This ICCV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Anchor Loss: Modulating Loss Scale based on Prediction Difficulty

Serim Ryou
California Institute of Technology

Seong-Gyun Jeong
CODE42.ai

Pietro Perona
California Institute of Technology

Abstract

We propose a novel loss function that dynamically rescales the cross entropy based on prediction difficulty regarding a sample. Deep neural network architectures in image classification tasks struggle to disambiguate visually similar objects. Likewise, in human pose estimation symmetric body parts often confuse the network with assigning indiscriminative scores to them. This is due to the output prediction, in which only the highest confidence label is selected without taking into consideration a measure of uncertainty. In this work, we define the prediction difficulty as a relative property coming from the confidence score gap between positive and negative labels. More precisely, the proposed loss function penalizes the network to avoid the score of a false prediction being significant. To demonstrate the efficacy of our loss function, we evaluate it on two different domains: image classification and human pose estimation. We find improvements in both applications by achieving higher accuracy compared to the baseline methods.

1. Introduction

In many computer vision tasks, deep neural networks produce bi-modal prediction scores when the labeled sample point is confused with the other class. Figure 1 illustrates some examples of network predictions with the presence of visually confusing cases. In all cases, though the network produces a non-trivial score about the correct label, the output prediction is wrong by taking the highest confidence label. For examples, human body parts are mostly composed of symmetric pairs. Even advanced deep architectures [19, 34] are vulnerable to mistaking subtle differences of the left-and-right body parts [39]. Also, in image recognition, the output label confusion of look-alike instances is an unsolved problem [21]. Nevertheless, these tasks employ straightforward loss functions to optimize model parameters, *e.g.*, mean squared error or cross entropy.

In practice, look-alike instances incur an ambiguity in prediction scores, but it is hard to capture subtle differences in the network outputs by measuring the divergence of true

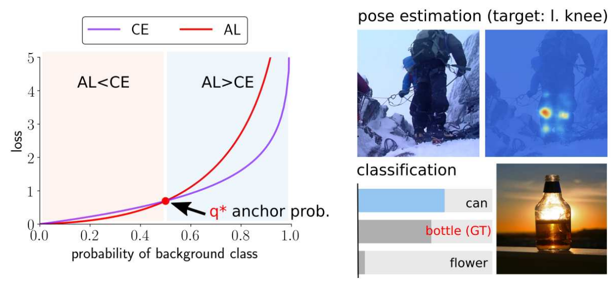
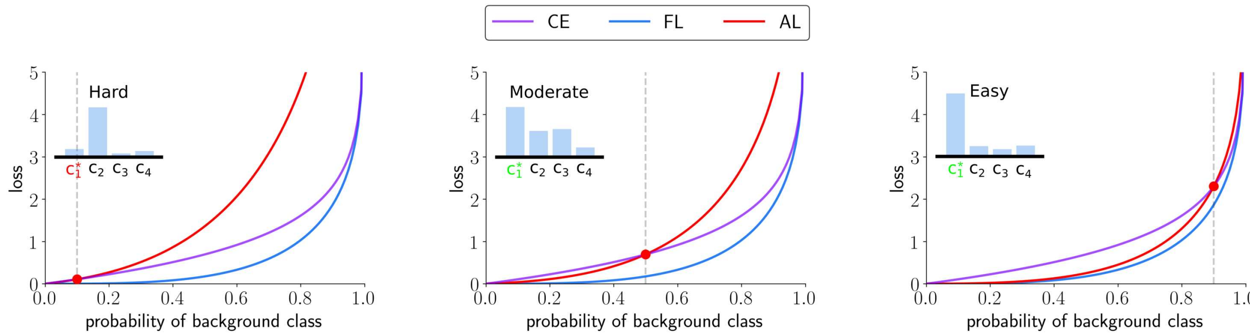


Figure 1. The overview of anchor loss. A network is confused about left-and-right body parts due to the symmetrical appearance of the human body, and struggles to disambiguate visually similar objects. Although the network output scores on the correct labels are relatively high, the final prediction is always chosen by the index of the highest score, resulting in a wrong prediction. Our loss function is designed to resolve this issue by penalizing more than cross entropy when the non-target (background) probability is higher than the anchor probability.

and predicted distributions. Most classification tasks afterward make a final decision by choosing a label with the highest confidence score. We see that the relative score from the output distribution becomes an informative cue to resolve the confusion regarding the final prediction. We thus propose a novel loss function, which self-regulates its scale based on the relative difficulty of the prediction.

We introduce *anchor loss* that adaptively reshapes the loss values using the network outputs. Specifically, the proposed loss function evaluates the prediction difficulties using the relative confidence gap between the target and background output scores, produced by the network, to capture the uncertainty. In other words, we increase the loss for hard samples (Figure 2a), while we down-weight the loss when a sample leads the network to assign a relatively high confidence score about the target class (Figure 2c). Finally, the anchor loss alleviates the need for a post-processing step by taking the prediction difficulty into account while training.

This idea, adjusting the loss scales based on prediction difficulty, has been applied to the task of object detection, which inherently suffers from severe class imbalance issue (countless background vs. scarce object proposals). Focal loss [31] is designed to overcome such class imbalance by avoiding major gradient updates on trivial predictions.



(a) $q_* = 0.1$ (b) $q_* = 0.5$ (c) $q_* = 0.9$

Figure 2. We depict how the anchor probability q_* affects our loss function compared to standard cross entropy (CE) and focal loss (FL) [31]. While FL always depresses the loss values for the samples producing trivial outcomes, anchor loss dynamically re-scales its loss values based on the relative difficulties of the target and the anchor probability. For these plots, the anchor probability is chosen as the prediction score ($q_* = q_{C_1}$) on the true positive label (C_1). Thus, if the networks produce higher score on the background label compared to the anchor, our loss encourages the network to correct the relative order of the predictions by penalizing more than the cross entropy.

