Learning Imbalanced Datasets with Label-Distribution-Aware Margin Loss

Kaidi Cao
Stanford University
kaidicao@stanford.edu

Colin Wei
Stanford University
colinwei@stanford.edu

Adrien Gaidon
Toyota Research Institute
adrien.gaidon@tri.global

Nikos Arechiga
Toyota Research Institute
nikos.arechiga@tri.global

Tengyu Ma
Stanford University
tengyuma@stanford.edu

Abstract

Deep learning algorithms can fare poorly when the training dataset suffers from heavy class-imbalance but the testing criterion requires good generalization on less frequent classes. We design two novel methods to improve performance in such scenarios. First, we propose a theoretically-principled label-distribution-aware margin (LDAM) loss motivated by minimizing a margin-based generalization bound. This loss replaces the standard cross-entropy objective during training and can be applied with prior strategies for training with class-imbalance such as re-weighting or re-sampling. Second, we propose a simple, yet effective, training schedule that defers re-weighting until after the initial stage, allowing the model to learn an initial representation while avoiding some of the complications associated with re-weighting or re-sampling. We test our methods on several benchmark vision tasks including the real-world imbalanced dataset iNaturalist 2018. Our experiments show that either of these methods alone can already improve over existing techniques and their combination achieves even better performance gains[1].

1 Introduction

Modern real-world large-scale datasets often have long-tailed label distributions [Van Horn and Perona, 2017, Krishna et al., 2017, Lin et al., 2014, Everingham et al., 2010, Guo et al., 2016, Thomee et al., 2015, Liu et al., 2019]. On these datasets, deep neural networks have been found to perform poorly on less represented classes [He and Garcia, 2008, Van Horn and Perona, 2017, Buda et al., 2018]. This is particularly detrimental if the testing criterion places more emphasis on minority classes. For example, accuracy on a uniform label distribution or the minimum accuracy among all classes are examples of such criteria. These are common scenarios in many applications [Cao et al., 2018, Merler et al., 2019, Hinnefeld et al., 2018] due to various practical concerns such as transferability to new domains, fairness, etc.

The two common approaches for learning long-tailed data are re-weighting the losses of the examples and re-sampling the examples in the SGD mini-batch (see [Buda et al., 2018, Huang et al., 2016, Cui et al., 2019, He and Garcia, 2008, He and Ma, 2013, Chawla et al., 2002] and the references therein). They both devise a training loss that is in expectation closer to the test distribution, and therefore can achieve better trade-offs between the accuracies of the frequent classes and the minority classes. However, because we have fundamentally less information about the minority classes and the models

[1] Code available at https://github.com/kaidic/LDAM-DRW.

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

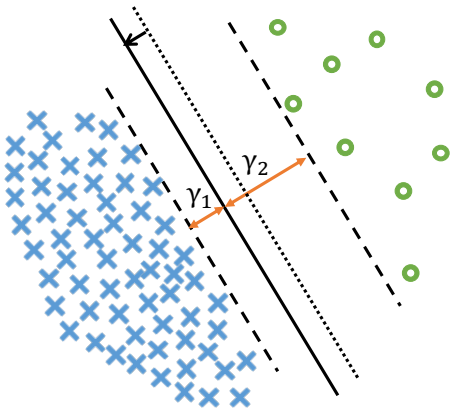


Figure 1: For binary classification with a linearly separable classifier, the margin γ_i of the i-th class is defined to be the the minimum distance of the data in the i-th class to the decision boundary. We show that the test error with the uniform label distribution is bounded by a quantity that scales in $\frac{1}{\gamma_1\sqrt{n_1}} + \frac{1}{\gamma_2\sqrt{n_2}}$. As illustrated here, fixing the direction of the decision boundary leads to a fixed $\gamma_1 + \gamma_2$, but the trade-off between γ_1, γ_2 can be optimized by shifting the decision boundary. As derived in Section 3.1, the optimal trade-off is $\gamma_i \propto n_i^{-1/4}$ where n_i is the sample size of the i-th class.

deployed are often huge, over-fitting to the minority classes appears to be one of the challenges in improving these methods.

We propose to regularize the minority classes more strongly than the frequent classes so that we can improve the generalization error of minority classes without sacrificing the model's ability to fit the frequent classes. Implementing this general idea requires a data-dependent or label-dependent regularizer — which in contrast to standard ℓ_2 regularization depends not only on the weight matrices but also on the labels — to differentiate frequent and minority classes. The theoretical understanding of data-dependent regularizers is sparse (see [Wei and Ma, 2019, Nagarajan and Kolter, 2019, Arora et al., 2018] for a few recent works.)

We explore one of the simplest and most well-understood data-dependent properties: the margins of the training examples. Encouraging a large margin can be viewed as regularization, as standard generalization error bounds (e.g., [Bartlett et al., 2017, Wei et al., 2018]) depend on the inverse of the minimum margin among all the examples. Motivated by the question of generalization with respect to minority classes, we instead study the minimum margin *per class* and obtain per-class and uniform-label test error bounds.[2] Minimizing the obtained bounds gives an optimal trade-off between the margins of the classes. See Figure 1 for an illustration in the binary classification case.

Inspired by the theory, we design a label-distribution-aware loss function that encourages the model to have the optimal trade-off between per-class margins. The proposed loss extends the existing soft margin loss [Wang et al., 2018a] by encouraging the minority classes to have larger margins. As a label-dependent regularization technique, our modified loss function is orthogonal to the re-weighting and re-sampling approach. In fact, we also design a deferred re-balancing optimization procedure that allows us to combine the re-weighting strategy with our loss (or other losses) in a more efficient way.

