# Influence-Balanced Loss for Imbalanced Visual Classification

Seulki Park  Jongin Lim  Younghan Jeon  Jin Young Choi
ASRI, Dept. of Electrical and Computer Engineering, Seoul National University
{seulki.park, ljin0429, yh1992, jychoi}@snu.ac.kr

## Abstract

*In this paper, we propose a balancing training method to address problems in imbalanced data learning. To this end, we derive a new loss used in the balancing training phase that alleviates the influence of samples that cause an overfitted decision boundary. The proposed loss efficiently improves the performance of any type of imbalance learning methods. In experiments on multiple benchmark data sets, we demonstrate the validity of our method and reveal that the proposed loss outperforms the state-of-the-art cost-sensitive loss methods. Furthermore, since our loss is not restricted to a specific task, model, or training method, it can be easily used in combination with other recent re-sampling, meta-learning, and cost-sensitive learning methods for class-imbalance problems. Our code is made available at* https://github.com/pseulki/IB-Loss.

## 1. Introduction

Despite the remarkable success of deep neural networks (DNNs) these days, many areas of computer vision suffer from highly imbalanced datasets. Many real-world data exhibit skewed distributions [23, 16, 11, 24, 10], in which the number of samples per class differs greatly. This imbalance between classes can be problematic, since the model trained on such imbalanced data tends to overfit the dominant (majority) classes [18, 14, 4]. That is, while the overall performance appears to be satisfactory, the model performs poorly on minority classes. To overcome the class imbalance problem, extensive research has recently been conducted to improve the generalization performance by reducing the overwhelming influence of the dominant class on the model.

The research on imbalanced learning can be divided into three approaches: data-level approach, cost-sensitive re-weighting approach, and meta-learning approach. The data-level approach aims to directly balance the training data distributions via re-sampling (i.e., under-sampling or over-sampling) [6, 32] or by generating synthetic samples [28]. Meanwhile, the cost-sensitive re-weighting approach aims to design new loss functions to re-weight samples by considering their importance [33, 17, 22]. Finally, the meta-learning approach enhances the performance of the data-level and/or cost-sensitive re-weighting approach via meta-learning [31, 25, 30]. Most recent data-level approaches require a heavy computational burden. Moreover, under-sampling can lose some valuable information, and over-sampling or data generation can cause overfitting on certain repetitive samples. The meta-learning approach requires additional unbiased data [31] or a meta-sampler [30], which is computationally expensive in practice. Therefore, our work focuses on the cost-sensitive re-weighting approach to design a new loss function that is simple but efficient.

The cost-sensitive re-weighting approach aims to assign class penalties to shift the decision boundary in a way that reduces the bias induced by the data imbalance. For this purpose, the most commonly adopted method is to re-weight samples inversely to the number of training samples in each class to assign more weights for the minority classes [17, 33, 8]. These methods have focused on only global-level class distribution and assign the same fixed weight to all samples belonging to the same class. However, not all samples in a dataset play an equal role in determining the model parameters [7]. That is, some samples have greater influences on forming a decision boundary. Hence, each sample needs to be re-weighted differently according to its impact on the model.

Recently, numerous studies have been conducted in which each sample is considered to design sample-wise loss functions [9, 22, 27]. Specifically, these methods down-weight well-classified samples and assign more weights to *hard examples*, which yield high errors. This re-weighting might lead to the complete training when the high capacity of DNNs is sufficient to finally memorize the whole training data [34, 3]. This implies that DNN is overfitted to hard samples, which are located at the overlapping region between the majority and minority classes. In the imbalanced data, most hard samples are majority samples that enforce the decision boundary to be complex and shift to the minority region.

To address the aforementioned problem, in this paper, we propose a loss-sensitive method to down-weight samples that cause overfitting of a DNN trained with highly imbalanced data. To this end, we derive a formula that measures how much each sample influences the complex and biased decision boundary. To derive the formula, we utilize the influence function [7], which has been widely used in robust statistics. Using the derived formula, we design a novel loss function, called influence-balanced (IB) loss, that adaptively assigns different weights to samples according to their influence on a decision boundary. Specifically, we re-weight the loss proportionally to the inverse of the influence of each sample. Our method is divided into two phases: standard training and fine-tuning for influence balancing. During the fine-tuning phase, the proposed IB loss alleviates the influence of the samples that cause overfitting of the decision boundary.

Through extensive experiments on multiple benchmark data sets, we demonstrate the validity of our method, and show that the proposed method outperforms the state-of-the-art cost-sensitive re-weighting methods. Furthermore, since our IB loss is not restricted to a specific task, model, or training method, it can be easily utilized in combination with other recent data-level algorithms and hybrid methods for class-imbalance problems.

The main contributions of this paper are as follows:

- We discover that the existing loss-based loss methods can lead a decision boundary of DNNs to eventually overfit to the majority classes.
- We design a novel influence-balanced loss function to re-weight samples more effectively in such a way that the overfitting of the decision boundary can be alleviated.
- We demonstrate that simply substituting our proposed loss for the standard cross-entropy loss significantly improves the generalization performance on highly imbalanced data.

## 2. Related Work

### 2.1. Class Imbalance Learning

To solve the imbalanced learning problem, numerous studies have been conducted. The research can be divided into three approaches: data-level, cost-sensitive re-weighting, and meta-learning approaches.

**Data-level approach.** The data-level approach aims to directly balance the training data distributions by re-sampling (e.g., under-sampling the majority classes or over-sampling the minority classes) [6, 32] or generating synthetic samples [28]. However, under-sampling can lose some valuable information, and it is not applicable when the data imbalance between classes is significant. Although over-sampling or data generation could be effective, these methods are susceptible to overfitting to certain repetitive samples, and often require a longer training time.