However, while the focal loss uniformly down-weights easy samples to ignore, the proposed loss function leverages the confidence gap between the target and non-target output values to modulate the loss scale of the samples in the training phase. We define the prediction difficulty using a reference value which we call *anchor probability* q_* obtained from the network predictions. The way to pick an anchor probability becomes a design choice. One way to use it is by taking the target prediction score as an anchor probability to modulate the background (non-target) loss values. As depicted in Figure 2, the proposed loss function varies based on the anchor probabilities q_*.

We propose anchor loss for improving the prediction of networks on the most semantically confusing cases at training time. Specifically, the proposed anchor loss dynamically controls its magnitude based on prediction difficulty, defined from the network outputs. We observe that our loss function encourages the separation gap between the true labeled score and the most competitive hypothesis. Our main contributions are: (i) the formulation of a novel loss function (anchor loss) for the task of image classification (Section 3.2); (ii) the adaptation of this loss function to human pose estimation (Section 3.3); and (iii) a graphical interpretation about the behavior of the anchor loss function compared to other losses (Figure 2 and 4). With extensive experiments, we show consistent improvements using anchor loss in terms of accuracy for image classification and human pose estimation tasks.

2. Related Work

Class Imbalance Issue. Image classification task suffers class imbalance issue from the long-tail distribution of real-world image datasets. Typical strategies to mitigate this issue are class re-sampling [8, 18, 6] or cost-sensitive learning [50, 23, 14]. Class re-sampling methods [8, 6] redistribute the training data by oversampling the minority class or undersampling the majority class data. Cost-sensitive learning [23, 14] adjusts the loss value by assigning more weights on the misclassified minority classes. Above mentioned prior methods mainly focus on compensating scarce data by innate statistics of the dataset. On the other hand, our loss function renders prediction difficulties from network outputs without requiring prior knowledge about the data distributions.

Relative Property in Prediction. Several researchers attempt to separate confidence scores of the foreground and background classes for the robustness [17, 47]. Pairwise ranking [17] has been successfully adopted in the multi-label image classification task, but efficient sampling becomes an issue when the vocabulary size increases. From the idea of employing a margin constraint between classes, L-softmax loss [33] combines the last fully-connected layer, softmax, and the cross entropy loss to encourage intra-class compactness and inter-class separability in the feature space. While we do not regularize the ordinality of the outputs, our loss function implicitly embodies the concept of ranking. In other words, the proposed loss function rules out a reversed prediction about target and background classes with re-scaling loss values.

Outliers Removal vs. Hard Negative Mining. Studies about robust estimation [24, 48], try to reduce the contribution on model parameter optimization from anomaly samples. Specifically, noise-robust losses [20, 49, 38] have been introduced to support the model training even in the presence of the noise in annotations. Berrada *et al*. [5] address the label confusion problem in the image classification task, such as incorrect annotation or multiple categories present in a single image, and propose a smooth loss function for top-k classification. Deep regression approaches [2, 3] reduce the impact of outliers by minimizing M-estimator with various robust penalties as a loss function. Barron [2] proposed a generalization of common robust loss functions with a single continuous-valued robustness parameter, where the loss function is interpreted as a probability distribution to adapt the robustness.

On the contrary, there have been many studies with an opposite view in various domains, by handling the loss contribution from hard examples as a significant learning signal. Hard negative mining, originally called *Bootstrapping* [41], follows an iterative bootstrapping procedure by selecting background examples for which the detector triggers a false alarm. Online hard example mining (OHEM) [40] successfully adopts this idea to train deep ConvNet detectors in the object detection task. Pose estimation community also explored re-distributing gradient update based on the sample difficulty. Online Hard Keypoint Mining (OHKM) [10] re-weights the loss by sampling few keypoint heatmaps which have high loss contribution, and the gradient is propagated only through the selected heatmaps. Our work has a similar viewpoint to the latter works to put more emphasis on the hard examples.

Focal Loss. One-stage object detection task has an inherent class imbalance issue due to a huge gap between the number of proposals and the number of boxes containing real objects. To resolve this extreme class imbalance issue, some works perform sampling hard examples while training [40, 15, 32], or design a loss function [31] to reshape loss by down-weighting the easy examples. Focal loss [31] also addresses the importance of learning signal from hard examples in the one-stage object detection task. Without sampling processes, focal loss efficiently rescales the loss function and prevents the gradient update from being overwhelmed by the easy-negatives. Our work is motivated by the mathematical formulation of focal loss [31], where predefined modulating term increases the importance of correcting hard examples.

Human Pose Estimation. Human pose estimation is a problem of localizing human body part locations in an input image. Most of the current works [34, 10, 45, 46, 28, 42] use a deep convolutional neural network and generate the output as a 2D heatmap, which is encoded as a gaussian map centered at each body part location. Hourglass network [34] exploits the iterative refinements on the predictions from the repeated encoder-decoder architecture design to capture complex spatial relationships. Even with deep architectures, disambiguating look-alike body parts remain as a main problem [39] in pose estimation community. Recent methods [46, 11, 28], built on top of the hourglass network, use multi-scale and body part structure information to improve the performance by adding more architectural components.

While there has been much interest in finding a good architecture tailored to the pose estimation problem, the vast majority of papers simply use mean squared error (MSE), which computes the L2 distance between the output and the prediction heatmap, as a loss function for this task. OHKM [10], which updates the gradient from the selected set of keypoint heatmaps, improves the performance when properly used in the refinement step. On the other hand, we propose a loss scaling scheme that efficiently redistributes the loss values without sampling hard examples.