In summary, our main contributions are (i) we design a label-distribution-aware loss function to encourage larger margins for minority classes, (ii) we propose a simple deferred re-balancing optimization procedure to apply re-weighting more effectively, and (iii) our practical implementation shows significant improvements on several benchmark vision tasks, such as artificially imbalanced CIFAR and Tiny ImageNet [tin], and the real-world large-scale imbalanced dataset iNaturalist'18 [Van Horn et al., 2018].

2 Related Works

Most existing algorithms for learning imbalanced datasets can be divided in to two categories: re-sampling and re-weighting.

Re-sampling. There are two types of re-sampling techniques: over-sampling the minority classes (see e.g., [Shen et al., 2016, Zhong et al., 2016, Buda et al., 2018, Byrd and Lipton, 2019] and references therein) and under-sampling the frequent classes (see, e.g., [He and Garcia, 2008, Japkowicz and Stephen, 2002, Buda et al., 2018] and the references therein.) The downside of under-sampling is

[2] The same technique can also be used for other test label distribution as long as the test label distribution is known. See Section C.5 for some experimental results.

that it discards a large portion of the data and thus is not feasible when data imbalance is extreme. Over-sampling is effective in a lot of cases but can lead to over-fitting of the minority classes [Chawla et al., 2002, Cui et al., 2019]. Stronger data augmentation for minority classes can help alleviate the over-fitting [Chawla et al., 2002, Zou et al., 2018].

Re-weighting. Cost-sensitive re-weighting assigns (adaptive) weights for different classes or even different samples. The vanilla scheme re-weights classes proportionally to the inverse of their frequency [Huang et al., 2016, 2019, Wang et al., 2017]. Re-weighting methods tend to make the optimization of deep models difficult under extreme data imbalanced settings and large-scale scenarios [Huang et al., 2016, 2019]. Cui et al. [2019] observe that re-weighting by inverse class frequency yields poor performance on frequent classes, and thus propose re-weighting by the inverse effective number of samples. This is the main prior work that we empirically compare with.

Another line of work assigns weights to each sample based on their individual properties. Focal loss [Lin et al., 2017] down-weights the well-classified examples; Li et al. [2018] suggests an improved technique which down-weights examples with either very small gradients or large gradients because examples with small gradients are well-classified and those with large gradients tend to be outliers.

In a recent work [Byrd and Lipton, 2019], Byrd and Lipton study the effect of importance weighting and show that empirically importance weighting does not have a significant effect when no regularization is applied, which is consistent with the theoretical prediction in [Soudry et al., 2018] that logistical regression without regularization converges to the max margin solution. In our work, we explicitly encourage rare classes to have higher margin, and therefore we don't converge to a max margin solution. Moreover, in our experiments, we apply non-trivial ℓ_2-regularization to achieve the best generalization performance. We also found deferred re-weighting (or deferred re-sampling) are more effective than re-weighting and re-sampling from the beginning of the training.

In contrast, and orthogonally to these papers above, our main technique aims to improve the generalization of the minority classes by applying additional regularization that is orthogonal to the re-weighting scheme. We also propose a deferred re-balancing optimization procedure to improve the optimization and generalization of a generic re-weighting scheme.

Margin loss. The hinge loss is often used to obtain a "max-margin" classifier, most notably in SVMs [Suykens and Vandewalle, 1999]. Recently, Large-Margin Softmax [Liu et al., 2016], Angular Softmax [Liu et al., 2017a], and Additive Margin Softmax [Wang et al., 2018a] have been proposed to minimize intra-class variation in predictions and enlarge the inter-class margin by incorporating the idea of angular margin. In contrast to the class-independent margins in these papers, our approach encourages bigger margins for minority classes. Uneven margins for imbalanced datasets are also proposed and studied in [Li et al., 2002] and the recent work [Khan et al., 2019, Li et al., 2019]. Our theory put this idea on a more theoretical footing by providing a concrete formula for the desired margins of the classes alongside good empirical progress.

Label shift in domain adaptation. The problem of learning imbalanced datasets can be also viewed as a label shift problem in transfer learning or domain adaptation (for which we refer the readers to the survey [Wang and Deng, 2018] and the reference therein). In a typical label shift formulation, the difficulty is to detect and estimate the label shift, and after estimating the label shift, re-weighting or re-sampling is applied. We are addressing a largely different question: can we do better than re-weighting or re-sampling when the label shift is known? In fact, our algorithms can be used to replace the re-weighting steps of some of the recent interesting work on detecting and correcting label shift [Lipton et al., 2018, Azizzadenesheli et al., 2019].

Distributionally robust optimization (DRO) is another technique for domain adaptation (see [Duchi et al., Hashimoto et al., 2018, Carmon et al., 2019] and the reference therein.) However, the formulation assumes no knowledge of the target label distribution beyond a bound on the amount of shift, which makes the problem very challenging. We here assume the knowledge of the test label distribution, using which we design efficient methods that can scale easily to large-scale vision datasets with significant improvements.

Meta-learning. Meta-learning is also used in improving the performance on imbalanced datasets or the few shot learning settings. We refer the readers to [Wang et al., 2017, Shu et al., 2019, Wang et al., 2018b] and the references therein. So far, we generally believe that our approaches that modify the losses are more computationally efficient than meta-learning based approaches.

3 Main Approach

3.1 Theoretical Motivations

Problem setup and notations. We assume the input space is $\mathbb{R}^d$ and the label space is $\{1, \ldots, k\}$. Let x denote the input and y denote the corresponding label. We assume that the class-conditional distribution $\mathcal{P}(x \mid y)$ is the same at training and test time. Let $\mathcal{P}_j$ denote the class-conditional distribution, i.e. $\mathcal{P}_j = \mathcal{P}(x \mid y = j)$. We will use $\mathcal{P}_{\text{bal}}$ to denote the balanced test distribution which first samples a class uniformly and then samples data from $\mathcal{P}_j$.