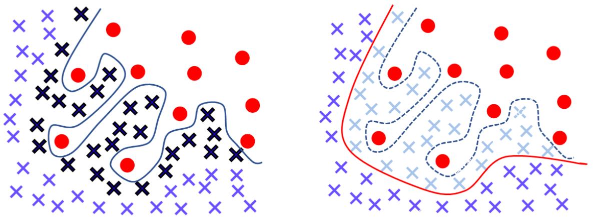
**Re-weighting approach.** Cost-sensitive re-weighting methods assign different weights to samples to adjust their importance. Commonly used methods include re-weighting samples inversely proportional to the number of the class [17, 33] or the square root of class frequency [26]. Instead of heuristically using the number of classes, Cui et al. [8] proposed using the effective number of samples. While these methods can successfully assign more weights to the minority samples, they assign the same weights to all samples belonging to the same class, regardless of each importance.

To assign different weights to each sample according to its importance on the model, numerous methods were proposed for re-weighting samples based on their difficulties or losses [22, 9, 27]. That is, these methods down-weight well-classified samples and assign more weights to hard examples. These re-weighting methods might cause DNNs to be overfitted to the hard examples, since the high capacity of DNNs is sufficient to memorize the training data in the end [3]. In class imbalanced data, the hard examples are likely generated from the majority classes. As such, the minority samples are assigned smaller weights. Therefore, we need a more elaborate mean of re-weighting samples that can alleviate the overfitting to the majority samples. Meanwhile, Cao et al. [5] proposed label-distribution-aware margin loss to solve the overfitting to the minority classes by regularizing the margins.

**Meta-learning approach.** Recently, the meta-learning-based approach [31, 25, 30] has emerged to enhance the performance of both approaches. Shu et al. [31] proposed a meta-learning process to learn a weighting function, while Liu et al. [25] proposed a re-sampling method by combining the advantage of ensemble learning and meta-learning. Furthermore, Ren et al. [30] proposed the meta-sampler and a balanced softmax that accommodates the shift of the distributions between the training data and test data. Although these methods can achieve satisfactory performance, these methods are somewhat difficult to implement in practice. For example, meta-weight-net [31] requires additional unbiased data for learning, and the meta-sampler in [30] is computationally expensive in practice. On the other hand, our proposed loss is simple to implement because it does not require a hyperparameter, a specially designed architecture, or additional learning for data re-sampling. Therefore, it is easy to use in collaboration with other methods.

### 2.2. Influence function.

The influence function was proposed to find the influential instance of a sample to a model, which has been studied for decades in robust statistics [13, 7]. Recently, attempts have been made to use influence function in deep neural networks [1, 19]. For example, Koh and Liang [19] employed the influence function to understand DNNs. While the influence function is primarily used as a diagnostic tool after the



(a) Original decision boundary.

(b) Proposed method.

Figure 1. **Illustration of the key concept of our approach.** The red and blue marks belong to the minority and majority classes, respectively, in binary classification. (a) The black border line represents an initial decision boundary formed on an imbalanced dataset. The black × samples have greater influence on the decision boundary than do the blue × samples, since the decision boundary would substantially change without the black × samples. (b) Our proposed method aims to down-weight the samples (light blue × samples) that have a large influence on the overfitted decision boundary (dotted line) to create a smoother decision boundary (the red line).

training of a model, our work first attempts to apply it to a learning scheme, in which we design the influence-balanced loss by utilizing the influence function during training.

## 3. Method

To address the imbalanced data learning problem, our idea is to re-weight samples by their influences on a decision boundary to create a more generalized decision boundary. First, we present the key idea of our proposed method in Section 3.1. For the background, we briefly review the influence function in Section 3.2 and then derive the IB loss in Sections 3.3, 3.4, and 3.5. Finally, the training scheme is presented in Section 3.6.

### 3.1. Key Idea of Proposed Method

In this section, we explain how the re-weighting of samples according to their influence can help to form a well-generalized decision boundary on class imbalance data. It is well known that the high capacity of DNNs is sufficient to finally memorize the entire training data [34, 3]. This implies that DNN can be overfitted to samples that are located at the overlapping region between the majority and minority classes, as illustrated in Figure 1 (a). In the imbalanced data, many majority samples invade among sparse minority samples and become dominant in the overlapping area, thereby enforcing the decision boundary to be complex and shift to the minority region.

Furthermore, the black × samples in Figure 1 (a) have a stronger influence on forming the decision boundary, as they support the decision boundary, which substantially changes when the samples are removed. Thus, it can be said that the dominant samples with high influence are likely to create a complex and biased decision boundary. As illustrated in Figure 1 (b), by down-weighting the highly influential samples, the decision boundary can be smoothed via fine-tuning. To this end, we derive an influence-balanced (IB) loss by employing the influence function [7], which measures the training sample's influence on the model.

### 3.2. Influence Function

The influence function [7] allows us to estimate the change in the model parameters when a sample is removed, without actually removing the data and retraining the model. Let $f(x, w)$ denote a model parameterized by $w$ with $n$ training data $(x_1, y_1), \cdots, (x_n, y_n)$, where $x_i$ is the $i$-th training sample, and $y_i$ is its label. Given the empirical risk $R(w) = \frac{1}{n}\sum_{i=1}^{n} L(y_i, f(x_i, w))$, the optimal parameter after initial training is defined by $w^* \overset{\text{def}}{=} \text{argmin}_w R(w)$.