3. Method

In this section, we introduce *anchor loss* and explain the design choices for image classification and pose estimation tasks. First, we define the prediction difficulty and provide related examples. We then present the generalized form of the anchor loss function. We tailor our loss function on visual understanding tasks: image classification and human pose estimation. Finally, we give theoretical insight in comparison to other loss functions.

3.1. Anchor Loss

The inference step for most classification tasks chooses the label index corresponding to the highest probability. Figure 1 shows sample outputs from the model trained with cross entropy. Although optimizing the networks with the cross entropy encourages the predicted distribution to resemble the true distribution, it does not convey the relative property between the predictions on each class.

Anchor loss function dynamically reweighs the loss value with respect to prediction difficulty. The prediction difficulty is determined by measuring the divergence between the probabilities of the true and false predictions. Here the anchor probability q_* becomes a reference value for determining the prediction difficulty. The definition of anchor probability q_* is arbitrary and becomes a design choice. However, in practice, we observed that setting anchor probability to the target class prediction score gives the best performance, so we use it for the rest of the paper. With consideration of the prediction difficulties, we formulate the loss function as follows:

$$\ell(p, q; \gamma) = -\underbrace{(1 + \overbrace{q - q_*}^{\text{prediction difficulty}})^{\gamma}}_{\text{modulator}} \underbrace{(1-p)\log(1-q)}_{\text{cross entropy}}, \tag{1}$$

where p and q denote empirical label and predicted probabilities, respectively. The anchor probability q_* is determined by the primitive logits, where the anchor is the prediction score on the true positive label. Here, $\gamma \geq 0$ is a hyperparameter that controls the dynamic range of the loss function. Our loss is separable into two parts: modulator and cross entropy. The modulator is a monotonic increasing function that takes relative prediction difficulties into account, where the domain is bounded by $|q - q_*| < 1$. Suppose q_* be the target class prediction score. In an easy prediction scenario, the network assigns a correct label for the given sample point; hence q_* will be larger than any q. We illustrate the prediction difficulties as follows:

- **Easy case** ($q < q_*$): the loss function is suppressed, and thus rules out less informative samples when updating the model;
- **Moderate case** ($q = q_*$): the loss function is equivalent to cross entropy, since the modulator becomes 1; and
- **Hard case** ($q > q_*$): the loss function penalizes more than cross entropy for most of the range, since the true positive probability q_* is low.

As a result, we apply different loss functions for each sample.

3.2. Classification

For image classification, we adopt sigmoid-binary cross entropy as a basic setup to diversify the way of scaling loss values. Unlike softmax, sigmoid activation handles each class output probability as an independent variable, where each label represents whether the image contains an object of corresponding class or not. This formulation also enables our loss function to capture subtle differences from the output space by modulating the loss values on each label.

For image classification, we obtained the best performance when we set the anchor probability to the output score of the target class. The mathematical formulation becomes as follows:

$$\begin{aligned} &\ell_{cls}(p, q; \gamma) \qquad (2) \\ &= -\sum_{k=1}^{K} p_k \log q_k + (1 - p_k)(1 + q_k - q_*)^{\gamma} \log(1 - q_k), \end{aligned}$$

where p_k and q_k represent the empirical label and the predicted probability for class k. We add a margin variable δ to anchor probability q_* to penalize the output variables which have lower but close to the true positive prediction score. Thus the final anchor probability becomes $q_* = q_t - \delta$, where t represents the target index ($p_t = 1$), and we set δ to 0.05.

(a) input (b) heatmap (c) mask

Figure 3. How an anchor probability is chosen for the pose estimation task. For the target body part of right shoulder (b), the maximum confidence score inside the solid red circle becomes an anchor probability to modulate the loss values in mask areas (c).

3.3. Pose Estimation

Current pose estimation methods generate a keypoint heatmap for each body part at the end of the prediction stage, and predict the pixel location that has the highest probability. The main difference of pose estimation and object classification tasks is that the target has spatial dependency between adjacent pixel locations. As a result, assigning a single pixel as the true positive may incur a huge penalty on adjacent pixels. To alleviate this issue, we adopt a gaussian heatmap centered on the target keypoint as the same encoding scheme as the previous works [34, 45, 10], and apply our loss function on only true negative pixels ($p_i = 0$). In other words, we use a mask variable $M(p)$ to designate the pixel locations where our loss function applies, and use standard binary cross entropy on unmasked locations.

$$M(p) = \begin{cases} 1 & \text{if } p = 0, \\ 0 & \text{otherwise.} \end{cases} \qquad (3)$$

As in object classification, we found that using true-positive probability value to penalize background pixel locations gives better performance. Considering the spatial dependency, anchor probabilities are chosen spatially from the circle of high confidence, where the ground truth probability is greater than 0.5. That is,

$$q_* = \max_{i \forall p_i > 0.5} q_i, \qquad (4)$$

We illustrate this procedure in Figure 3. For simplicity, we denote the standard binary cross entropy as ℓ_{BCE}. Finally, our loss function for pose estimation problem is defined as:

$$\begin{aligned} \ell_{pose}(p, q; \gamma) = &[M(p) * (1 + q - q_*)^{\gamma} \qquad (5) \\ &+ (1 - M(p))] * \ell_{BCE}(p, q), \end{aligned}$$

3.4. Relationship to Other Loss Functions

Our goal is to design a loss function which takes the relative property of the inference step into account. In this section, we discuss how binary cross entropy (6) and focal loss [31] (7) relate to anchor loss. Let $p \in \{0, 1\}$ denote the ground truth, and $q \in [0, 1]$ represent predicted distribution. The loss functions are

$$\ell_{CE}(p, q) = -\big[p \log(q) + (1-p)\log(1-q)\big], \tag{6}$$

$$\ell_{FL}(p, q; \gamma) = -\big[p(1-q)^{\gamma} \log(q) + (1-p)q^{\gamma}\log(1-q)\big], \tag{7}$$

