One-stage object detection results on LVIS v1 validation set. We compare popular one-stage detectors with ResNet-50 and ResNet-101 backbones, on 1x schedule.

Framework	Backbone	Method	AP_r	AP_c	AP_f	mAP_{bbox}
FCOS	ResNet-50	Focal Loss [26]	11.2	21.0	**27.8**	22.0
		ECM Loss	**14.5**	**22.7**	27.6	**23.2**
FCOS	ResNet-101	Focal Loss [26]	14.1	22.6	**29.8**	24.0
		ECM Loss	**17.2**	**24.2**	29.6	**25.1**
ATSS	ResNet-50	Focal Loss [26]	8.6	20.5	**29.8**	22.1
		ECM Loss	**15.8**	**23.5**	29.5	**24.5**
ATSS	ResNet-101	Focal Loss [26]	12.9	24.0	**31.9**	25.2
		ECM Loss	**17.7**	**25.6**	31.5	**26.5**
VarifocalNet	ResNet-50	Varifocal Loss [49]	14.2	23.6	**30.7**	24.8
		ECM Loss	**17.1**	**25.5**	29.7	**25.7**

For reference, Table 2 compares our method on Mask R-CNN with ResNet-50 backbone on 1× schedule. Our ECM Loss outperforms all prior approaches by 1.1 mAP_{segm} and 1.3 mAP_{bbox}. Our method achieves a 10 mAP gain over cross-entropy loss baseline, and 3.1 AP_r gain over the state-of-the-art. Our method shows similar gains on the new evaluation metrics $mAP^{fixed}_{boundary}$ and mAP^{fixed}_{bbox} whereas prior methods tend show a more moderate improvement on new metrics. This improvement is particularly noteworthy, as our approach uses no additional hyperparameters, and is competitive out of the box.

Table 3 compares our ECM Loss with baseline losses on FCOS, ATSS and VarifocalNet, trained with ResNet-50 and ResNet-101 backbones on 1x schedule. The ECM Loss shows consistent gains over Focal Loss and its variants. With FCOS, ECM improves box mAP 1.2 and 1.1 points, respectively, for ResNet-50 and ResNet-101 backbones. With ATSS, ECM Loss improves Focal Loss 7.2 and 4.8 points on AP_r, and 2.4 and 1.3 points mAP, respectively. We further test on VarifocalNet, a recently proposed one-stage detector, and show similar advantage using the ResNet-50 backbone. Our ECM loss consistently improves the overall performance of a one-stage detector, especially in rare classes. It thus serves as a true plug-in replacement to the standard cross-entropy or focal losses. Table 4 further analyze our ECM Loss with Focal Loss on FCOS and ATSS trained with ResNet-50 backbone on 2x schedule. Our ECM maintains improvement of 0.9 point mAP for FCOS, and 0.5 point mAP for ATSS. For AP_r, both methods consistently improve 2.7 and 2.1 points, respectively.

In Table 5, we compare ECM Loss with a variant of Equalization Loss on the class hierarchy of Zhou *et al.* [53]. Although OpenImages have a long-tail distribution of classes, the number of classes and the associated prior probabilities are very different. Nevertheless, ECM Loss improves over the baseline for 1.2 mAP. This result confirms the generality of our method.

Table 4: One-stage object detection results on LVIS v1 validation set. We compare different methods with ResNet-50 backbone on 2x schedule.

Framework	Backbone	Method	AP_r	AP_c	AP_f	mAP_{bbox}
FCOS	ResNet-50	Focal Loss	12.0	22.9	**29.5**	23.5
		ECM Loss	**14.7**	**23.0**	**29.5**	**24.4**
ATSS	ResNet-50	Focal Loss	14.5	24.3	**31.8**	25.6
		ECM Loss	**16.6**	**25.2**	31.3	**26.1**

Table 5: Comparisons of ECM Loss on OpenImages dataset following the evaluation protocol of Zhou *et al.* [53]. We compare using a Cascade R-CNN with ResNet-50 backbone.

Framework	Backbone	Method	Schedule	mAP
Cascade R-CNN	ResNet-50	EQL + Hier. [40, 53]	2x	64.6
		ECM Loss	2x	**65.8**

For all our experiments, the class frequencies were measured directly from the annotation set of LVIS v1 training dataset. Note that for each class, the negative sample not only includes other foreground classes but also the background class. However, the prior probability for background class is not defined apriori from the dataset itself since it solely depends on the particular detection framework of choice. Hence, we measure the background frequency for each detector of choice and factor it into the final derivation of overall class frequencies. This can be done within the first few iterations during training. We then compute the effective class-margins with the derived optimal solution in Eqn. (8) and finally define the surrogate scoring function (11).

6 Conclusion

In this paper, we tackle the long-tail object detection problem using a statistical approach. We connect the training objective and the detection evaluation objective in the form of margin theory. We show how a probabilistic version of average precision is optimized using a ranking and then margin-based binary classification problem. We present a novel loss function, called Effective Class-Margin (ECM) Loss, to optimize the margin-based classification problem. This ECM loss serves as a plug-in replacement to standard cross-entropy-based losses across various detection frameworks, backbones, and detector designs. The ECM loss consistently improves the performance of the detector in a long-tail setting. The loss is simple and hyperparameter-free.