During the fine-tuning phase, to address the imbalance issue, we re-weight loss proportionally to the inverse of the influence of a sample. The influence of a point $(x, y)$ can be approximated by the parameter change if the distribution of the training data at that point is slightly modified. A new parameter when removing the training point $(x, y)$ is derived as $w_{x,\varepsilon} \overset{\text{def}}{=} \text{argmin}_w R(w) + \varepsilon L(y, f(x, w))$. Then, under the assumption that $\nabla_w R(w) \approx 0$ for $w$ in the vicinity of $w^*$, we can utilize the influence function in [1, 19] to re-weight the sample-wise loss during the fine-tuning phase. The influence function is given by

$$\mathcal{I}(x; w) = -H^{-1}\nabla_w L(y, f(x, w)), \tag{1}$$

where $H \overset{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n} \nabla_w^2 L(y_i, f(x_i, w))$ is the Hessian and is positive definite based by assumption that $L$ is strictly convex in a local convex basin around the optimal point $w^*$.

### 3.3. Influence-balanced weighting factor

From $\mathcal{I}(x; w)$, we derive the IB loss. Since $\mathcal{I}(x; w)$ is a vector that requires heavy computation of the inverse Hessian, it is nearly impossible to directly use this. Therefore, we solve this problem by modifying $\mathcal{I}(x; w)$ to a simple but effective influence-balanced weighting factor. First, since we need the relative influence of the training samples, not the absolute values, we can simply ignore the inverse Hessian in $\mathcal{I}(x; w)$. This is because the inverse of hessian is commonly multiplied by all the training samples. Then, we design the IB weighting factor as follows:

$$\mathcal{IB}(x; w) = ||\nabla_w L(y, f(x, w))||_1 \tag{2}$$

Equation 2 turns out to be the magnitude of the gradient vector. Anand et al. [2] revealed that the net error gradient vector is dominated by the major classes in the class imbalance problem. Hence, re-weighting samples by the magnitude of the gradient vector can successfully down-weight samples from dominant classes. In the Experiments section, we justify the choice of the L1 norm. In the following section, we demonstrate how the IB weighting factor can be used with the actual loss.

### 3.4. Influence-Balanced Loss

When using the softmax cross-entropy loss, Equation (2) can be further simplified. The cross-entropy loss is denoted by $L(y, f(x,w)) = -\sum_k^K y_k \log f_k$, where $y_k$ is a ground truth, and $f_k$ is the $k$-th output of the model $f(x,w)$, with $K$ total classes. Since we are interested in the overfitting on the decision boundary of the model, we focus on the change in the last fully connected (FC) layer of a deep neural network. Let $h = [h_1, \cdots, h_L]^T$ be a hidden feature vector, an input to the FC layer, and $f(x,w) = [f_1, \cdots, f_K]^T$ be the output denoted by $f_k := \sigma(w_k^T h)$, where $\sigma$ is the softmax function. The weight matrix of the FC layer is denoted by $w = [w_1, \cdots, w_K]^T \in R^{K\times f}$.

Then, the gradient of the loss w.r.t. $w_{kl}$ is computed as

$$\frac{\partial}{\partial w_{kl}} L(y, f(x,w)) = (f_k - y_k)h_l. \tag{3}$$

The same results are obtained for the cross-entropy loss with a sigmoid function or a mean squared error (MSE) loss for regression. Then, IB weighting factor in (2) is given by

$$\begin{aligned}\mathcal{IB}(x;w) &= \sum_k^K \sum_l^L |(f_k - y_k)h_l| \\ &= \sum_k^K |(f_k - y_k)| \sum_l^L |h_l| \\ &= ||f(x,w) - y||_1 \cdot ||h||_1,\end{aligned} \tag{4}$$

of which inverse can be used for the re-weighting factor to down-weight an influential sample in fine-tuning to adjust the decision boundary that enhance the imbalanced data learning. Finally, the influence-balanced loss is given by

$$L_{IB}(y, f(x,w) = \frac{L(y, f(x,w))}{||f(x,w) - y||_1 \cdot ||h||_1}. \tag{5}$$

The proposed influence-balanced term constrains the decision boundary to not overfit to influential majority samples (see Figure 1(b)).

### 3.5. Influence-Balanced Class-wise Re-weighting

Moreover, we add a class-wise re-weighting term $\lambda_k$ to the IB-loss in (5) as

$$L_{IB}(w) = \frac{1}{m} \sum_{(x,y)\in D_m} \lambda_k \frac{L(y, f(x,w))}{||f(x,w) - y||_1 \cdot ||h||_1}, \tag{6}$$

where $\lambda_k = \alpha n_k^{-1} / \sum_{k'=1}^K n_{k'}^{-1}$. Here, $n_k$ is the number of samples in the $k$-th class in the training dataset, and normalization is performed to make $\lambda_k$ have a similar scale for every class. $\alpha$ is introduced as a hyper-parameter for an adjustment.

```
Algorithm 1: Influence-Balanced Training
Input  : training dataset $D = (X, Y)$.
Output: influence-balanced model $f(x,w)$.
Phase 1: Normal training
Initialize the model with random parameters $w$.
for $t = 1$ to $T_1$ do
    sample mini-batch $D_m$ from $D$
    $L(w) \leftarrow \frac{1}{m}\sum_{(x,y)\in D_m} L(y, f(x,w))$
    update $w^t = w^{t-1} - \eta\nabla L(w)$
end
Phase 2: Fine-tuning for influence balancing
for $t = T_1 + 1$ to $T$ do
    sample mini-batch $D_m$ from $D$
    $L_{IB}(w) \leftarrow \frac{1}{m}\sum_{(x,y)\in D_m} \lambda_k \frac{L(y,f(x,w))}{||f(x,w)-y||_1 \cdot ||h||_1}$
    update $w^t = w^{t-1} - \eta\nabla L(w)$
end
```

The class-wise re-weighting yields the following two effects. First, $\lambda_k$ mitigates the bias of the decision boundary arising from the overall imbalanced distribution through the slow-down of the majority loss minimization. Second, $\lambda_k$ further controls the sample-wise re-weighting depending on the class to which a highly influential sample belongs. That is, if the sample belongs to a majority class, $\lambda_k$ further down-weights the sample because the decision boundary is likely to be overfitted by the majority sample. Meanwhile, if the sample belongs to a minority class, $\lambda_k$ becomes smaller than that of a majority sample and does not down-weight the loss much, because the large influence of the minority sample is natural due to the data scarcity.