For a model $f : \mathbb{R}^d \to \mathbb{R}^k$ that outputs k logits, we use $L_{\text{bal}}[f]$ to denote the standard 0-1 test error on the balanced data distribution:

$$L_{\text{bal}}[f] = \Pr_{(x,y)\sim\mathcal{P}_{\text{bal}}}[f(x)_y < \max_{\ell \neq y} f(x)_\ell]$$

Similarly, the error L_j for class j is defined as $L_j[f] = \Pr_{(x,y)\sim\mathcal{P}_j}[f(x)_y < \max_{\ell\neq y} f(x)_\ell]$. Suppose we have a training dataset $\{(x_i, y_i)\}_{i=1}^n$. Let n_j be the number of examples in class j. Let $S_j = \{i : y_i = j\}$ denote the example indices corresponding to class j.

Define the margin of an example (x, y) as

$$\gamma(x, y) = f(x)_y - \max_{j \neq y} f(x)_j \tag{1}$$

Define the training margin for class j as:

$$\gamma_j = \min_{i \in S_j} \gamma(x_i, y_i) \tag{2}$$

We consider the separable cases (meaning that all the training examples are classified correctly) because neural networks are often over-parameterized and can fit the training data well. We also note that the minimum margin of all the classes, $\gamma_{\min} = \min\{\gamma_1, \ldots, \gamma_k\}$, is the classical notion of training margin studied in the past [Koltchinskii et al., 2002].

Fine-grained generalization error bounds. Let $\mathcal{F}$ be the family of hypothesis class. Let $\mathrm{C}(\mathcal{F})$ be some proper complexity measure of the hypothesis class $\mathcal{F}$. There is a large body of recent work on measuring the complexity of neural networks (see [Bartlett et al., 2017, Golowich et al., 2017, Wei and Ma, 2019] and references therein), and our discussion below is orthogonal to the precise choices. When the training distribution and the test distribution are the same, the typical generalization error bounds scale in $\mathrm{C}(\mathcal{F})/\sqrt{n}$. That is, in our case, if the test distribution is also imbalanced as the training distribution, then

$$\text{imbalanced test error} \lesssim \frac{1}{\gamma_{\min}}\sqrt{\frac{\mathrm{C}(\mathcal{F})}{n}} \tag{3}$$

Note that the bound is oblivious to the label distribution, and only involves the minimum margin across all examples and the total number of data points. We extend such bounds to the setting with balanced test distribution by considering the margin of each class. As we will see, the more fine-grained bound below allows us to design new training loss function that is customized to the imbalanced dataset.

Theorem 1 (Informal and simplified version of Theorem 2)**.** *With high probability $(1 - n^{-5})$ over the randomness of the training data, the error L_j for class j is bounded by*

$$L_j[f] \lesssim \frac{1}{\gamma_j}\sqrt{\frac{\mathrm{C}(\mathcal{F})}{n_j}} + \frac{\log n}{\sqrt{n_j}} \tag{4}$$

where we use $\lesssim$ to hide constant factors. As a direct consequence,

$$L_{\text{bal}}[f] \lesssim \frac{1}{k}\sum_{j=1}^{k}\left(\frac{1}{\gamma_j}\sqrt{\frac{\mathrm{C}(\mathcal{F})}{n_j}} + \frac{\log n}{\sqrt{n_j}}\right) \tag{5}$$

Class-distribution-aware margin trade-off. The generalization error bound (4) for each class suggests that if we wish to improve the generalization of minority classes (those with small n_j's), we should aim to enforce bigger margins γ_j's for them. However, enforcing bigger margins for minority classes may hurt the margins of the frequent classes. What is the optimal trade-off between the margins of the classes? An answer for the general case may be difficult, but fortunately we can obtain the optimal trade-off for the binary classification problem.

With $k = 2$ classes, we aim to optimize the balanced generalization error bound provided in (5), which can be simplified to (by removing the low order term $\frac{\log n}{\sqrt{n_j}}$ and the common factor $\mathrm{C}(\mathcal{F})$)

$$\frac{1}{\gamma_1\sqrt{n_1}} + \frac{1}{\gamma_2\sqrt{n_2}} \tag{6}$$

At the first sight, because γ_1 and γ_2 are complicated functions of the weight matrices, it appears difficult to understand the optimal margins. However, we can figure out the relative scales between γ_1 and γ_2. Suppose $\gamma_1, \gamma_2 > 0$ minimize the equation above, we observe that any $\gamma_1' = \gamma_1 - \delta$ and $\gamma_2' = \gamma_2 + \delta$ (for $\delta \in (-\gamma_2, \gamma_1)$) can be realized by the same weight matrices with a shifted bias term (See Figure 1 for an illustration). Therefore, for γ_1, γ_2 to be optimal, they should satisfy

$$\frac{1}{\gamma_1\sqrt{n_1}} + \frac{1}{\gamma_2\sqrt{n_2}} \geq \frac{1}{(\gamma_1 - \delta)\sqrt{n_1}} + \frac{1}{(\gamma_2 + \delta)\sqrt{n_2}} \tag{7}$$

The equation above implies that

$$\gamma_1 = \frac{C}{n_1^{1/4}}, \text{ and } \gamma_2 = \frac{C}{n_2^{1/4}} \tag{8}$$

for some constant C. Please see a detailed derivation in the Section A.