Acknowledgments. This material is in part based upon work supported by the National Science Foundation under Grant No. IIS-1845485 and IIS-2006820.

References

1. Bartlett, P., Foster, D.J., Telgarsky, M.: Spectrally-normalized margin bounds for neural networks. arXiv preprint arXiv:1706.08498 (2017)
2. Bartlett, P.L., Mendelson, S.: Rademacher and gaussian complexities: Risk bounds and structural results. J. Mach. Learn. Res. **3**(null), 463–482 (mar 2003)
3. Cai, Z., Vasconcelos, N.: Cascade r-cnn: Delving into high quality object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 6154–6162 (2018)
4. Cao, K., Wei, C., Gaidon, A., Arechiga, N., Ma, T.: Learning imbalanced datasets with label-distribution-aware margin loss. arXiv preprint arXiv:1906.07413 (2019)
5. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European conference on computer vision. pp. 213–229. Springer (2020)
6. Chen, K., Wang, J., Pang, J., Cao, Y., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z., Xu, J., Zhang, Z., Cheng, D., Zhu, C., Cheng, T., Zhao, Q., Li, B., Lu, X., Zhu, R., Wu, Y., Dai, J., Wang, J., Shi, J., Ouyang, W., Loy, C.C., Lin, D.: MMDetection: Open mmlab detection toolbox and benchmark. arXiv preprint arXiv:1906.07155 (2019)
7. Chen, K., Lin, W., Li, J., See, J., Wang, J., Zou, J.: Ap-loss for accurate one-stage object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence **43**(11), 3782–3798 (2020)
8. Cheng, B., Girshick, R., Dollár, P., Berg, A.C., Kirillov, A.: Boundary iou: Improving object-centric image segmentation evaluation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15334–15342 (2021)
9. Dave, A., Dollár, P., Ramanan, D., Kirillov, A., Girshick, R.: Evaluating large-vocabulary object detectors: The devil is in the details. arXiv preprint arXiv:2102.01066 (2021)
10. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp. 248–255 (2009). https://doi.org/10.1109/CVPR.2009.5206848
11. Deng, J., Guo, J., Xue, N., Zafeiriou, S.: Arcface: Additive angular margin loss for deep face recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4690–4699 (2019)
12. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. International Journal of Computer Vision **88**(2), 303–338 (Jun 2010)
13. Feng, C., Zhong, Y., Huang, W.: Exploring classification equilibrium in long-tailed object detection. In: ICCV (2021)
14. Ghiasi, G., Cui, Y., Srinivas, A., Qian, R., Lin, T.Y., Cubuk, E.D., Le, Q.V., Zoph, B.: Simple copy-paste is a strong data augmentation method for instance segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2918–2928 (2021)
15. Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 1440–1448 (2015)
16. Gupta, A., Dollar, P., Girshick, R.: Lvis: A dataset for large vocabulary instance segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5356–5364 (2019)

17. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 2961–2969 (2017)
18. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
19. Kakade, S.M., Sridharan, K., Tewari, A.: On the complexity of linear prediction: Risk bounds, margin bounds, and regularization. In: Proceedings of the 21st International Conference on Neural Information Processing Systems. p. 793–800. NIPS'08, Curran Associates Inc., Red Hook, NY, USA (2008)
20. Kang, B., Xie, S., Rohrbach, M., Yan, Z., Gordo, A., Feng, J., Kalantidis, Y.: Decoupling representation and classifier for long-tailed recognition. In: International Conference on Learning Representations (2020), https://openreview.net/forum?id=r1gRTCVFvB
21. Koltchinskii, V., Panchenko, D.: Empirical margin distributions and bounding the generalization error of combined classifiers. The Annals of Statistics **30**(1), 1–50 (2002)
22. Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Malloci, M., Kolesnikov, A., et al.: The open images dataset v4. International Journal of Computer Vision **128**(7), 1956–1981 (2020)
23. Li, X., Wang, W., Wu, L., Chen, S., Hu, X., Li, J., Tang, J., Yang, J.: Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. Advances in Neural Information Processing Systems **33**, 21002–21012 (2020)
24. Li, Y., Wang, T., Kang, B., Tang, S., Wang, C., Li, J., Feng, J.: Overcoming classifier imbalance for long-tail object detection with balanced group softmax. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10991–11000 (2020)
25. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2117–2125 (2017)
26. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. pp. 2980–2988 (2017)
27. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014)
28. Liu, W., Wen, Y., Yu, Z., Yang, M.: Large-margin softmax loss for convolutional neural networks. In: ICML. vol. 2, p. 7 (2016)
29. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. International Conference on Computer Vision (ICCV) (2021)
30. Oksuz, K., Cam, B., Akbas, E., Kalkan, S.: Localization recall precision (lrp): A new performance metric for object detection. In: European Conference on Computer Vision (ECCV) (2018)
31. Oksuz, K., Cam, B.C., Akbas, E., Kalkan, S.: A ranking-based, balanced loss function unifying classification and localisation in object detection. In: Advances in Neural Information Processing Systems (NeurIPS) (2020)
32. Oksuz, K., Cam, B.C., Akbas, E., Kalkan, S.: Rank & sort loss for object detection and instance segmentation. In: International Conference on Computer Vision (ICCV) (2021)

33. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 779–788 (2016)
34. Redmon, J., Farhadi, A.: Yolo9000: better, faster, stronger. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7263–7271 (2017)
35. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767 (2018)
36. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems **28**, 91–99 (2015)
37. Rolinek, M., Musil, V., Paulus, A., Vlastelica, M., Michaelis, C., Martius, G.: Optimizing rank-based metrics with blackbox differentiation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2020)
38. Shao, S., Li, Z., Zhang, T., Peng, C., Yu, G., Zhang, X., Li, J., Sun, J.: Objects365: A large-scale, high-quality dataset for object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (October 2019)
39. Tan, J., Lu, X., Zhang, G., Yin, C., Li, Q.: Equalization loss v2: A new gradient balance approach for long-tailed object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1685–1694 (2021)
40. Tan, J., Wang, C., Li, B., Li, Q., Ouyang, W., Yin, C., Yan, J.: Equalization loss for long-tailed object recognition. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11662–11671 (2020)
41. Tang, K., Huang, J., Zhang, H.: Long-tailed classification by keeping the good and removing the bad momentum causal effect. In: NeurIPS (2020)
42. Tian, Z., Shen, C., Chen, H., He, T.: Fcos: Fully convolutional one-stage object detection. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 9627–9636 (2019)
43. Wang, F., Cheng, J., Liu, W., Liu, H.: Additive margin softmax for face verification. IEEE Signal Processing Letters **25**(7), 926–930 (2018)
44. Wang, J., Zhang, W., Zang, Y., Cao, Y., Pang, J., Gong, T., Chen, K., Liu, Z., Loy, C.C., Lin, D.: Seesaw loss for long-tailed instance segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9695–9704 (2021)
45. Wang, T., Li, Y., Kang, B., Li, J., Liew, J., Tang, S., Hoi, S., Feng, J.: The devil is in classification: A simple framework for long-tail instance segmentation. arXiv preprint arXiv:2007.11978 (2020)
46. Wu, J., Song, L., Wang, T., Zhang, Q., Yuan, J.: Forest r-cnn: Large-vocabulary long-tailed object detection and instance segmentation. In: Proceedings of the 28th ACM International Conference on Multimedia. pp. 1570–1578 (2020)
47. Wu, Y., Kirillov, A., Massa, F., Lo, W.Y., Girshick, R.: Detectron2. `https://github.com/facebookresearch/detectron2` (2019)
48. Zhang, C., Pan, T.Y., Li, Y., Hu, H., Xuan, D., Changpinyo, S., Gong, B., Chao, W.L.: Mosaicos: a simple and effective use of object-centric images for long-tailed object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 417–427 (2021)
49. Zhang, H., Wang, Y., Dayoub, F., Sunderhauf, N.: Varifocalnet: An iou-aware dense object detector. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8514–8523 (2021)

50. Zhang, S., Chi, C., Yao, Y., Lei, Z., Li, S.Z.: Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9759–9768 (2020)
51. Zhang, S., Li, Z., Yan, S., He, X., Sun, J.: Distribution alignment: A unified framework for long-tail visual recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2361–2370 (2021)
52. Zhou, X., Koltun, V., Krähenbühl, P.: Probabilistic two-stage detection. arXiv preprint arXiv:2103.07461 (2021)
53. Zhou, X., Koltun, V., Krähenbühl, P.: Simple multi-dataset detection. In: arXiv preprint arXiv:2102.13086 (2021)
54. Zhou, X., Wang, D., Krähenbühl, P.: Objects as points. arXiv preprint arXiv:1904.07850 (2019)

A Pairwise Ranking Error

In this section, we will prove the second equality of Definition 2.

$$\begin{aligned} R_c &= P_{x' \sim D_{\neg c}, x \sim D_c} \left(s_x(x) < s_c(x') \right)) \\ &= 1 - E_{x' \sim D_{\neg c}} \left[P_{x \sim D_c} \left(s_x(x) > s_c(x') \right) \right)] \\ &= E_{x' \sim D_{\neg c}} \left[1 - r_c(s_c(x')) \right] \\ &= \int_0^1 \underbrace{P_{x' \sim D_{\neg c}}(r_c(s_c(x')) < \beta)}_{=g(\beta)} \mathrm{d}\beta = \tau. \end{aligned}$$

where the definition of g is

$$g(\beta) = P_{x' \sim D_{\neg c}} \left(s_c(x') > r_c^{-1}(\beta) \right) = P_{x' \sim D_{\neg c}} \left(r_c(s_c(x')) < \beta \right).$$

The above derivation connects the ranking error to g and the recall.

B AP - Pairwise Ranking Error Bound

In this section, we will prove Theorem 2. We first derive the and lower bounds to the variational objective $\int_0^1 \frac{x}{x + \alpha g(x)} \mathrm{d}x$ under constraint $\int_0^1 g(x) \mathrm{d}x = \tau$ for a function $g(x) \geq 0$. The AP bounds then directly reduce to the variational objective.