### 3.6. Influence-balanced Training Scheme

The influence-balanced training process comprises two phases: normal training and fine-tuning for balance. We refer to $T_1$ as the transition time from normal training to fine-tuning. During the normal training phase, the network is trained following any training scheme for the first $T_1$ epochs. Meanwhile, during the fine-tuning phase, the influence-balanced loss is applied to mitigate the overfitting of the decision boundary arising from the influential (noisy) majority samples. Since our IB loss during the fine-tuning phase alleviates the overfitting, it is advantageous to set $T_1$ as the epoch when the model has begun to converge to the local (global) minimum. Generally, it is recommended to set $T_1$ as half of the total training scheme. We present the performance change according to the number of training epochs during normal training in the Experiments section. As evident, our training does not require an additional training scheme or a specifically designed architecture. Thus, it can be utilized easily in any tasks suffering from imbalanced data. The pseudo-code of the training procedure is presented in Algorithm 1.

# 4. Experiments

## 4.1. Experimental Settings

**Datasets.** We verified the effectiveness of our method on three commonly used benchmark datasets: CIFAR-10, CIFAR-100 [20], Tiny ImageNet [21], and iNaturalist 2018 [16]. The CIFAR-10 and CIFAR-100 datasets consist of 50,000 training images and 10,000 test images with 10 and 100 classes, respectively. Meanwhile, Tiny ImageNet contains 200 classes for training, in which each class has 500 images. Its test set contains 10,000 images. Since CIFAR and Tiny ImageNet are evenly distributed, we have made these datasets imbalanced according to [8, 4], respectively. Primarily, we investigate two common types of imbalance: (i) long-tailed imbalance [8] and (ii) step imbalance [4]. In long-tailed imbalance, the number of training samples for each class decreases exponentially from the largest majority class to the smallest minority class. To construct long-tailed imbalanced datasets, the number of selected samples in the $k$-th class was set to $n_k \mu^k (\mu \in (0,1))$, where $n_k$ is the original number of the $k$-th class. Meanwhile, in step imbalance, the classes are divided into two groups: the majority class group and minority class group. Every class within a group contains the same number of samples, and the class in the majority class group has many more samples than that in the minority class group. For evaluation, we used the original test set. The imbalance ratio $\rho$ is defined by $\rho = \frac{\max_k\{n_k\}}{\min_k\{n_k\}}$. Thus, the imbalance ratio represents the degree of imbalance in the dataset. We evaluated the performance of our method under various imbalance ratios from 10 to 200.

The iNaturalist 2018 dataset is a large-scale real-world dataset containing 437,513 training images and 24,426 test images with 8,142 classes. iNaturalist 2018 exhibits long-tailed imbalance, whose imbalance ratio is 500. We used the official training and test splits in our experiments.

**Baselines.** We compared our algorithm with the following cost-sensitive loss methods: (1) Our baseline model, which is trained on the standard cross-entropy loss. Comparing our model with this baseline enables us to clearly understand how much our training scheme has improved the performance; (2) focal loss [22], which increases the relative loss for hard samples and down-weights well-classified samples; (3) CB loss [8], which re-weights the loss inversely proportional to the effective number of samples; (4) LDAM loss [5], which regularizes the minority classes to have larger margins.

Since our IB loss can be easily combined with other methods, we employee two further variants. First, IB + CB uses the effective number in CB loss, instead of using $\lambda_k$ in IB. Second, IB + focal uses focal loss during the fine-tuning phase, instead of using the cross-entropy loss. We demonstrate that combination with other methods can further improve the performance.

**Implementation Details.** We used PyTorch [29] to implement and train all the models in the paper, and we used ResNet architecture [15] for all datasets. For CIFAR datasets, we used randomly initialized ResNet-32. The networks were trained for 200 epochs with stochastic gradient descent (SGD) (momentum = 0.9). Following the training strategy in [8, 5], the initial learning rate was set to 0.1 and then decayed by 0.01 at 160 epochs and again at 180 epochs. Furthermore, we used a linear warm-up of the learning rate [12] in the first five epochs. Since our method uses a two-phase training schedule, we trained for the first 100 epochs with the standard cross-entropy loss, then fine-tuned the networks using the IB loss for the next 100 epochs. We trained the models for CIFAR on a single NVIDIA GTX 1080Ti with a batch size of 128. For Tiny ImageNet, we employed ResNet-18 and used the stochastic gradient descent with a momentum of 0.9, and weight decay of $2\text{e}{-4}$ for training. The networks were initially trained for 50 epochs, and then fine-tuned for the subsequent 50 epochs with IB loss. The learning rate at the start was set to 0.1 and was dropped by a factor of 0.1 after 50 and 90 epochs. For iNaturalist 2018, we trained ResNet-50 with four GTX 1080Ti GPUs. The networks were initially trained for 50 epochs and then fine-tuned for the subsequent 150 epochs with IB loss. The learning rate at the start was set to 0.01 and was decreased by a factor of 0.1 after 30 and 180 epochs.