Fast rate vs slow rate, and the implication on the choice of margins. The bound in Theorem 1 may not necessarily be tight. The generalization bounds that scale in $1/\sqrt{n}$ (or $1/\sqrt{n_i}$ here with imbalanced classes) are generally referred to the "slow rate" and those that scale in $1/n$ are referred to the "fast rate". With deep neural networks and when the model is sufficiently big enough, it is possible that some of these bounds can be improved to the fast rate. See [Wei and Ma, 2019] for some recent development. In those cases, we can derive the optimal trade-off of the margin to be $n_i \propto n_i^{-1/3}$.

3.2 Label-Distribution-Aware Margin Loss

Inspired by the trade-off between the class margins in Section 3.1 for two classes, we propose to enforce a class-dependent margin for multiple classes of the form

$$\gamma_j = \frac{C}{n_j^{1/4}} \tag{9}$$

We will design a soft margin loss function to encourage the network to have the margins above. Let (x, y) be an example and f be a model. For simplicity, we use $z_j = f(x)_j$ to denote the j-th output of the model for the j-th class.

The most natural choice would be a multi-class extension of the hinge loss:

$$\mathcal{L}_{\text{LDAM-HG}}((x, y); f) = \max(\max_{j \neq y}\{z_j\} - z_y + \Delta_y, 0) \tag{10}$$

$$\text{where } \Delta_j = \frac{C}{n_j^{1/4}} \text{ for } j \in \{1, \ldots, k\} \tag{11}$$

Here C is a hyper-parameter to be tuned. In order to tune the margin more easily, we effectively normalize the logits (the input to the loss function) by normalizing last hidden activation to ℓ_2 norm 1, and normalizing the weight vectors of the last fully-connected layer to ℓ_2 norm 1, following the previous work [Wang et al., 2018a]. Empirically, the non-smoothness of hinge loss may pose difficulties for optimization. The smooth relaxation of the hinge loss is the following cross-entropy loss with enforced margins:

$$\mathcal{L}_{\text{LDAM}}((x,y);f) = -\log \frac{e^{z_y - \Delta_y}}{e^{z_y - \Delta_y} + \sum_{j \neq y} e^{z_j}} \tag{12}$$

$$\text{where } \Delta_j = \frac{C}{n_j^{1/4}} \text{ for } j \in \{1, \ldots, k\} \tag{13}$$

In the previous work [Liu et al., 2016, 2017a, Wang et al., 2018a] where the training set is usually balanced, the margin Δ_y is chosen to be a label independent constant C, whereas our margin depends on the label distribution.

Remark: Attentive readers may find the loss $\mathcal{L}_{\text{LDAM}}$ somewhat reminiscent of the re-weighting because in the binary classification case — where the model outputs a single real number which is passed through a sigmoid to be converted into a probability, — both the two approaches change the gradient of an example by a scalar factor. However, we remark two key differences: the scalar factor introduced by the re-weighting only depends on the class, whereas the scalar introduced by $\mathcal{L}_{\text{LDAM}}$ also depends on the output of the model; for multiclass classification problems, the proposed loss $\mathcal{L}_{\text{LDAM}}$ affects the gradient of the example in a more involved way than only introducing a scalar factor. Moreover, recent work has shown that, under separable assumptions, the logistical loss, with weak regularization [Wei et al., 2018] or without regularization [Soudry et al., 2018], gives the max margin solution, which is in turn not effected by any re-weighting by its definition. This further suggests that the loss $\mathcal{L}_{\text{LDAM}}$ and the re-weighting may complement each other, as we have seen in the experiments. (Re-weighting would affect the margin in the non-separable data case, which is left for future work.)

3.3 Deferred Re-balancing Optimization Schedule

Cost-sensitive re-weighting and re-sampling are two well-known and successful strategies to cope with imbalanced datasets because, in expectation, they effectively make the imbalanced training distribution closer to the uniform test distribution. The known issues with applying these techniques are (a) re-sampling the examples in minority classes often causes heavy over-fitting to the minority classes when the model is a deep neural network, as pointed out in prior work (e.g., [Cui et al., 2019]), and (b) weighting up the minority classes' losses can cause difficulties and instability in optimization, especially when the classes are extremely imbalanced [Cui et al., 2019, Huang et al., 2016]. In fact, Cui et al. [2019] develop a novel and sophisticated learning rate schedule to cope with the optimization difficulty.

We observe empirically that re-weighting and re-sampling are both inferior to the vanilla empirical risk minimization (ERM) algorithm (where all training examples have the same weight) before annealing the learning rate in the following sense. The features produced before annealing the learning rate by re-weighting and re-sampling are worse than those produced by ERM. (See Figure 6 for an ablation study of the feature quality performed by training linear classifiers on top of the features on a large balanced dataset.)

Inspired by this, we develop a deferred re-balancing training procedure (Algorithm 1), which first trains using vanilla ERM with the LDAM loss before annealing the learning rate, and then deploys a re-weighted LDAM loss with a smaller learning rate. Empirically, the first stage of training leads to a good initialization for the second stage of training with re-weighted losses. Because the loss is non-convex and the learning rate in the second stage is relatively small, the second stage does not move the weights very far. Interestingly, with our LDAM loss and deferred re-balancing training, the vanilla re-weighting scheme (which re-weights by the inverse of the number of examples in each class) works as well as the re-weighting scheme introduced in prior work [Cui et al., 2019]. We also found that with our re-weighting scheme and LDAM, we are less sensitive to early stopping than [Cui et al., 2019].