For the sake of conciseness, we define the probability of ground truth as $q_t = pq + (1-p)(1-q)$. Then we replace the loss functions as follows:

$$\ell_{CE}(q_t) = -\log(q_t), \tag{8}$$

$$\ell_{FL}(q_t; \gamma) = -(1-q_t)^{\gamma}\log(q_t), \tag{9}$$

where q represents the output vector from the network. The modulating factor $(1-q_t)^{\gamma}$ with focusing parameter γ reshapes the loss function to down-weight easy samples. Focal loss was introduced to resolve the extreme class imbalance issue in object detection, where the majority of the loss is comprised of easily classified background examples. Object detection requires the absolute threshold value to decide the candidate box is foreground or background. On the other hand, classification requires the confidence score of the ground truth label to be higher than all other label scores.

If we set $q_* = 1 - p$, which means $q_* = 1$ for the background classes and $q_* = 0$ for the target class:

$$q_* = \begin{cases} 1 & p = 0 \quad \text{background classes,} \\ 0 & p = 1 \quad \text{target class,} \end{cases} \tag{10}$$

then the modulator becomes:

$$(1 - q_t + q_*) = \begin{cases} (1-(1-q)+1) = (1-q) & p = 0, \\ (1-q+0) = q & p = 1, \end{cases} \tag{11}$$

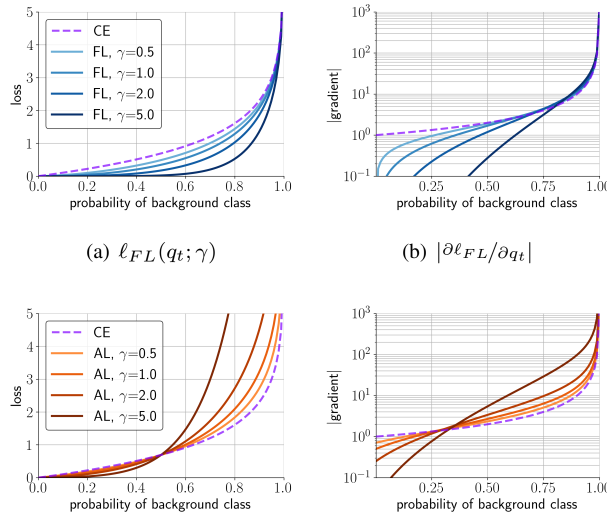
and feeding this modulator value to anchor loss becomes a mathematical formulation of focal loss:

$$\ell_{AL}(p, q; \gamma) = -\big[p(1-q)^{\gamma}\log(q) + (1-p)q^{\gamma}\log(1-q)\big], \quad \text{where } q_* = 1 - p. \tag{12}$$

If we set $\gamma = 0$, the the modulator term becomes 1, and anchor loss becomes binary cross entropy.

3.5. Gradient Analysis

We compute the gradient of our loss function and compare with the binary cross entropy and the focal loss. For simplicity, we focus on the loss of background label, which we discuss in Section 3.1. Note that we detach the anchor probability q_* while backpropagation and only use it as a scaling term in the modulator.



(a) $\ell_{FL}(q_t; \gamma)$ (b) $|\partial\ell_{FL}/\partial q_t|$

(c) $\ell_{AL}(q_t; \gamma), q_* = 0.5$ (d) $|\partial\ell_{AL}/\partial q_t|$

Figure 4. Gradient figure: sample gradient output of background probability distribution. Compared to the cross entropy, the magnitude of gradient increases when the prediction is higher than the anchor probability.

$$\ell_{AL}(q) = -(1 + q - q_*)^{\gamma}\log(1-q) \tag{13}$$

$$\frac{\partial \ell_{AL}}{\partial q}(q) = -(1+q-q_*)^{\gamma-1}\left[\gamma\log(1-q) - \frac{1+q-q_*}{1-q}\right] \tag{14}$$

Figure 4 shows the gradient of our loss function, focal loss, and cross entropy. Compared to the cross entropy, the gradient values of focal loss are suppressed for all ranges. On the other hand, our loss function assigns larger gradient values when the prediction is higher than the anchor probability, and vice versa.

4. Experiments

We conduct experiments on image classification and human pose estimation. In this section, we briefly overview the methods that we use in each domain, and discuss the experimental results.

4.1. Image Classification

Datasets. For the object classification, we evaluate our method on CIFAR-10/100 [29] and ImageNet (ILSVRC 2012) [13]. CIFAR 10 and 100 each consist of 60,000 images with 32×32 size of 50,000 training and 10,000 testing

Table 1. Classification accuracy on CIFAR (ResNet-110)

Loss Fn.	Parameter	CIFAR-10 Top-1	CIFAR-100 Top-1	CIFAR-100 Top-5
CE		93.91 ± 0.12	72.98 ± 0.35	92.55 ± 0.30
BCE		93.69 ± 0.08	73.88 ± 0.22	92.03 ± 0.42
OHEM	$\rho = 0.9, 0.9$	93.90 ± 0.10	73.03 ± 0.29	$\underline{92.61} \pm 0.21$
FL	$\gamma = 2.0, 0.5$	94.05 ± 0.23	74.01 ± 0.04	92.47 ± 0.40
Ours				
AL	$\gamma = 0.5, 0.5$	$\underline{94.10} \pm 0.15$	$\underline{74.25} \pm 0.34$	$\mathbf{92.62 \pm 0.50}$
AL w/ warmup	$\gamma = 0.5, 2.0$	$\mathbf{94.17 \pm 0.13}$	$\mathbf{74.38 \pm 0.45}$	92.45 ± 0.05

Table 2. Classification accuracies on ImageNet (ResNet-50)

Loss Fn.	Parameter	Top-1	Top-5
CE		76.39	93.20
OHEM	$\rho = 0.8$	76.27	93.21
FL	$\gamma = 0.5$	76.72	93.06
AL (ours)	$\gamma = 0.5$	**76.82**	93.03

images. In our experiment, we randomly select 5,000 images for the validation set. CIFAR-10 dataset has 10 labels with 6,000 images per class, and CIFAR-100 dataset has 100 classes each containing 600 images.