As a simple but important implementation trick, we added $\epsilon = 0.001$ to $\mathcal{IB}(x;w)$ to prevent numerical instability in inversion when the influence approaches zero. We discuss the influence of the hyperparameter ($\epsilon$) in the following section.

## 4.2. Analysis

To validate the proposed method, we conducted extensive experiments.

**Is influence meaningful for re-weighting?** First, to confirm whether influence can act as a meaningful clue of re-weighting for class imbalance learning, we compared the influences between a balanced dataset and an imbalanced dataset. For an imbalanced CIFAR-10, we used the long-tailed version of CIFAR-10 with the imbalance ratio $\rho = 100$, in which the largest class, 'plane' (i.e., class index 0), contains 5,000 samples, while the smallest class, 'truck' (i.e., class index 9), contains only 50 samples. We trained ResNet-32 with a standard cross-entropy loss for 200 epochs, as described in Implementation Details, on both the balanced (original) and imbalanced CIFAR-10. We plotted the influences of both classes in Figure 2. We scaled the influences to between 0 and 1 for each dataset. Since the minority class contains only 50 samples, we selected the highest 50 samples for comparison. As illustrated in Figure 2, there was little difference in the distributions of the in-

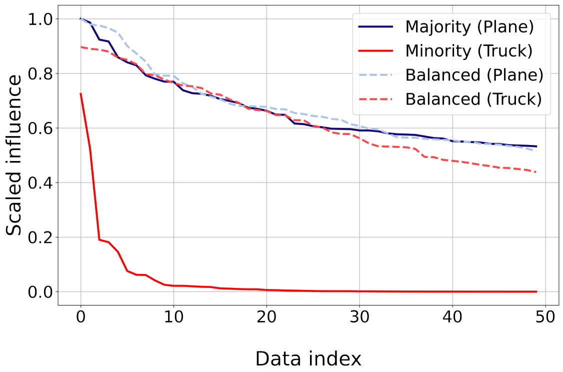


Figure 2. **Comparison of Influences between balanced and imbalanced dataset.** We plotted the influences of samples on ResNet-32 trained on the original CIFAR-10 and the imbalanced version of CIFAR-10. The solid and dashed lines represent the influences of the imbalanced data and balanced data, respectively. While there is little difference in the balanced dataset, it can be seen that the influence of the dominant class is much greater than that of the minor class in the imbalance dataset.

fluences between the classes in the balanced dataset. However, in the imbalanced dataset, the minority samples had significantly less influence on the model than did the majority samples. This result corroborates that majority samples greatly contribute to forming a decision boundary, and reweighting their influences can improve the generalization of the model.

**Magnitude of Influence.** In Section 3.3, we used $L_1$ norm to compute the magnitude of the influences. We investigated performance variations depending on three vector norms to compute the magnitude of the gradient vector $\nabla_w L(y, f(x, w))$: $L_1$, $L_2$, $L_\infty$. As indicated in Table 1, $L_1$ norm, which provides a distinctive change of influence around the equilibrium point, exhibits the best classification accuracy on CIFAR-10 with multiple imbalance ratios.

Table 1. Comparison of norms. Using $L_1$ norm yields the best performance.

| | CIFAR-10 | | CIFAR-100 | |
|---|---|---|---|---|
| Imbalance ($\rho$) | 100 | 20 | 100 | 20 |
| $L_1$ | **78.41** | **85.80** | **40.85** | **52.85** |
| $L_2$ | 75.67 | 84.35 | 36.41 | 50.95 |
| $L_\infty$ | 77.23 | 84.30 | 37.48 | 50.99 |

**Timing for starting fine-tuning for balancing.** Our training scheme is divided into two phases: normal training and fine-tuning for balancing. This must determine the transition time between normal training and fine-tuning for balancing. Hence, we investigated the results on how much the transition time affects the performance and determined the best transition time. For this, we experimented on the long-tailed version of CIFAR-10 with imbalance ratios of $\rho = 10$ and 100. In Figure 3, the $X$-axis represents the number of training epochs $T_1$ for the normal training phase. We varied the transition time, $T_1$, from 0 to 120 while the total number of training epochs was fixed at 200. The solid line represents the classification accuracy earned by the models for each training schedule. To analyze the relationship between the convergence of the normal training phase and the transition timing, we plotted the standard cross-entropy loss without adopting the IB loss for the whole training epochs (dashed lines).

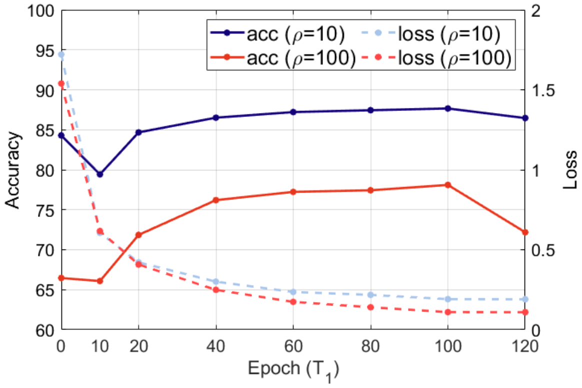


Figure 3. **Influence-balanced training scheme.** We varied the *training epochs for the normal training*, $T_1$, to determine the best transition time from the normal training to the influence-balance fine-tuning. We achieved the best performance when setting the transition time to the point when the training loss converges.