Algorithm 1 Deferred Re-balancing Optimization with LDAM Loss

Require: Dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$. A parameterized model f_θ

Initialize the model parameters θ randomly
for $t = 1$ to T_0 **do**
  $\mathcal{B} \leftarrow \text{SampleMiniBatch}(\mathcal{D}, m)$ ▷ a mini-batch of m examples
  $\mathcal{L}(f_\theta) \leftarrow \frac{1}{m} \sum_{(x,y)\in\mathcal{B}} \mathcal{L}_{\text{LDAM}}((x, y); f_\theta)$
  $f_\theta \leftarrow f_\theta - \alpha \nabla_\theta \mathcal{L}(f_\theta)$ ▷ one SGD step
  Optional: $\alpha \leftarrow \alpha/\tau$ ▷ anneal learning rate by a factor τ if necessary

for $t = T_0$ to T **do**
  $\mathcal{B} \leftarrow \text{SampleMiniBatch}(\mathcal{D}, m)$ ▷ A mini-batch of m examples
  $\mathcal{L}(f_\theta) \leftarrow \frac{1}{m} \sum_{(x,y)\in\mathcal{B}} n_y^{-1} \cdot \mathcal{L}_{\text{LDAM}}((x, y); f_\theta)$ ▷ standard re-weighting by frequency
  $f_\theta \leftarrow f_\theta - \alpha \frac{1}{\sum_{(x,y)\in\mathcal{B}} n_y^{-1}} \nabla_\theta \mathcal{L}(f_\theta)$ ▷ one SGD step with re-normalized learning rate

4 Experiments

We evaluate our proposed algorithm on artificially created versions of IMDB review [Maas et al., 2011], CIFAR-10, CIFAR-100 [Krizhevsky and Hinton, 2009] and Tiny ImageNet [Russakovsky et al., 2015, tin] with controllable degrees of data imbalance, as well as a real-world large-scale imbalanced dataset, iNaturalist 2018 [Van Horn et al., 2018]. Our core algorithm is developed using PyTorch [Paszke et al., 2017].

Baselines. We compare our methods with the standard training and several state-of-the-art techniques and their combinations that have been widely adopted to mitigate the issues with training on imbalanced datasets: (1) Empirical risk minimization (ERM) loss: all the examples have the same weights; by default, we use standard cross-entropy loss. (2) Re-Weighting (RW): we re-weight each sample by the inverse of the sample size of its class, and then re-normalize to make the weights 1 on average in the mini-batch. (3) Re-Sampling (RS): each example is sampled with probability proportional to the inverse sample size of its class. (4) CB [Cui et al., 2019]: the examples are re-weighted or re-sampled according to the inverse of the effective number of samples in each class, defined as $(1 - \beta^{n_i})/(1 - \beta)$, instead of inverse class frequencies. This idea can be combined with either re-weighting or re-sampling. (5) Focal: we use the recently proposed focal loss [Lin et al., 2017] as another baseline. (6) SGD schedule: by SGD, we refer to the standard schedule where the learning rates are decayed a constant factor at certain steps; we use a standard learning rate decay schedule.

Our proposed algorithm and variants. We test combinations of the following techniques proposed by us. (1) DRW and DRS: following the proposed training Algorithm 1, we use the standard ERM optimization schedule until the last learning rate decay, and then apply re-weighting or re-sampling for optimization in the second stage. (2) LDAM: the proposed Label-Distribution-Aware Margin losses as described in Section 3.2.

When two of these methods can be combined, we will concatenate the acronyms with a dash in between as an abbreviation. The main algorithm we propose is LDAM-DRW. Please refer to Section B for additional implementation details.

4.1 Experimental results on IMDB review dataset

IMDB review dataset consists of 50,000 movie reviews for binary sentiment classification [Maas et al., 2011]. The original dataset contains an evenly distributed number of positive and negative reviews. We manually created an imbalanced training set by removing 90% of negative reviews. We train a two-layer bidirectional LSTM with Adam optimizer [Kingma and Ba, 2014]. The results are reported in Table 1.

Table 1: Top-1 validation errors on imbalanced IMDB review dataset. Our proposed approach LDAM-DRW outperforms the baselines.

Approach	Error on positive reviews	Error on negative reviews	Mean Error
ERM	2.86	70.78	36.82
RS	7.12	45.88	26.50
RW	5.20	42.12	23.66
LDAM-DRW	4.91	30.77	17.84

Table 2: Top-1 validation errors of ResNet-32 on imbalanced CIFAR-10 and CIFAR-100. The combination of our two techniques, LDAM-DRW, achieves the best performance, and each of them individually are beneficial when combined with other losses or schedules.

Dataset	Imbalanced CIFAR-10				Imbalanced CIFAR-100			
Imbalance Type	long-tailed		step		long-tailed		step	
Imbalance Ratio	100	10	100	10	100	10	100	10
ERM	29.64	13.61	36.70	17.50	61.68	44.30	61.45	45.37
Focal [Lin et al., 2017]	29.62	13.34	36.09	16.36	61.59	44.22	61.43	46.54
LDAM	26.65	13.04	33.42	15.00	60.40	43.09	60.42	43.73
CB RS	29.45	13.21	38.14	15.41	66.56	44.94	66.23	46.92
CB RW [Cui et al., 2019]	27.63	13.46	38.06	16.20	66.01	42.88	78.69	47.52
CB Focal [Cui et al., 2019]	25.43	12.90	39.73	16.54	63.98	42.01	80.24	49.98
HG-DRS	27.16	14.03	29.93	14.85	-	-	-	-
LDAM-HG-DRS	24.42	12.72	24.53	12.82	-	-	-	-
M-DRW	24.94	13.57	27.67	13.17	59.49	43.78	58.91	44.72
LDAM-DRW	**22.97**	**11.84**	**23.08**	**12.19**	**57.96**	**41.29**	**54.64**	**40.54**

4.2 Experimental results on CIFAR

Imbalanced CIFAR-10 and CIFAR-100. The original version of CIFAR-10 and CIFAR-100 contains 50,000 training images and 10,000 validation images of size 32×32 with 10 and 100 classes, respectively. To create their imbalanced version, we reduce the number of training examples per class and keep the validation set unchanged. To ensure that our methods apply to a variety of settings, we consider two types of imbalance: long-tailed imbalance [Cui et al., 2019] and step imbalance [Buda et al., 2018]. We use imbalance ratio ρ to denote the ratio between sample sizes of the most frequent and least frequent class, i.e., $\rho = \max_i\{n_i\}/\min_i\{n_i\}$. Long-tailed imbalance follows an exponential decay in sample sizes across different classes. For step imbalance setting, all minority classes have the same sample size, as do all frequent classes. This gives a clear distinction between minority classes and frequent classes, which is particularly useful for ablation study. We further define the fraction of minority classes as μ. By default we set $\mu = 0.5$ for all experiments.