Implementation details. For CIFAR, we train ResNet-110 [19] with our loss function and compare with other loss functions and OHEM. We randomly flip and crop the images padded with 4 pixels on each side for data augmentation. All the models are trained with PyTorch [36]. Note that our loss is summed over class variables and averaged over batch. The learning rate is set to 0.1 initially, and dropped by a factor of 0.1 at 160 and 180 epochs respectively. In addition, we train ResNet-50 models on ImageNet using different loss functions. We use 8 GPUs and batch size of 224. To accelerate training, we employ a mixed-precision. We apply minimal data augmentation, *i.e.*, random cropping of 224×224 and horizontal flipping. The learning rate starts from 0.1 and decays 0.1 every 30 epoch. We also perform learning rate warmup strategy for first 5 epochs as proposed in [19].

Results. For CIFAR, we train and test the network three times and report the mean and standard deviation in Table 1. We report top-1 and top-5 accuracy and compare the score with other loss functions and OHEM. OHEM computes the loss values for all samples in a batch, chooses the samples of high loss contribution with a ratio of ρ, and updates the gradient only using those samples. As we can see in the Table 1, our loss function has shown improvements over all loss functions we evaluated. For CIFAR 100, performance improved by simply replacing the cross entropy to the binary cross entropy, and anchor loss gives further gain by exploiting the automated re-scaling scheme. With our experimental setting, we found that sampling hard examples (OHEM) does not help. We tried out few different sampling ratio settings, but found performance degradation over all ratios.

Table 3. Ablation studies on CIFAR-100 (ResNet-110)

		Top-1	Top-5
Static anchor probabilities			
$\gamma = 0.5$	$q_* = 0.8$	73.74	92.45
$\gamma = 0.5$	$q_* = 0.5$	73.77	92.30
$\gamma = 0.5$	$q_* = 0.1$	73.11	92.08
Dynamic anchor probabilities			
$\gamma = 0.5$	-	**74.25**	**92.62**
$\gamma = 1.0$	-	73.59	92.04
$\gamma = 2.0$	-	71.86	91.46

Ablation Studies. As an ablation study, we report the top-1 and top-5 accuracy on CIFAR-100 by varying the γ in Table 3. For classification task, low γ yielded a good performance. We also perform experiments with fixed anchor probabilities to see how the automated sample difficulty from the network helps training. The results in Table 3 show that using the network output to define sample difficulty and rescale the loss based on this value helps the network keep a good learning signal.

CE warmup strategy. To accelerate and stabilize the training process, we use CE for first few epochs and then replace loss function to AL. We tested CE warmup on CIFAR-100 for the first 5 epochs (Figure 5). With the warmup strategy, the ratio of hard samples was decreased; in other words, loss function less fluctuated. As a result, we achieved the highest top-1 accuracy of 74.38% (averaged out multiple runs) regardless of a high $\gamma = 2$ value.

4.2. Human Pose Estimation

We evaluate our method on two different human pose estimation datasets: single-person pose on MPII [1] and LSP [26] dataset. The single-person pose estimation problem assumes that the position and the scale information of a target person are given.

Implementation details. For the task of human pose estimation, we use the Hourglass network [34] as a baseline and only replace the loss function with the proposed loss during training. Note that we put sigmoid activation layer on top of the standard architecture to perform classification. Pose models are trained using Torch [12] framework. The input size is set to 256×256, batch size is 6, and the model is trained with a single NVIDIA Tesla V100 GPU. Learning rate is set to 0.001 for the first 100 epochs and dropped by half and 0.2 iteratively at every 20 epoch. Testing is held by averaging the heatmaps over six-scale image pyramid with flipping.

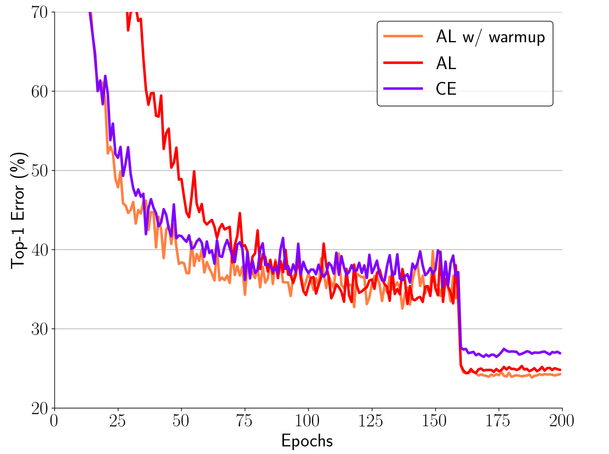


Figure 5. Validation curves of ResNet-110 on CIFAR-100 dataset. We compare our loss function to CE.