From Figure 3, it can be observed that the proposed method demonstrates robust performance regardless of the choice of transition time $T_1$. Yet, the transition to fine-tuning after the 100th epoch yields the best performance when the training loss has converged. Since the influence function is derived from the loss minimization context [19], it is reasonable to begin the fine-tuning phase after the learning converges.

**Effects of $\epsilon$.** As mentioned in Implementation Details, for all datasets, we added the hyperparameter ($\epsilon = 0.001$) to $\mathcal{IB}(x; w)$ to prevent numerical instability. To analyze the effects of the hyperparameter, we conducted experiments with the following denominators for the IB loss (5): (a) $\mathcal{IB}(x; w) + 1\mathrm{e}{-8}$, (b) $\mathcal{IB}(x; w) + 1\mathrm{e}{-3}$, (c) $\mathcal{IB}(x; w) + 1\mathrm{e}{-2}$, and (d) $1\mathrm{e}{-3}$. We iterated experiments three times with different random seeds on the long-tailed CIFAR-10 ($\rho = 100$). As presented in Table 2, setting $\epsilon$ to $1\mathrm{e}{-3}$ yields the best performance. Thus, we set $\epsilon$ as $1\mathrm{e}{-3}$ in all the experiments. However, when we did not use the IB weighting factor, the accuracy greatly decreased.

Table 2. Effects of $\epsilon$.

| Epsiilon | (a) IB+1e-8 | (b) IB+1e-3 | (c) IB+1e-2 | (d) 1e-3 |
|---|---|---|---|---|
| Accuracy | 76.03 ± 0.97 | 78.17 ± 0.57 | 77.55 ± 0.55 | 64.91 ± 1.40 |

Table 3. Class-wise classification accuracy (%) of ResNet-32 on imbalanced CIFAR-10 dataset. The number of test samples for each class is the same as 1000. The best results are marked in bold.

| | **Imbalanced CIFAR-10** | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Class | plane | car | bird | cat | deer | dog | frog | horse | ship | truck |
| ***Long-Tailed*** ($\rho = 50$) | | | | | | | | | | |
| #Training samples | 5000 | 3237 | 2096 | 1357 | 878 | 568 | 368 | 238 | 154 | 100 |
| Baseline (CE) | **97.4** | **98.0** | **84.0** | **80.3** | 78.8 | 68.4 | 76.1 | 64.5 | 57.0 | 52.0 |
| Focal [22] | 91.6 | 95.1 | 73.1 | 59.2 | 67.8 | 67.2 | 84.2 | 77.3 | **83.9** | 61.8 |
| CB [8] | 92.9 | 96.3 | 79.2 | 75.1 | 82.4 | 69.9 | 75.0 | 69.1 | 73.6 | 66.8 |
| LDAM [5] | 96.9 | 98.5 | 82.9 | 74.7 | 82.8 | 69.0 | 78.5 | 69.9 | 65.3 | 66.0 |
| LDAM-DRW [5] | 94.8 | 97.8 | 82.6 | 72.3 | 85.3 | 73.0 | 82.0 | 76.7 | 75.8 | 72.4 |
| IB | 92.2 | 96.2 | 81.3 | 66.6 | **85.7** | **76.4** | 81.7 | 75.9 | 79.9 | **81.1** |
| IB + CB | 93.8 | 97.2 | 78.1 | 64.8 | 84.8 | 74.2 | **86.4** | **79.7** | 79.5 | 76.9 |
| IB + Focal | 90.9 | 96.1 | 81.7 | 69.0 | 82.0 | 75.7 | 85.2 | 77.5 | 80.2 | 76.8 |
| ***Step-Imbalance*** ($\rho = 50$) | | | | | | | | | | |
| #Training samples | 5000 | 5000 | 5000 | 5000 | 5000 | 100 | 100 | 100 | 100 | 100 |
| Baseline (CE) | 95.9 | **99.2** | **91.5** | **91.9** | 95.5 | 24.8 | 40.2 | 46.7 | 52.7 | 55.1 |
| Focal [22] | 96.3 | 93.9 | 91.2 | 90.5 | **95.7** | 20.0 | 46.7 | 48.8 | 56.1 | 57.6 |
| CB [8] | 87.4 | 96.3 | 76.8 | 77.0 | 85.7 | 34.6 | 61.5 | 56.5 | 68.7 | 63.8 |
| LDAM [5] | **96.4** | 98.5 | 91.1 | 90.2 | 94.6 | 28.3 | 50.3 | 57.0 | 56.2 | 64.4 |
| LDAM-DRW [5] | 94.5 | 97.2 | 88.0 | 84.5 | 94.3 | 50.4 | 69.9 | 71.4 | 74.6 | 76.0 |
| IB | 94.0 | 97.7 | 86.7 | 83.2 | 93.8 | 56.9 | 71.0 | **75.1** | 76.5 | 81.7 |
| IB + CB | 91.8 | 95.7 | 86.6 | 79.4 | 93.6 | 62.8 | 77.2 | 72.3 | 74.2 | **87.3** |
| IB + Focal | 91.2 | 96.4 | 83.3 | 77.1 | 92.0 | **64.8** | **78.0** | 74.4 | **83.5** | 83.1 |

### 4.3. Comparison of Class-Wise Accuracy.

In this section, to validate that the performance improvement has actually resulted from the minority classes, not from the majority classes, we report the class-wise accuracy on both the long-tailed and the step-imbalanced CIFAR-10. We compare the proposed method with the state-of-the-art cost-sensitive loss methods. Since previous studies do not report the class-wise accuracy on the imbalanced CIFAR-10, we implemented the baseline methods [22, 8, 5]. For the implementation of LDAM [5], we used their official implementation code to reproduce the results.