We report the top-1 validation error of various methods for imbalanced versions of CIFAR-10 and CIFAR-100 in Table 2. Our proposed approach is LDAM-DRW, but we also include a various combination of our two techniques with other losses and training schedule for our ablation study.

We first show that the proposed label-distribution-aware margin cross-entropy loss is superior to pure cross-entropy loss and one of its variants tailored for imbalanced data, focal loss, while no data-rebalance learning schedule is applied. We also demonstrate that our full pipeline outperforms the previous state-of-the-arts by a large margin. To further demonstrate that the proposed LDAM loss is essential, we compare it with regularizing by a uniform margin across all classes under the setting of cross-entropy loss and hinge loss. We use M-DRW to denote the algorithm that uses a cross-entropy loss with uniform margin [Wang et al., 2018a] to replace LDAM, namely, the Δ_j in equation (13) is chosen to be a tuned constant that does not depend on the class j. Hinge loss (HG) suffers from optimization issues with 100 classes so we constrain its experiment setting with CIFAR-10 only.

Table 3: Validation errors on iNaturalist 2018 of various approaches. Our proposed method LDAM-DRW demonstrates significant improvements over the previous state-of-the-arts. We include ERM-DRW and LDAM-SGD for the ablation study.

Loss	Schedule	Top-1	Top-5
ERM	SGD	42.86	21.31
CB Focal [Cui et al., 2019]	SGD	38.88	18.97
ERM	DRW	36.27	16.55
LDAM	SGD	35.42	16.48
LDAM	**DRW**	**32.00**	**14.82**

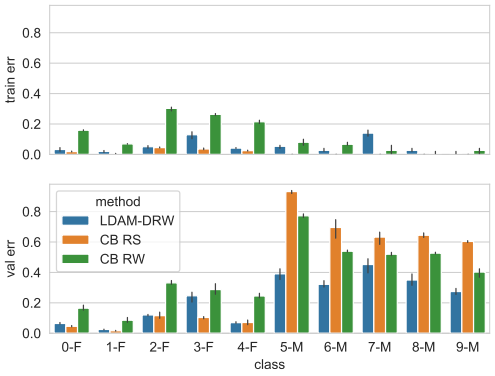


Figure 2: Per-class top-1 error on CIFAR-10 with step imbalance ($\rho = 100, \mu = 0.5$). Classes 0-F to 4-F are frequent classes, and the rest are minority classes. Under this extremely imbalanced setting RW suffers from under-fitting, while RS over-fits on minority examples. On the contrary, the proposed algorithm exhibits great generalization on minority classes while keeping the performance on frequent classes almost unaffected. This suggests we succeeded in regularizing minority classes more strongly.

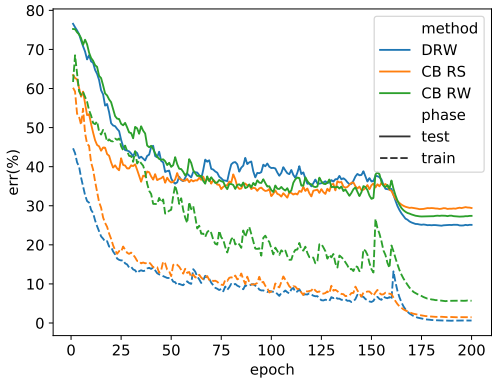


Figure 3: Imbalanced training errors (dotted lines) and *balanced* test errors (solid lines) on CIFAR-10 with long-tailed imbalance ($\rho = 100$). We anneal decay the learning rate at epoch 160 for all algorithms. Our DRW schedule uses ERM before annealing the learning rate and thus performs worse than RW and RS before that point, as expected. However, it outperforms the others significantly after annealing the learning rate. See Section 4.4 for more analysis.

Imbalanced but known test label distribution: We also test the performance of an extension of our algorithm in the setting where the test label distribution is known but not uniform. Please see Section C.5 for details.

4.3 Visual recognition on iNaturalist 2018 and imbalanced Tiny ImageNet

We further verify the effectiveness of our method on large-scale imbalanced datasets. The iNatualist species classification and detection dataset [Van Horn et al., 2018] is a real-world large-scale imbalanced dataset which has 437,513 training images with a total of 8,142 classes in its 2018 version. We adopt the official training and validation splits for our experiments. The training datasets have a long-tailed label distribution and the validation set is designed to have a balanced label distribution. We use ResNet-50 as the backbone network across all experiments for iNaturalist 2018. Table 3 summarizes top-1 validation error for iNaturalist 2018. Notably, our full pipeline is able to outperform the ERM baseline by 10.86% and previous state-of-the-art by 6.88% in top-1 error. Please refer to Appendix C.2 for results on imbalanced Tiny ImageNet.

4.4 Ablation study

Evaluating generalization on minority classes. To better understand the improvement of our algorithms, we show per-class errors of different methods in Figure 2 on imbalanced CIFAR-10. Please see the caption there for discussions.