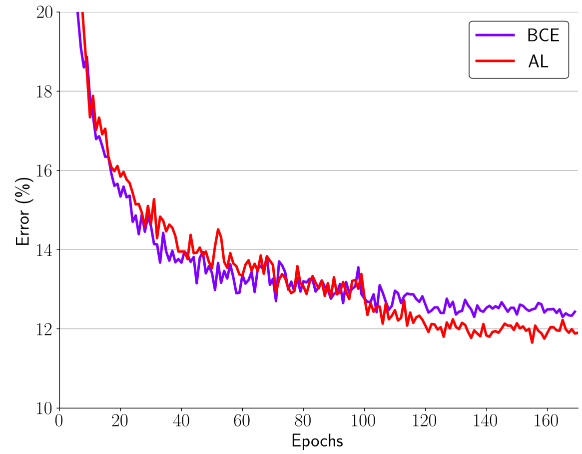


Figure 6. Validation curves of 2-stacked Hourglass on MPII dataset. We compare our loss function to BCE.

Datasets. The MPII human pose dataset consists of 20k training images over 40k people performing various activities. We follow the previous training/validation split from [43], where 3k images from training set are used for validation. The LSP dataset [26] is composed of 11k training images with LSP extended dataset [27], and containing mostly sports activities.

Results. We evaluate the single-person pose estimation results on standard Percentage of Correct Keypoints (PCK) metric, which defines correct prediction if the distance between the output and the ground truth position lies in α with respect to the scale of the person. α is set to 0.5 and 0.2 in MPII and LSP dataset, respectively. PCK score for each dataset is reported in Table 4 and 5.

For comparison, we split the performance table by hourglass-based architecture. The bottom rows are comparison between the methods built on top of Hourglass network. We achieve comparable results to the models built on top of hourglass network with more computational complexity on both datasets. We also report the validation score of the baseline method trained with mean squared error by conducting a single scale test for direct comparison between the losses in Table 6. We found consistent improvements over the symmetric parts; Due to appearance similarity on the symmetric body parts, our loss function automatically penalizes more on those parts during training, without having any additional constraint for the symmetric parts.

Ablation Studies. We conduct ablation studies by varying γ on 2-stacked hourglass network and report the score in Table 7. With proper selection of $\gamma = 2.0$, we can achieve better performance over all the losses.

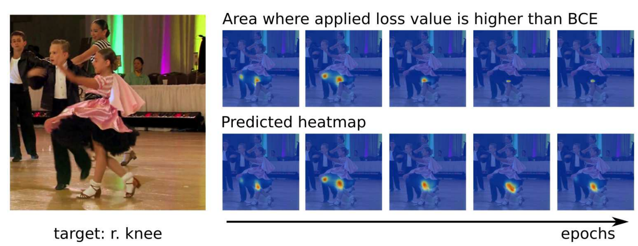


Figure 7. We visualize where anchor loss assigns higher loss values than the binary cross entropy and how it changes over training epochs. At the beginning, visually similar parts often get higher scores than the target body part, thus our loss function assigns higher weights on those pixel locations. Once the model is able to detect the target body part with high confidence, loss is down-weighted for most of the areas, so that the network can focus on finding more accurate location for the target body part.

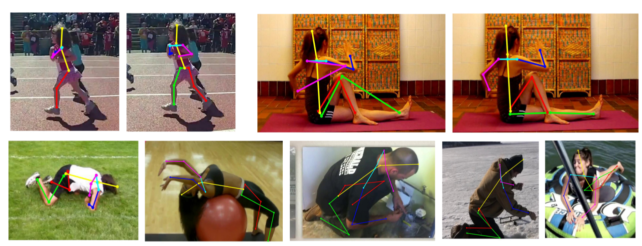


Figure 8. Qualitative results on human pose. The first row compares with the result from MSE loss (left) and our loss (right), and the second row contains some sample outputs. Model trained with the proposed loss function is robust at predicting symmetric body parts.

Qualitative Analysis. We visualize which area gets more penalty than the standard binary cross entropy in Fig 7. For the fist few epochs, we can see that visually similar parts of both target and non-target person get higher penalty. Once the model finds the correct body part locations, the loss function is down-weighted and the area of higher penalty

Table 4. PCK score on MPII dataset. The bottom rows show the performances of the methods built on top of hourglass network. The model trained with anchor loss shows comparative scores to the results from more complex models.

Method	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	Total
Tompson *et al.* [43]	96.1	91.9	83.9	77.8	80.9	72.3	64.8	82.0
Hu & Ramanan [22]	95.0	91.6	83.0	76.6	81.9	74.5	69.5	82.4
Pishchulin *et al.* [37]	94.1	90.2	83.4	77.3	82.6	75.7	68.6	82.4
Lifshitz *et al.* [30]	97.8	93.3	85.7	80.4	85.3	76.6	70.2	85.0
Gkioxary *et al.* [16]	96.2	93.1	86.7	82.1	85.2	81.4	74.1	86.1
Rafi *et al.* [44]	97.2	93.9	86.4	81.3	86.8	80.6	73.4	86.3
Insafutdinov *et al.* [25]	96.8	95.2	89.3	84.4	88.4	83.4	78.0	88.5
Belagiannis & Zisserman [4]	97.7	95.0	88.2	83.0	87.9	82.6	78.4	88.1
Wei *et al.* [45]	97.8	95.0	88.7	84.0	88.4	82.8	79.4	88.5
Bulat & Tzimiropoulos [7]	97.9	95.1	89.9	85.3	89.4	85.7	81.7	89.7
Ning *et al.* [35]	98.1	96.3	92.2	87.8	90.6	87.6	82.7	91.2
Tang *et al.* [42]	98.4	**96.9**	92.6	**88.7**	**91.8**	89.4	86.2	**92.3**
Hourglass model variants								
Chu *et al.* [11]	98.5	96.3	91.9	88.1	90.6	88.0	85.0	91.5
Chen *et al.* [9]	98.1	96.5	92.5	88.5	90.2	**89.6**	86.0	91.9
Yang *et al.* [46]	98.5	96.7	92.5	**88.7**	91.1	88.6	86.0	92.0
Ke *et al.* [28]	98.5	96.8	**92.7**	88.4	90.6	89.3	**86.3**	92.1
Hourglass + MSE [34]	98.2	96.3	91.2	87.1	90.1	87.4	83.6	90.9
Hourglass + AL (Ours)	**98.6**	96.6	92.3	87.8	90.8	88.8	86.0	91.9

Table 5. PCK score on LSP dataset. The bottom rows show the performances of the methods built on top of hourglass network. We achieve better performance on LSP dataset without adding the complexity, by training the network with anchor loss. For comparison, we also report the state-of-the-art score on the top row.