Lemma 1. *Consider the following variational problem*

$$\begin{gathered} \text{minimize}_g \int_0^1 \frac{x}{x + \alpha g(x)} \mathrm{d}x \\ \text{subject to} \int_0^1 g(x) \mathrm{d}x = \tau \\ g(x) \geq 0 \end{gathered}$$

The solution to this problem is

$$\max \left(1 - \sqrt{\frac{2}{3} \alpha \tau}, \frac{4}{9} \frac{1}{\frac{1}{2} + \alpha \tau} \right)$$

Proof. Consider the associated Euler-Lagrangian equation:

$$L(x, v(x), \lambda) = \int_0^1 \frac{x}{x + \alpha v(x)^2} \mathrm{d}x + \lambda \left(\int_0^1 v(x)^2 \mathrm{d}x - \tau \right)$$

where $g(x) = v(x)^2$ *for the non-negativity constraint. To solve for minima*

$$\begin{aligned}
\frac{d}{dv(x)}L(x, v(x), \lambda) &= -\frac{\alpha x v(x)}{(x + \alpha v(x)^2)^2} + \lambda v(x) = 0 \\
v(x)\alpha x &= \lambda v(x)(x + \alpha v(x)^2)^2 \\
\implies v(x) = 0 \quad or \quad x + \alpha v(x)^2 &= \sqrt{\frac{x\alpha}{\lambda}} = \sqrt{x}\sqrt{\frac{\alpha}{\lambda}} \\
\implies v(x)^2 &= \max\left(0, \sqrt{\frac{x}{\alpha\lambda}} - \frac{x}{\alpha}\right) = 1_{[x \leq \frac{\alpha}{\lambda}]}\left(\sqrt{\frac{x}{\alpha\lambda}} - \frac{x}{\alpha}\right) \\
\implies \int_0^1 \alpha v(x)^2 \mathrm{d}x = \alpha\tau &= \int_0^1 1_{[x \leq \frac{\alpha}{\lambda}]}\left(\sqrt{\frac{\alpha x}{\lambda}} - x\right)\mathrm{d}x \\
&= \int_0^\kappa \left(\sqrt{\frac{\alpha x}{\lambda}} - x\right)\mathrm{d}x = \frac{2}{3}\kappa^{\frac{3}{2}}\sqrt{\frac{\alpha}{\lambda}} - \frac{\kappa^2}{2}
\end{aligned}$$

where $\kappa = \min(1, \frac{\alpha}{\lambda})$. *For* $\kappa = 1$*:*

$$\begin{aligned}
\alpha\tau = \frac{2}{3}\sqrt{\frac{\alpha}{\lambda}} - \frac{1}{2} \implies \sqrt{\frac{\alpha}{\lambda}} &= \frac{3}{2}\left(\alpha\tau + \frac{1}{2}\right) \\
\implies x + \alpha v(x)^2 = \sqrt{x}\sqrt{\frac{\alpha}{\lambda}} &= \sqrt{x}\frac{3}{2}\left(\alpha\tau + \frac{1}{2}\right) \\
\implies \int_0^1 \frac{x}{x + \alpha v(x)^2}\mathrm{d}x = \int_0^1 \frac{x}{\sqrt{x}\frac{3}{2}(\alpha\tau + \frac{1}{2})}\mathrm{d}x &= \int_0^1 \sqrt{x}\mathrm{d}x \frac{1}{\frac{3}{2}(\frac{1}{2} + \alpha\tau)} \\
&= \frac{4}{9}\frac{1}{\frac{1}{2} + \alpha\tau}
\end{aligned}$$

For $\kappa < 1$*:*

$$\begin{aligned}
\alpha\tau = \frac{2}{3}\left(\frac{\alpha}{\lambda}\right)^{\frac{3}{2}}\sqrt{\frac{\alpha}{\lambda}} - \frac{1}{2}\left(\frac{\alpha}{\lambda}\right)^2 &= \frac{1}{6}\left(\frac{\alpha}{\lambda}\right)^2 \implies \lambda = \frac{1}{6}\sqrt{\frac{\alpha}{\tau}} \\
\implies \int_0^1 \frac{x}{x + \alpha v(x)^2}\mathrm{d}x &= \int_0^\kappa \frac{x}{\sqrt{\frac{\alpha x}{\lambda}}}\mathrm{d}x + \int_\kappa^1 \frac{x}{x + 0}\mathrm{d}x \\
&= \sqrt{\frac{\lambda}{\alpha}}\int_0^\kappa \sqrt{x}\mathrm{d}x + \int_\kappa^1 \mathrm{d}x \\
&= \frac{2}{3}\sqrt{\frac{\lambda}{\alpha}}\kappa^{\frac{3}{2}} + 1 - \kappa \\
&= \frac{2}{3}\sqrt{\frac{\lambda}{\alpha}}\left(\frac{\alpha}{\lambda}\right)^{\frac{3}{2}} + 1 - \frac{\alpha}{\lambda} \\
&= 1 - \frac{1}{3}\frac{\alpha}{\lambda} \\
&= 1 - \sqrt{\frac{2}{3}\alpha\tau}
\end{aligned}$$

Each case yields on lower bound, hence the combined lower bound is

$$\max\left(1-\sqrt{\frac{2}{3}\alpha\tau}, \frac{4}{9}\frac{1}{\frac{1}{2}+\alpha\tau}\right)$$

