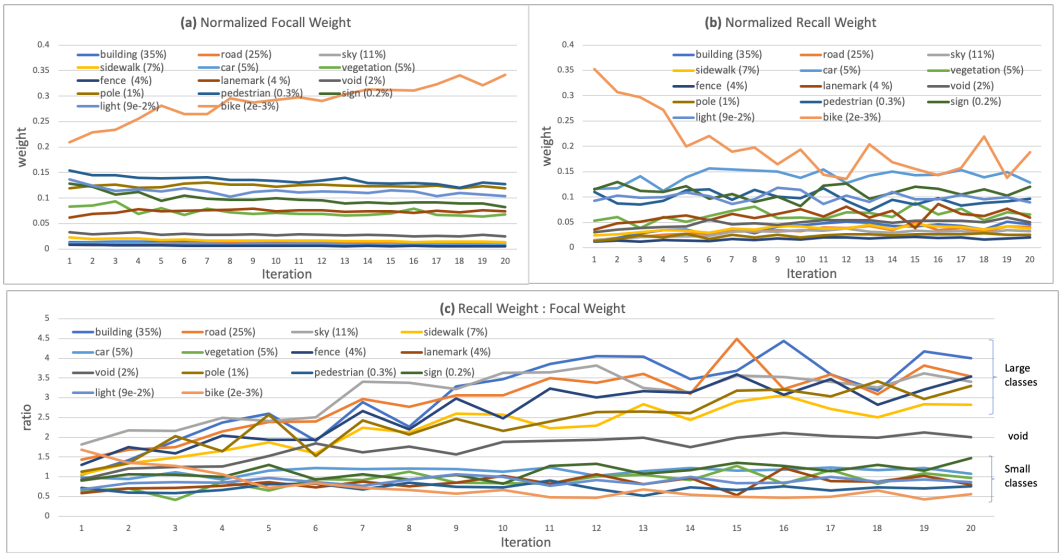

Figure 4: **(a) Normalized focal weight over iterations. (b) Normalized recall weight over iterations. (c) The ratio of recall weight and focal weight over iterations.** We present the three plots to argue that the focal loss can be minimized by making the correct predictions more confident instead of encouraging wrong predictions to become correct.

A Appendix

A.1 Analysis of Focal Loss

In the main paper, we pointed out that the focal loss can "cheat" by increasing the probability of easy classes instead of encouraging more wrong predictions to become correct. Let's explore this insight and compare it to the proposed recall loss qualitatively. We list the focal loss and recall loss here for convenience.

$$RecallCE = -\sum_{c=1}^{C} \sum_{n:y_i=c} (1 - \mathcal{R}_{c,t}) \log(p_{n,t})$$

$$FocalCE = -\sum_{c=1}^{C} \sum_{n:y_n=c} (1 - p_{n,t})^{\gamma} \log(p_{n,t})$$

where y_n and $p_{n,t}$ denote the label of sample n and predicted probability (confidence) for the label at time t. $R_{c,t}$ denotes the recall of class c at time t. We use $\gamma = 1$ in following discussion. We will compare the average focal weight, $\frac{1}{N_c} \sum_{n:y_n=c} (1 - p_{n,t})$, with the recall weight,$1 - \mathcal{R}_{c,t}$, for each class. Because the weights are independently calculated for each class in both losses, the weights do not add up to one. We report *normalized* weights across class instead because the relative percentage of each weight determines the focus of a loss function. The absolute weight only scales the gradient. In fig 4, we report the normalized focal weights, recall weights and recall-focal weight ratio over time during training. We have the following observations and follow-up discussion.