Method	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	Total
Lifshitz *et al.* [30]	96.8	89.0	82.7	79.1	90.9	86.0	82.5	86.7
Pishchulin *et al.* [37]	97.0	91.0	83.8	78.1	91.0	86.7	82.0	87.1
Insafutdinov *et al.* [25]	97.4	92.7	87.5	84.4	91.5	89.9	87.2	90.1
Wei *et al.* [45]	97.8	92.5	87.0	83.9	91.5	90.8	89.9	90.5
Bulat&Tzimiropoulos [7]	97.2	92.1	88.1	85.2	92.2	91.4	88.7	90.7
Ning *et al.* [35]	98.2	94.4	91.8	89.3	94.7	95.0	93.5	93.9
Tang *et al.* [42]	98.3	**95.9**	**93.5**	**90.7**	**95.0**	**96.6**	**95.7**	**95.1**
Hourglass model variants								
Chu *et al.* [11]	98.1	93.7	89.3	86.9	93.4	94.0	92.5	92.6
Yang *et al.* [46]	98.3	94.5	92.2	88.9	94.4	95.0	93.7	93.9
Hourglass + AL (Ours)	**98.6**	94.8	92.5	89.3	93.9	94.8	94.0	94.0

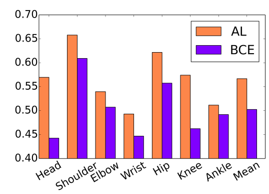
Table 6. Validation Results on MPII dataset. We report the validation score of the result using different losses with the same single-scale testing setup.

Method	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	Mean
Hourglass + MSE	**96.73**	95.94	90.39	85.40	89.04	85.17	81.86	89.32
Hourglass + AL (Ours)	96.45	**96.04**	**90.46**	**86.00**	**89.20**	**86.84**	**83.68**	**89.93**

Table 7. Hyperparameter search and comparison to other losses on MPII dataset with 2-stacked hourglass network.

Method	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	Mean
BCE	96.42	95.35	89.82	84.72	88.47	85.17	81.13	88.84
MSE	96.42	95.30	89.57	84.63	88.78	85.07	**81.77**	88.89
FL	**96.52**	**95.47**	89.71	84.87	88.38	84.75	81.25	88.81
AL, $\gamma = 5$	96.35	95.04	89.26	84.56	**88.99**	**85.51**	81.37	88.84
AL, $\gamma = 1$	96.35	95.40	89.60	85.11	88.59	84.85	**81.77**	88.94
AL, $\gamma = 2$	96.49	95.45	**90.08**	**85.42**	88.64	85.31	81.60	**89.11**

is focused only on few pixel locations, which helps fine adjustments on finding more accurate locations. We also show some sample outputs in Fig 8. For comparison, the top row shows some outputs from the model trained with MSE (left) and anchor loss (right). We can see that the network trained with proposed loss is robust at predicting symmetric parts.



Double-counting. For the task of human pose estimation, we observe a double-counting problem, where the predicted heatmap shows multiple peaks. To analyze how AL behaves in those cases, we depict the ratio of the correct prediction when double-counting problems are encountered on MPII dataset. Overall, AL assigns correct body parts compared to BCE.

5. Conclusion

In this paper, we presented anchor loss function which adaptively rescales the standard cross entropy function based on prediction difficulty. The network automatically evaluates the prediction difficulty by measuring the divergence among the network outputs regarding true positive and false positive predictions. The proposed loss function has shown strong empirical results on two different domains: image classification and human pose estimation. A simple drop-in replacement for standard cross entropy loss gives performance improvement. With a proper selection of designing the re-weighing scheme and anchor probability, the anchor loss can be applied to diverse machine learning and computer vision applications.

Acknowledgement We would like to thank Joseph Marino and Matteo Ruggero Ronchi for their valuable comments. This work was supported by funding from Disney Research.

References

[1] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2D human pose estimation: New benchmark and state of the art analysis. In *Proc. IEEE CVPR*, 2014.

[2] Jonathan T. Barron. A general and adaptive robust loss function. In *Proc. IEEE CVPR*, 2019.

[3] Vasileios Belagiannis, Christian Rupprecht, Gustavo Carneiro, and Nassir Navab. Robust optimization for deep regression. In *Proc. IEEE ICCV*, 2015.

[4] Vasileios Belagiannis and Andrew Zisserman. Recurrent human pose estimation. In *Proc. IEEE FG*, 2017.

[5] Leonard Berrada, Andrew Zisserman, and M. Pawan Kumar. Smooth loss functions for deep top-k classification. In *ICLR*, 2018.

[6] Mateusz Buda, Atsuto Maki, and Maciej A. Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249 – 259, 2018.

[7] Adrian Bulat and Georgios Tzimiropoulos. Human pose estimation via convolutional part heatmap regression. In *ECCV*, 2016.

[8] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote: Synthetic minority oversampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

[9] Yu Chen, Chunhua Shen, Xiu-Shen Wei, Lingqiao Liu, and Jian Yang. Adversarial posenet: A structure-aware convolutional network for human pose estimation. In *Proc. IEEE ICCV*, 2017.