□

Bonus: The two bounds meet at $\frac{2}{3}$:

$$\frac{4}{9}\frac{1}{\frac{1}{2}+\alpha\tau} = 1-\sqrt{\frac{2}{3}\alpha\tau} = \frac{2}{3} \quad \text{for} \quad \alpha\tau = \frac{1}{6}.$$

Lemma 2. *Consider the following variational problem*

$$\begin{aligned} &\text{maximize}_g \int_0^1 \frac{x}{x+\alpha g(x)}\mathrm{d}x \\ &\text{subject to} \int_0^1 g(x)\mathrm{d}x = \tau \\ &g(x) \geq 0 \end{aligned}$$

Proof. First, let us re-formulate the problem as following

$$\begin{aligned} &\text{minimize}_g \int_0^1 \frac{\alpha g(x)}{x+\alpha g(x)}\mathrm{d}x \\ &\text{subject to} \int_0^1 g(x)\mathrm{d}x = \tau \\ &g(x) \geq 0 \end{aligned}$$

Without the equality constraint $\int_0^1 g(x)\mathrm{d}x = \tau$, the objective is minimized at $g(x) = 0$ for all $x \in [0,1]$. The equality constraint assigns certain values $g(x)$ a positive mass. The optimal solution will assign $g(x) = 0$ for $x < 1-\tau$, and $g(x) = 1$ for $x \geq 1-\tau$. To see this, consider a value $g(x_1) = \frac{\epsilon}{\alpha}$ for $x_1 < 1-\tau$ and one or move values $g(x_2) \leq 1-\frac{\epsilon}{\alpha}$ for $x_2 > 1-\tau$. Here, a solution $\hat{g}(x_1) = 0$ and $\hat{g}(x_2) = g(x_2) + \frac{\epsilon}{\alpha}$ has a lower objective

$$\begin{aligned} \Delta &= \left(\frac{\alpha g(x_1)}{x_1+\alpha g(x_1)} + \frac{\alpha g(x_2)}{x_2+\alpha g(x_2)}\right) - \left(\frac{\alpha \hat{g}(x_1)}{x_1+\alpha \hat{g}(x_1)} + \frac{\alpha \hat{g}(x_2)}{x_2+\alpha \hat{g}(x_2)}\right) \\ &= \left(\frac{\alpha g(x_1)}{x_1+\alpha g(x_1)} + \frac{\alpha g(x_2)}{x_2+\alpha g(x_2)}\right) - \frac{\alpha g(x_2)+\epsilon}{x_2+\alpha g(x_2)+\epsilon} \\ &= \frac{\epsilon}{x_1+\epsilon} - \frac{\epsilon x_2}{(x_2+\alpha g(x_2))(x_2+\alpha g(x_2)+\epsilon)} > 0 \end{aligned}$$

Here $\Delta > 0$ and the new objective is lower since $x_2 + \alpha g(x_2) > x_2$ and $x_2 > x_1$ thus $(x_2+\alpha g(x_2))(x_2+\alpha g(x_2)+\epsilon) > x_2(x_1+\epsilon)$.

Thus the zero-mass region should be where x is low as lower x increases the objective. Hence, the optimality will happen when $g(x) = 0$ for $x \in [0, 1-\tau]$, and $g(x) = 1$ for $x \in [1-\tau, 1]$. Thus:

$$\begin{aligned}\int_{1-\tau}^{1} \frac{\alpha}{\alpha + x} \mathrm{d}x &= \alpha \int_{1-\tau}^{1} \frac{1}{\alpha + x} \mathrm{d}x = \alpha(\log(1+\alpha) - \log(1-\tau+\alpha)) \\ &= -\alpha \log\left(1 - \frac{\tau}{1+\alpha}\right) \\ \implies \max_g \int_0^1 \frac{x}{x + \alpha g(x)} \mathrm{d}x &= 1 + \alpha \log\left(1 - \frac{\tau}{1+\alpha}\right)\end{aligned}$$

which concludes the proof. □

Lemma 1 and Lemma 2 for the bounds to the AP.

Theorem 4. *Average Precision can be bounded from above and below as following*

$$1 + \alpha_c \log\left(1 - \frac{R_c}{1+\alpha_c}\right) \geq AP_c \geq \max\left(1 - \sqrt{\frac{2}{3}\alpha_c R_c}, \frac{8}{9}\frac{1}{1+2\alpha_c R_c}\right) \tag{12}$$

Proof. Let us recap the definitions of AP and R:

$$\begin{aligned}AP_c &= \int_0^1 \frac{\beta}{\beta + \alpha_c P_{x\sim D_{\neg c}}(s_c(x) > r_c^{-1}(\beta))} \mathrm{d}\beta \\ &= \int_0^1 \frac{\beta}{\beta + \alpha_c \underbrace{P_{x\sim D_{\neg c}}(r_c(s_c(x)) < \beta)}_{=g(\beta)}} \mathrm{d}\beta \\ R_c &= \int_0^1 \underbrace{P_{x'\sim D_{\neg c}}(r_c(s_c(x')) < \beta)}_{=g(\beta)} \mathrm{d}\beta = \tau\end{aligned}$$

where the second line of AP_c is because r_c is strictly monotonously decreasing. With $x = \beta$ and $g(x) = P_{x\sim D_{\neg c}}(r_c(s_c(x)) < \beta)$, Lemma 1 and Lemma 2 are directly applicable for a function $0 \leq g(x) \leq 1$ with a fixed $R_c = \tau$. The corresponding upper and lower bounds of $\mathcal{L}_c^{Det}$ in Theorem 2 is a direct consequence of this theorem since $\mathcal{L}_c^{Det} = 1 - AP_c$. □