The overall results are reported in Table 3. As presented in Table 3, existing methods exhibit severe performance degradation in the minority classes. That is, the reported improvements from the existing methods were attributed to the majority classes, not the minority classes. In contrast, the proposed IB loss exhibited a significant improvement in all the minority classes.

It is noteworthy that the performance improvement was not significant, especially on the step-imbalanced CIFAR-10 with the focal loss [22] method. We argue that this demonstrates that most hard examples are majority samples in highly imbalanced data and that those samples enforce the decision boundary to be overfitted. In contrast, our proposed influence-balanced re-weighing can mitigate the influences of the majority samples that cause overfitting. As a result, it can achieve robust and superior performance for the minority classes with a very small number of samples.

Although using the influence-balanced loss alone can achieve significant enhancement for the classification of the minority classes, it is beneficial to combine it with other methods. For example, the results indicate that applying the influence-balanced loss with the focal loss can encourage the network to learn 'good' hard samples, while down-weighting the influential ones that induce overfitting.

### 4.4. Comparison with State-of-the-Art

**Experimental results on CIFAR.** The overall classification accuracy is provided in Table 4. The model performance is reported on the unbiased test set as the same as the other methods. The results indicate that adopting the proposed influence-balanced loss significantly improves the generalization performance and outperforms the recent cost-sensitive loss methods. On multiple benchmark datasets, using IB loss alone could achieve the best performance. This suggests that it is effective for the robustness of the model to balance the influence of samples responsible for overfitting the decision boundary. When combined with other methods [8, 22], we could further improve the accuracy on multiple datasets. This indicates that our proposed method of down-weighting influential samples that induce overfitting can benefit other methods as well.

Table 4. Classification accuracy (%) of ResNet-32 on imbalanced CIFAR-10 and CIFAR-100 datasets. "†" indicates that the results are copied from the original paper, and "‡" means that the results are from the experiments in CB [8]. The best results are marked in bold.

| | **Imbalanced CIFAR-10** | | | | | **Imbalanced CIFAR-100** | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Imbalance ($\rho$) | 200 | 100 | 50 | 20 | 10 | 200 | 100 | 50 | 20 | 10 |
| ***Long-Tailed*** | | | | | | | | | | |
| Baseline (CE) | 66.28 | 70.87 | 78.22 | 82.43 | 86.49 | 33.54 | 38.05 | 43.71 | 51.21 | 56.96 |
| ‡Focal [22] | 65.29 | 70.38 | 76.71 | 82.76 | 86.66 | 35.62 | 38.41 | 44.32 | 51.95 | 55.78 |
| †CB [8] | 68.89 | 74.57 | 79.27 | 84.36 | 87.49 | 36.23 | 39.60 | 45.32 | 52.59 | 57.99 |
| †LDAM [5] | - | 73.35 | - | - | 86.96 | - | 39.6 | - | - | 56.91 |
| †LDAM-DRW [5] | - | 77.03 | - | - | 88.16 | - | 42.04 | - | - | 57.99 |
| IB | 73.96 | 78.26 | **81.70** | **85.8** | **88.25** | 37.31 | **42.14** | 46.22 | 52.63 | 57.13 |
| IB + CB | 73.69 | 78.04 | 81.54 | 85.42 | 88.09 | 37.06 | 41.31 | 46.16 | 52.74 | 56.78 |
| IB + Focal | **75.05** | **79.76** | 81.51 | 85.31 | 88.04 | **38.23** | 42.06 | **47.49** | **53.28** | **58.20** |
| ***Step-Imbalance*** | | | | | | | | | | |
| Baseline (CE) | 56.97 | 64.81 | 69.35 | 79.71 | 84.16 | 38.29 | 39.27 | 41.65 | 48.55 | 54.13 |
| †LDAM [5] | - | 66.58 | - | - | 85.00 | - | 39.58 | - | - | 56.27 |
| †LDAM-DRW [5] | - | 76.92 | - | - | 87.81 | - | 45.36 | - | - | 59.46 |
| IB | 72.15 | 76.53 | 81.66 | 85.41 | 87.72 | 39.66 | **45.39** | **48.93** | 53.57 | 57.96 |
| IB + CB | 69.96 | 75.97 | 82.09 | 85.27 | **88.01** | 39.69 | 45.27 | 48.80 | 53.42 | 57.86 |
| IB + Focal | **74.12** | **77.97** | **82.38** | **85.68** | 87.90 | **40.39** | 44.96 | 48.92 | **54.53** | **59.54** |

**Experimental results on Tiny ImageNet.** We evaluated our method on Tiny ImageNet. While we performed the experiments for the other baselines, the results of LDAM were copied from their original paper. As presented in Table 5, IB loss outperforms other baselines on Tiny ImageNet as well.

**Experimental results on iNaturalist 2018.** We evaluated our method on the large-scale real-world image data, iNaturalist 2018. We compared our method with the state-of-the-art loss-based methods. Table 6 reveals that simply balancing the influence of loss could achieve considerable improvement.