[10] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. In *Proc. IEEE CVPR*, 2017.

[11] Xiao Chu, Wei Yang, Wanli Ouyang, Cheng Ma, Alan L. Yuille, and Xiaogang Wang. Multi-context attention for human pose estimation. In *Proc. IEEE CVPR*, 2017.

[12] R. Collobert, K. Kavukcuoglu, and C. Farabet. Torch7: A matlab-like environment for machine learning. In *BigLearn, NIPS Workshop*, 2011.

[13] Jia Deng, Wei Dong, Richard Socher, Li jia Li, Kai Li, and Fei-Fei Li. ImageNet: A large-scale hierarchical image database. In *Proc. IEEE CVPR*, 2009.

[14] Qi Dong, Shaogang Gong, and Xiatian Zhu. Class rectification hard mining for imbalanced deep learning. In *Proc. IEEE ICCV*, 2017.

[15] Pedro F. Felzenszwalb, Ross B. Girshick, and David A. McAllester. Cascade object detection with deformable part models. In *Proc. IEEE CVPR*, 2010.

[16] Georgia Gkioxari, Alexander Toshev, and Navdeep Jaitly. Chained predictions using convolutional neural networks. In *ECCV*, 2016.

[17] Yunchao Gong, Yangqing Jia, Thomas K. Leung, Alexander Toshev, and Sergey Ioffe. deep convolutional ranking for multi label image annotation. In *ICLR*, 2014.

[18] Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. In *ICIC*, 2005.

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

[20] Dan Hendrycks, Mantas Mazeika, Duncan Wilson, and Kevin Gimpel. Using trusted data to train deep networks on labels corrupted by severe noise. In *NuerIPS*, 2018.

[21] Derek Hoiem, Yodsawalai Chodpathumwan, and Qieyun Dai. Diagnosing error in object detectors. In *ECCV*, 2012.

[22] Peiyun Hu and Deva Ramanan. Bottom-up and top-down reasoning with hierarchical rectified gaussians. In *Proc. IEEE CVPR*, 2016.

[23] Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. Learning deep representation for imbalanced classification. In *Proc. IEEE CVPR*, 2016.

[24] Peter J. Huber. Robust estimation of a location parameter. *Annals of Mathematical Statistics*, 35(1):73–101, Mar. 1964.

[25] Eldar Insafutdinov, Leonid Pishchulin, Bjoern Andres, Mykhaylo Andriluka, and Bernt Schiele. DeeperCut: A deeper, stronger, and faster multi-person pose estimation model. In *ECCV*, 2016.

[26] Sam Johnson and Mark Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *BMVC*, 2010.

[27] Sam Johnson and Mark Everingham. Learning effective human pose estimation from inaccurate annotation. In *Proc. IEEE CVPR*, 2011.

[28] Lipeng Ke, Ming-Ching Chang, Honggang Qi, and Siwei Lyu. Multi-scale structure-aware network for human pose estimation. In *ECCV*, 2018.

[29] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

[30] Ita Lifshitz, Ethan Fetaya, and Shimon Ullman. Human pose estimation using deep consensus voting. In *ECCV*, 2016.

[31] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proc. IEEE ICCV*, 2017.

[32] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: Single shot multibox detector. In *ECCV*, 2015.

[33] Weiyang Liu, Yandong Wen, Zhiding Yu, and Meng Yang. Large-margin softmax loss for convolutional neural networks. In *ICML*, 2016.

[34] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, 2016.

[35] Guanghan Ning, Zhi Zhang, and Zhiquan He. Knowledge-guided deep fractal neural networks for human pose estimation. *IEEE Trans. Multimedia*, 20(5):1246–1259, 2018.

[36] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS-Workshops*, 2017.

[37] Leonid Pishchulin, Eldar Insafutdinov, Siyu Tang, Bjoern Andres, Mykhaylo Andriluka, Peter Gehler, and Bernt Schiele. DeepCut: Joint subset partition and labeling for multi person pose estimation. In *Proc. IEEE CVPR*, 2016.

[38] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *ICML*, 2018.

[39] Matteo Ruggero Ronchi and Pietro Perona. Benchmarking and error diagnosis in multi-instance pose estimation. In *Proc. IEEE ICCV*, 2017.

[40] Abhinav Shrivastava, Abhinav Gupta, and Ross B. Girshick. Training region-based object detectors with online hard example mining. In *Proc. IEEE CVPR*, 2016.

[41] Kah Kay Sung. *Learning and Example Selection for Object and Pattern Detection*. PhD thesis, 1996.

[42] Wei Tang, Pei Yu, and Ying Wu. Deeply learned compositional models for human pose estimation. In *ECCV*, 2018.

[43] Jonathan Tompson, Ross Goroshin, Arjun Jain, Yann LeCun, and Christoph Bregler. Efficient object localization using convolutional networks. In *Proc. IEEE CVPR*, 2015.

[44] Juergen Gall Umer Rafi, Bastian Leibe and Ilya Kostrikov. An efficient convolutional network for human pose estimation. In *BMVC*, 2016.

[45] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *Proc. IEEE CVPR*, 2016.

[46] Wei Yang, Shuang Li, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Learning feature pyramids for human pose estimation. In *Proc. IEEE ICCV*, 2017.

[47] Min-Ling Zhang and Zhi-Hua Zhou. Multilabel neural networks with applications to functional genomics and text categorization. *IEEE Trans. Knowl. Data Eng.*, 18(10):1338–1351, Oct. 2006.

[48] Tong Zhang. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *ICML*, 2004.

[49] Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In *NeurIPS*, 2018.

[50] Zhi-Hua Zhou and Xu-Ying Liu. Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Trans. Knowl. Data Eng.*, 18(1):63–77, Feb. 2006.