C Optimal Margins

Similar to Cao et al. [4], we aim to find optimal binary margins γ_+ and γ_- under separability condition. This reduces the problem into following:

$$\text{minimize}_{\gamma_+,\gamma_-} \quad \frac{1}{\gamma_+}\sqrt{\frac{1}{n_+}} + \frac{1}{\gamma_-}\sqrt{\frac{1}{n_-}} \tag{13}$$

$$\text{subject to} \quad \gamma_+ + \gamma_- = 1 \tag{14}$$

Here the constraint is due to the fact that $s_-(x) = 1 - s_+(x)$ in binary case and thus $\gamma_- = 1 - \gamma_+$. Solving the constrained optimization problem

$$L(\gamma_+, \gamma_-, \lambda) = \frac{1}{\gamma_+}\sqrt{\frac{1}{n_+}} + \frac{1}{\gamma_-}\sqrt{\frac{1}{n_-}} + \lambda(\gamma_+ + \gamma_- - 1) \tag{15}$$

$$\implies \frac{\partial}{\partial \gamma_+} L(\gamma_+, \gamma_-, \lambda) = -\frac{1}{{\gamma_+}^2}\sqrt{\frac{1}{n_+}} + \lambda = 0 \tag{16}$$

$$\implies \gamma_+ = \sqrt{\frac{{n_+}^{-\frac{1}{2}}}{\lambda}}, \quad \gamma_- = \sqrt{\frac{{n_-}^{-\frac{1}{2}}}{\lambda}} \tag{17}$$

$$\implies \frac{\partial}{\partial \lambda} L(\gamma_+, \gamma_-, \lambda) = \gamma_+ + \gamma_- - 1 = 0 \tag{18}$$

$$\implies \gamma_+ + \gamma_- = \sqrt{\frac{{n_+}^{-\frac{1}{2}}}{\lambda}} + \sqrt{\frac{{n_-}^{-\frac{1}{2}}}{\lambda}} = 1 \tag{19}$$

$$\implies \sqrt{\lambda} = \frac{\sqrt{{n_+}^{-\frac{1}{2}}} + \sqrt{{n_-}^{-\frac{1}{2}}}}{1} \tag{20}$$

$$\implies \gamma_+ = \frac{{n_+}^{-\frac{1}{4}}}{{n_+}^{-\frac{1}{4}} + {n_-}^{-\frac{1}{4}}} = \frac{{n_-}^{\frac{1}{4}}}{{n_+}^{\frac{1}{4}} + {n_-}^{\frac{1}{4}}} \tag{21}$$

$$\gamma_- = \frac{{n_-}^{-\frac{1}{4}}}{{n_+}^{-\frac{1}{4}} + {n_-}^{-\frac{1}{4}}} = \frac{{n_+}^{\frac{1}{4}}}{{n_+}^{\frac{1}{4}} + {n_-}^{\frac{1}{4}}} \tag{22}$$

which are as desired. The exact same process can be repeated for each class $c \in C$ and we will have our Effective Class-Margins. □

D Surrogate Scoring Function

In this section, we will justify the choice of our surrogate scoring function.

$$\hat{s}_c(x) = \frac{w_c^+ e^{f(x)_c}}{w_c^+ e^{f(x)_c} + w_c^- e^{-f(x)_c}} \tag{23}$$

The decision boundary is then

$$\hat{s}_c(x) = \hat{s}_{\neg c}(x) = 1 - \hat{s}_c(x) \tag{24}$$

$$\implies \frac{w_c^+ e^{f(x)_c}}{w_c^+ e^{f(x)_c} + w_c^- e^{-f(x)_c}} = \frac{w_c^- e^{-f(x)_c}}{w_c^+ e^{f(x)_c} + w_c^- e^{-f(x)_c}} \tag{25}$$

$$\implies \log w_c^+ + f(x)_c = \log w_c^- - f(x)_c \tag{26}$$

$$\implies f(x)_c = \frac{1}{2}(\log w_c^- - \log w_c^+) \tag{27}$$

In the unweighted sigmoid function, this point will lie at

$$s_c(x) = \frac{e^{f(x)_c}}{e^{f(x)_c} + e^{-f(x)_c}} \tag{28}$$

$$= \frac{e^{\frac{1}{2}(\log w_c^- - \log w_c^+)}}{e^{\frac{1}{2}(\log w_c^- - \log w_c^+)} + e^{\frac{1}{2}(\log w_c^+ - \log w_c^-)}} \tag{29}$$

$$= \frac{\sqrt{\frac{w_c^-}{w_c^+}}}{\sqrt{\frac{w_c^-}{w_c^+}} + \sqrt{\frac{w_c^+}{w_c^-}}} = \frac{\sqrt{\frac{\gamma_c^+}{\gamma_c^-}}}{\sqrt{\frac{\gamma_c^+}{\gamma_c^-}} + \sqrt{\frac{\gamma_c^-}{\gamma_c^+}}} = \frac{\gamma_c^+}{\gamma_c^+ + \gamma_c^-} \tag{30}$$

$$= \gamma_c^+ \tag{31}$$

$$\implies s_{\neg c}(x) = \gamma_c^- \tag{32}$$

since $\gamma_c^+ + \gamma_c^- = 1$. Hence, we have shown that our surrogate scoring function with effective class-margins shifts the decision boundary of sigmoid function to γ_c^+ and γ_c^- as desired. □