1) In focal loss the average bike class weight increased *relatively* over time whereas the bike class weight percentage decreased in recall loss. This means that the focal weight $\frac{1}{N_c} \sum_{n:y_n=c} (1 - p_{n,t})$ for the bike class did not decrease as fast as other classes. In other words, the confidence for the bike class did not increase relatively to others. The recall-focal weight ratio plot tells the same story. 2) We observe that the ratios of large classes are mostly large than one and increasing while ratios of small classes are below one and constant. This means that focal loss tends to assign lower and decreasing weights, which means higher and increasing confidence, to large classes. Colloquially, focal loss finds it easier to increase the confidence of large classes to reduce loss than increase the confidence of small classes. This limits its ability to correct wrong small class predictions. However,

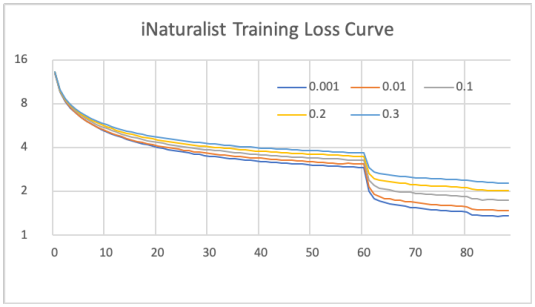

Figure 5: iNaturalist Training Loss with different α values. We observe that smaller α decreases training loss further.

because recall loss uses the metric recall instead of predicted probability as the weights, there is no benefit to continue to increase the confidence of a sample once it is already a true positive. For recall loss, the only way to further reduce loss is by encouraging positive classifications in small classes. The decreasing recall weight for the bike class in fig. 4 shows that the performance of the bike class was increasing over time.

A.2 Implementation

We use DeepLabV3 (Chen et al., 2017) with resnet-$\{18, 101\}$ (He et al., 2016) backbones for semantic segmentation experiments. On Synthia, images are resized to 768 by 384. The resnet models are trained for 100 thousand iterations. On Cityscapes, images are resized to 769 by 769, and models are trained for 90 thousand iterations following Cordts et al. (2016). We use the Adam optimizer (Kingma & Ba, 2014) with a learning rate of 10^{-3} and 10^{-4} without annealing respectively. Note that better performance can be achieved with larger batch size and using SGD optimizer with a learning schedule. However, we focus on comparing relative performance of different loss functions under the same training hyperparameters for segmentation. On image classification, we follow the setting on previous works (Kang et al., 2020) and use resnet-152 for Place-LT and resnet-50 for inaturalist2018. Specifically for Place-LT, we use SGD with a initial learning rate 0.005; for inaturalist2018, we use SGD with an initial learning rate 0.01. A batch size of 128 is used for both datasets.

A.3 Training Curve for iNaturalist

The iNaturalist dataset has 8142 number of classes. This extreme large number of classes poses challenges to recall estimation. We proposed to use Exponential Moving Avergage (EMA) in Sec. 3.3 in the main paper. In addition to the validation accuracy with different α values reported in the main paper. We also present the training curves for the corresponding α. We observe that smaller α value yields consistently lower training loss. This alse shows the effectiveness of using EMA to estimate recall.

B Related Work

The paper(Khan et al., 2017) proposes to iteratively optimize both the model parameters and also a cost-sensitive layer which is integrated into the cost function. Specifically in the proposed cost-sensitive cross entropy loss, the cost sensitive layer does not affect gradient backpropagation. It only affects the output. The loss for training the cost-sensitive layer depends on a separability matrix, a recall confusion matrix and a histogram matrix that encodes the probability of each class. We use recall to regularize cross-entropy loss directly, resulting in a simple yet effective loss function with minimum need for tuning and faster training compared to the iterative optimization procedures in this paper.

The rectification loss paper (Dong et al., 2018) proposed 3 types of rectification losses with 2 variants in each. Therefore, there are in total 6 losses. The best performing one is based on a triplet ranking

loss which enforces inter-class and intra-class constraints. The triplet ranking loss uses hard-positive mining and hard-negative mining to construct triplets for training. The final loss is a weighted sum of the rectification loss and conventional cross-entropy. The weight is determined through cross-validation.

The uncertainty-sensitive loss (Khan et al., 2019) assumes that rare class is more uncertain. It first obtains uncertainty measurement from MCDO and then formulates a max-marge loss to enforce larger margin for higher uncertain classes. They also model sample-level uncertainty using a multi-variate Gaussian distribution and incorporate it in the max-margin loss.

Affinity loss (Hayat et al., 2019) is a multi-tasking max-margin loss where the loss learns clustering and classification. The overall idea is to reduce intra-class variations and maximizing inter-class distances. The loss learns multiple feature prototypes for each class and enforces margin between them.