Table 5. Class. accuracy (%) of ResNet-18 on Tiny ImageNet.

| | ***Long-Tailed*** | | ***Step-Imbalance*** | |
|---|---|---|---|---|
| Imbalance ($\rho$) | 100 | 10 | 100 | 10 |
| Baseline (CE) | 38.52 | 36.62 | 36.74 | 51.11 |
| Focal [22] | 38.95 | 54.02 | 38.24 | 41.77 |
| CB [8] | 41.37 | 54.82 | 37.35 | 54.3 |
| LDAM* [5] | 37.47 | 52.78 | 39.37 | 52.57 |
| IB | **42.65** | **57.22** | **41.13** | **54.83** |

Table 6. Class. accuracy (%) of ResNet-50 on iNaturalist 2018.

| | **iNaturalist 2018** | |
|---|---|---|
| Method | top1 | top5 |
| Baseline (CE) | 57.30 | 79.48 |
| Focal [22] | 58.03 | 78.65 |
| CB [8] | 61.12 | 81.03 |
| LDAM [5] | 64.58 | 83.52 |
| IB | **65.39** | **84.98** |

## 5. Conclusion

In this paper, we propose a novel influence-balanced loss to solve the overfitting of the majority classes in a class imbalance problem. A model trained on imbalanced class data is susceptible to overfitting due to the high capacity of DNN and the scarcity of samples in certain classes. Therefore, as learning progresses, existing methods are likely to produce undesirable results, such as assigning higher weights to samples from majority classes. Unlike the existing methods, IB loss can robustly assign weights because it directly focuses on a sample's influence on the model. We conducted experiments to demonstrate that our method can improve generalization performance under a class imbalance setting. In addition, our method is easy to be implemented and integrated into existing methods. In the future, we plan to extend our method by incorporating data-level methods or other recent meta-learning methods.

## Acknowledgement

This work was supported by Institute of Information & Communications Technology Planning & Evaluation(IITP) grants funded by the Korea government(MSIT) (No.B0101-15-0266, Development of High Performance Visual Big-Data Discovery Platform for Large-Scale Realtime Data Analysis) and (2017-0-00306, Multimodal sensor-based intelligent systems for outdoor surveillance robots).

## References

[1] Héctor Allende, Rodrigo Salas, and Claudio Moraga. A robust and effective learning algorithm for feedforward neural networks based on the influence function. In *Pattern Recognition and Image Analysis*, 2003. 2, 3

[2] R. Anand, K. G. Mehrotra, C. K. Mohan, and S. Ranka. An improved algorithm for neural network classification of imbalanced training sets. *IEEE Transactions on Neural Networks*, 1993. 3

[3] Devansh Arpit, Stanisław Jastrzundefinedbski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. A closer look at memorization in deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, 2017. 1, 2, 3

[4] Mateusz Buda, Atsuto Maki, and Maciej A. Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106, 2018. 1, 5

[5] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *Advances in Neural Information Processing Systems*, 2019. 2, 5, 7, 8

[6] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote: Synthetic minority oversampling technique. *J. Artif. Int. Res.*, 2002. 1, 2

[7] R. Dennis Cook and Sanford Weisberg. Residuals and Influence in Regression. 1982. 1, 2, 3

[8] Y. Cui, M. Jia, T. Lin, Y. Song, and S. Belongie. Class-balanced loss based on effective number of samples. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2, 5, 7, 8

[9] Q. Dong, S. Gong, and X. Zhu. Class rectification hard mining for imbalanced deep learning. In *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017. 1, 2

[10] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. 1

[11] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88, 2009. 1

[12] Priya Goyal, Piotr Dollár, Ross B. Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch SGD: training imagenet in 1 hour. *CoRR*, 2017. 5

[13] F.R. Hampel. *Robust Statistics: The Approach Based on Influence Functions*. Probability and Statistics Series. Wiley, 1986. 2

[14] Haibo He and E.A. Garcia. Learning from imbalanced data. *Knowledge and Data Engineering, IEEE Transactions on*, 21, 2009. 1

[15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 5

[16] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alexander Shepard, Hartwig Adam, Pietro Perona, and Serge J. Belongie. The inaturalist species classification and detection dataset. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 2018. 1, 5

[17] C. Huang, Y. Li, C. C. Loy, and X. Tang. Learning deep representation for imbalanced classification. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 2

[18] Nathalie Japkowicz and Shaju Stephen. The class imbalance problem: A systematic study. *Intelligent Data Analysis*, pages 429–449, 2002. 1

[19] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1885–1894, Sydney, Australia, 2017. PMLR. 2, 3, 6

[20] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto*, 2009. 5

[21] Ya Le and X. Yang. Tiny imagenet visual recognition challenge. 2015. 5

[22] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017. 1, 2, 5, 7, 8

[23] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollar, and Larry Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 1

[24] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X. Yu. Large-scale long-tailed recognition in an open world. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, 2019. 1

[25] Zhining Liu, Pengfei Wei, Jing Jiang, Wei Cao, Jiang Bian, and Yi Chang. MESA: boost ensemble imbalanced learning with meta-sampler. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. 1, 2

[26] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, 2018. 2

[27] T. Malisiewicz, A. Gupta, and A. A. Efros. Ensemble of exemplar-svms for object detection and beyond. In *2011 International Conference on Computer Vision*, 2011. 1, 2

[28] Sankha Subhra Mullick, Shounak Datta, and Swagatam Das. Generative adversarial minority oversampling. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, 2019. 1, 2

[29] 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-W*, 2017. 5

[30] Jiawei Ren, Cunjun Yu, Shunan Sheng, Xiao Ma, Haiyu Zhao, Shuai Yi, and Hongsheng Li. Balanced meta-softmax for long-tailed visual recognition. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. 1, 2

[31] Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. Meta-weight-net: Learning an explicit mapping for sample weighting. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 2019. 1, 2

[32] Jason Van Hulse, Taghi M. Khoshgoftaar, and Amri Napolitano. Experimental perspectives on learning from imbalanced data. In *Proceedings of the 24th International Conference on Machine Learning*, 2007. 1, 2

[33] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Learning to model the tail. In *Advances in Neural Information Processing Systems*, 2017. 1, 2

[34] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017. 1, 3