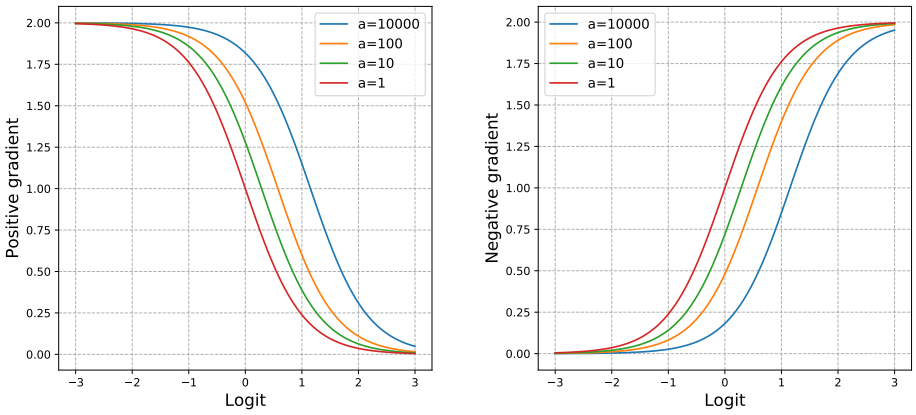


Fig. 5: Visualization of the positive and negative gradients from ECM Loss with different positive and negative samples ratios.

E Margins vs Weights vs Gradients

In this section, we provide more intuitions about the relationship between margins, weights, and gradients. The gradient of positive and negative samples with

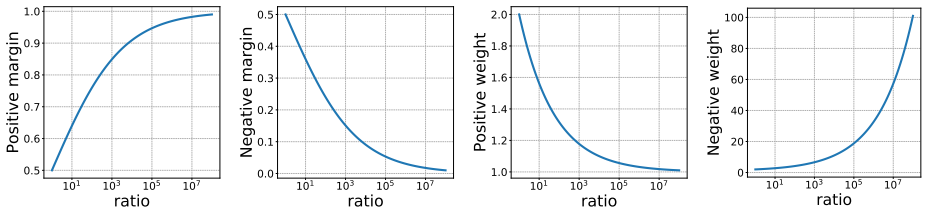
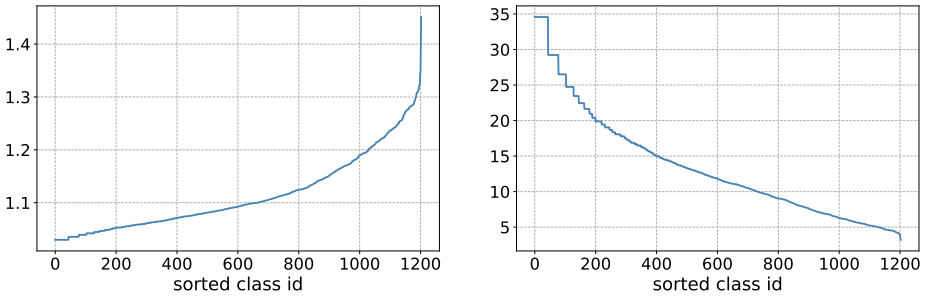


Fig. 6: Visualization of the positive and negative margins and weights as a function of the sample ratio α_c.



(a) w_c^+ (b) w_c^-

Fig. 7: Visualization of the computed the positive and negative weights used in our scoring function $\hat{s}_c$ for each class $c \in C$. The prior distribution is from LVIS v1 training annotations over 1203 classes. The weights are sorted in ascending order and background probability is measured with Mask R-CNN with ResNet-50 backbone.

ECM Loss is the following:

$$\frac{\partial}{\partial f(x)_c}\ell_{\text{ECM}}(x,y) = \frac{2w_{\neg c}e^{f(x)_{\neg c}}}{w_c e^{f(x)_c} + w_{\neg c}e^{f(x)_{\neg c}}} \propto w_{\neg c} = \frac{1}{\gamma_{\neg c}} \propto n_{\neg c} \tag{33}$$

$$\frac{\partial}{\partial f(x)_{\neg c}}\ell_{\text{ECM}}(x,y) = \frac{2w_c e^{f(x)_c}}{w_c e^{f(x)_c} + w_{\neg c}e^{f(x)_{\neg c}}} \propto w_c \ = \frac{1}{\gamma_c} \ \propto n_c \tag{34}$$

where we omit detection weight m_c for simplicity. The positive gradient is greater for rare classes (higher $n_{\neg c}$) compared to frequent classes, whereas the negative gradient is lower. This coincides with the intuitions from prior works based on heuristics or indirect measure of a model. Below, we show visualization of positive and negative gradients as a function of logit with different positive and negative ratios $a_c = \frac{n_{\neg c}}{n_c}$. In Figure 5, low a means frequent classes whereas high a means rare classes. Our surrogate scoring function $\hat{s}_c$ balances the gradient values based on the frequency of each class. Frequent classes get lower positive

gradient and higher negative gradient (red in 5) whereas rare classes get higher positive gradient and lower negative gradient (blue in 5). In Figure 6, we further visualize the relationship between the positive and negative margins and weights as a function of the ratio α_c. In Figure 7, we visualize the computed weights for our scoring function $\hat{s}_c$ in Equation 11 for w_c^+ (left) and w_c^- (right).

F Implementation Details

In this section, we will discuss details of the experiments and implementation.

Background count. We empirically measure the frequency of background samples as a ratio of foreground and background samples within a batch, r, in the *classification layer* of a detector during the first few iterations. This ratio will then be used to derive *dataset-level* count of background samples as

$$n_{bg} = r \sum_{c \in C} n_c \tag{35}$$

where n_c is the number of positive samples of class c in the training dataset. Then, we compute the sample count of each class *with background* as

$$n_c^+ = n_c \tag{36}$$

$$n_c^- = \Big(\sum_{c' \in C \cup \{bg\}} n_{c'} \Big) - n_c \tag{37}$$

and use it to compute the effective class-margins. For one-stage detectors, we only count for foreground classes as foreground and background imbalance is managed from focal weight [26].

Two-stage detectors. We train two-stage instance segmentation models based on Mask R-CNN [17] and Cascade Mask R-CNN [3] with various backbones, ResNet-50 and ResNet-101 [18] with Feature Pyramid Network [25] pretrained on ImageNet-1K [10], Swin Transformer [29] pretrained on ImageNet-21K with 224x224 image resolution. We train with for 12 or 24 epochs with Repeated Factor Sampler (RFS) [16] on a $1\times$ or $2\times$ schedule respectively. For CNN backbones, we use the SGD optimizer with 0.9 momentum, initial learning rate of 0.02, weight decay of 0.0001, with step-wise scheduler decaying learning rate by 0.1 after 8 and 11 epochs for 1x, and 20 and 22 epochs for 2x, and batch size of 16 on 8 GPUs. Please note that the baseline methods are trained with their optimal learning rate schedule with decaying schedule of 16 and 22 epochs. For example, Mask R-CNN with ResNet-50 trained with Seesaw Loss [44] on decaying schedule of 20 and 22 epochs result with 26.7 mAP_{segm} and 26.9 mAP_{bbox}, whereas decaying schedule of 16 and 22 (default) result with 26.7 mAP_{segm} and 27.3 mAP_{bbox}. For Transformer backbones, we use the AdamW optimizer with an initial learning rate of 0.00005, beta set to $(0.9, 0.999)$, weight decay of 0.05, with Cosine-annealing scheduler. For all our models, we follow the standard data augmentation during training: random horizontal flipping and multi-scale image

resizing to fit the shorter side of image to $(640, 672, 704, 736, 768, 800)$ at random, and the longer side kept smaller than 1333. For Swin Transformer, we use a larger range of scale of the short side of image from 480 to 800. For two-stage detectors, we normalize the classification layers with temperature $\tau = 20$ for both box and mask classifications, and apply foreground calibration as post-process following prior practices [9,44,51]. LVIS has more instances than COCO. We thus increase the per-image detection limit to 300 from 100 and set the confidence threshold to 0.0001. This is common practice in LVIS [16]. For OpenImages, we train Cascade R-CNN with ResNet-50 backbone following the baseline and experimental setup of Zhou *et al.* [53]. All models in this experiment were trained for 180k iterations with a class-aware Sampler.

One-stage detectors. For one-stage detectors, Focal Loss [26] is the standard choice of the loss function. It effectively diminishes loss values for "easy" samples such as the background. In this experiment, we test the compatibility of our method with Focal weights. Specifically, we apply the computed focal weights to our ECM Loss. Instead of a binary cross-entropy on the surrogate scoring function $\hat{s}$, we use the focal loss. We use a number of popular one-stage detectors: FCOS [42], ATSS [50] and VarifocalNet [49]. Each method uses either Focal Loss or a variant [49]. We use the default hyperparameter for all types of focal weights: $\gamma = 2, \alpha = 0.25$ for Focal Loss, $\gamma = 2$ and $\alpha = 0.75$ for Varifocal Loss [49]. In LVIS v1.0 models expect to see more instances. We thus double the per-pyramid level number of anchor candidates from 9 to 18 for ATSS and VarifocalNet. Similar to 2-stage detectors, we increased per-image detection to 300 and set the confidence threshold to 0.0001. We train on ResNet-50 and ResNet-101 backbones for 12 epochs for 1x and 24 epochs for 2x, batch size of 16 on 8 GPUs. We set the learning rate to be 0.01 which was the optimal learning rate for the baselines. For all other settings, we follow the two-stage experiments.