Evaluating deferred re-balancing schedule. We compare the learning curves of deferred re-balancing schedule with other baselines in Figure 3. In Figure 6 of Section C.3, we further show that even though ERM in the first stage has slightly worse or comparable balanced test error compared to RW and RS, in fact the features (the last-but-one layer activations) learned by ERM are better than those by RW and RS. This agrees with our intuition that the second stage of DRW, starting from better features, adjusts the decision boundary and locally fine-tunes the features.

5 Conclusion

We propose two methods for training on imbalanced datasets, label-distribution-aware margin loss (LDAM), and a deferred re-weighting (DRW) training schedule. Our methods achieve significantly improved performance on a variety of benchmark vision tasks. Furthermore, we provide a theoretically-principled justification of LDAM by showing that it optimizes a uniform-label generalization error bound. For DRW, we believe that deferring re-weighting lets the model avoid the drawbacks associated with re-weighting or re-sampling until after it learns a good initial representation (see some analysis in Figure 3 and Figure 6). However, the precise explanation for DRW's success is not fully theoretically clear, and we leave this as a direction for future work.

Acknowledgements Toyota Research Institute ("TRI") provided funds and computational resources to assist the authors with their research but this article solely reflects the opinions and conclusions of its authors and not TRI or any other Toyota entity. We thank Percy Liang and Michael Xie for helpful discussions in various stages of this work.

References

Tiny imagenet visual recognition challenge. URL https://tiny-imagenet.herokuapp.com.

Sanjeev Arora, Rong Ge, Behnam Neyshabur, and Yi Zhang. Stronger generalization bounds for deep nets via a compression approach. *arXiv preprint arXiv:1802.05296*, 2018.

Kamyar Azizzadenesheli, Anqi Liu, Fanny Yang, and Animashree Anandkumar. Regularized learning for domain adaptation under label shifts. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rJl0r3R9KX.

Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems*, pages 6240–6249, 2017.

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.

Jonathon Byrd and Zachary Lipton. What is the effect of importance weighting in deep learning? In *International Conference on Machine Learning*, 2019.

Kaidi Cao, Yu Rong, Cheng Li, Xiaoou Tang, and Chen Change Loy. Pose-robust face recognition via deep residual equivariant mapping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5187–5196, 2018.

Yair Carmon, Yujia Jin, Aaron Sidford, and Kevin Tian. Variance reduction for matrix games. *arXiv preprint arXiv:1907.02056*, 2019.

Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

John C Duchi, Tatsunori Hashimoto, and Hongseok Namkoong. Distributionally robust losses against mixture covariate shifts.

Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

Noah Golowich, Alexander Rakhlin, and Ohad Shamir. Size-independent sample complexity of neural networks. *arXiv preprint arXiv:1712.06541*, 2017.

Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *European Conference on Computer Vision*, pages 87–102. Springer, 2016.

Tatsunori Hashimoto, Megha Srivastava, Hongseok Namkoong, and Percy Liang. Fairness without demographics in repeated loss minimization. In *International Conference on Machine Learning*, pages 1934–1943, 2018.

Haibo He and Edwardo A Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge & Data Engineering*, (9):1263–1284, 2008.

Haibo He and Yunqian Ma. *Imbalanced learning: foundations, algorithms, and applications*. John Wiley & Sons, 2013.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

J Henry Hinnefeld, Peter Cooman, Nat Mammo, and Rupert Deese. Evaluating fairness metrics in the presence of dataset bias. *arXiv preprint arXiv:1809.09245*, 2018.

Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. Learning deep representation for imbalanced classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5375–5384, 2016.

Chen Huang, Yining Li, Change Loy Chen, and Xiaoou Tang. Deep imbalanced learning for face recognition and attribute prediction. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

Nathalie Japkowicz and Shaju Stephen. The class imbalance problem: A systematic study. *Intelligent data analysis*, 6(5):429–449, 2002.

Sham M Kakade, Karthik Sridharan, and Ambuj Tewari. On the complexity of linear prediction: Risk bounds, margin bounds, and regularization. In *Advances in neural information processing systems*, pages 793–800, 2009.

Salman Khan, Munawar Hayat, Syed Waqas Zamir, Jianbing Shen, and Ling Shao. Striking the right balance with uncertainty. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 103–112, 2019.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Vladimir Koltchinskii, Dmitry Panchenko, et al. Empirical margin distributions and bounding the generalization error of combined classifiers. *The Annals of Statistics*, 30(1):1–50, 2002.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73, 2017.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Buyu Li, Yu Liu, and Xiaogang Wang. Gradient harmonized single-stage detector. *arXiv preprint arXiv:1811.05181*, 2018.

Yaoyong Li, Hugo Zaragoza, Ralf Herbrich, John Shawe-Taylor, and Jaz Kandola. The perceptron algorithm with uneven margins. In *ICML*, volume 2, pages 379–386, 2002.

Zeju Li, Konstantinos Kamnitsas, and Ben Glocker. Overfitting of neural nets under class imbalance: Analysis and improvements for segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 402–410. Springer, 2019.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

Zachary Lipton, Yu-Xiang Wang, and Alexander Smola. Detecting and correcting for label shift with black box predictors. In *International Conference on Machine Learning*, pages 3128–3136, 2018.

Weiyang Liu, Yandong Wen, Zhiding Yu, and Meng Yang. Large-margin softmax loss for convolutional neural networks. In *ICML*, volume 2, page 7, 2016.

Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 212–220, 2017a.

Yu Liu, Hongyang Li, and Xiaogang Wang. Rethinking feature discrimination and polymerization for large-scale recognition. *arXiv preprint arXiv:1710.00870*, 2017b.

Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2537–2546, 2019.

Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*, pages 142–150. Association for Computational Linguistics, 2011.

Michele Merler, Nalini Ratha, Rogerio S Feris, and John R Smith. Diversity in faces. *arXiv preprint arXiv:1901.10436*, 2019.

Vaishnavh Nagarajan and Zico Kolter. Deterministic PAC-bayesian generalization bounds for deep networks via generalizing noise-resilience. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Hygn2o0qKX.

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. 2017.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

Li Shen, Zhouchen Lin, and Qingming Huang. Relay backpropagation for effective learning of deep convolutional neural networks. In *European conference on computer vision*, pages 467–482. Springer, 2016.

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. *arXiv preprint arXiv:1902.07379*, 2019.

Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *The Journal of Machine Learning Research*, 19(1):2822–2878, 2018.

Johan AK Suykens and Joos Vandewalle. Least squares support vector machine classifiers. *Neural processing letters*, 9(3):293–300, 1999.

Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. Yfcc100m: The new data in multimedia research. *arXiv preprint arXiv:1503.01817*, 2015.

Grant Van Horn and Pietro Perona. The devil is in the tails: Fine-grained classification in the wild. *arXiv preprint arXiv:1709.01450*, 2017.

Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8769–8778, 2018.

Feng Wang, Jian Cheng, Weiyang Liu, and Haijun Liu. Additive margin softmax for face verification. *IEEE Signal Processing Letters*, 25(7):926–930, 2018a.

Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018.

Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Learning to model the tail. In *Advances in Neural Information Processing Systems*, pages 7029–7039, 2017.

Yu-Xiong Wang, Ross Girshick, Martial Hebert, and Bharath Hariharan. Low-shot learning from imaginary data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7278–7286, 2018b.

Colin Wei and Tengyu Ma. Data-dependent Sample Complexity of Deep Neural Networks via Lipschitz Augmentation. *arXiv e-prints*, art. arXiv:1905.03684, May 2019.

Colin Wei and Tengyu Ma. Improved sample complexities for deep networks and robust classification via an all-layer margin. *arXiv preprint arXiv:1910.04284*, 2019.

Colin Wei, Jason D Lee, Qiang Liu, and Tengyu Ma. On the margin theory of feedforward neural networks. *arXiv preprint arXiv:1810.05369*, 2018.

Q Zhong, C Li, Y Zhang, H Sun, S Yang, D Xie, and S Pu. Towards good practices for recognition & detection. In *CVPR workshops*, 2016.

Yang Zou, Zhiding Yu, BVK Kumar, and Jinsong Wang. Domain adaptation for semantic segmentation via class-balanced self-training. *arXiv preprint arXiv:1810.07911*, 2018.

A Missing Proofs and Derivations in Section 3.1

Let $L_{\gamma,j}$ denote the hard margin loss on examples from class j:

$$L_{\gamma,j}[f] = \Pr_{x\sim\mathcal{P}_j}[\max_{j'\neq j} f(x)_{j'} > f(x)_j - \gamma]$$

and let $\hat{L}_{\gamma,j}$ denote its empirical variant. For a hypothesis class $\mathcal{F}$, let $\hat{\mathfrak{R}}_j(\mathcal{F})$ denote the empirical Rademacher complexity of its class j margin:

$$\hat{\mathfrak{R}}_j(\mathcal{F}) = \frac{1}{n_j}\mathbb{E}_\sigma\left[\sup_{f\in\mathcal{F}}\sum_{i\in S_j}\sigma_i[f(x_i)_j - \max_{j'\neq j} f(x_i)_{j'}]\right]$$

where σ is a vector of i.i.d. uniform $\{-1,+1\}$ bits. The following formal versiom of Theorem 1 bounds the balanced-class generalization $\mathcal{P}_{\text{bal}}$ using samples from $\mathcal{P}$.

Theorem 2. *With probability $1-\delta$ over the randomness of the training data, for all choices of class-dependent margins $\gamma_1,\ldots,\gamma_k > 0$, all hypotheses $f\in\mathcal{F}$ will have balanced-class generalization bounded by*

$$L_{\text{bal}}[f] \le \frac{1}{k}\left(\sum_{j=1}^{k}\hat{L}_{\gamma_j,j}[f] + \frac{4}{\gamma_j}\hat{\mathfrak{R}}_j(\mathcal{F}) + \epsilon_j(\gamma_j)\right)$$

where $\epsilon_j(\gamma) \triangleq \sqrt{\frac{\log\log_2(\frac{2\max_{x\in\mathcal{X},f\in\mathcal{F}}|f(x)|}{\gamma}) + \log\frac{2c}{\delta}}{n_j}}$ is typically a low-order term in n_j. Concretely, the Rademacher complexity $\hat{\mathfrak{R}}_j(\mathcal{F})$ will typically scale as $\sqrt{\frac{\mathrm{C}(\mathcal{F})}{n_j}}$ for some complexity measure $\mathrm{C}(\mathcal{F})$, in which case

$$L_{\text{bal}}[f] \le \frac{1}{k}\left(\sum_{j=1}^{k}\hat{L}_{\gamma_j,j}[f] + \frac{4}{\gamma_j}\sqrt{\frac{\mathrm{C}(\mathcal{F})}{n_j}} + \epsilon_j(\gamma_j)\right)$$

Proof. We will prove generalization separately for each class j and then union bound over all classes.

Let $L_j[f]$ denote the test $0-1$ error of classifier f on examples drawn from $\mathcal{P}_j$. As the examples for class j is a set of n_j i.i.d. draws from the conditional distribution $\mathcal{P}_j$, we can apply the standard margin-based generalization bound (Theorem 2 of [Kakade et al., 2009]) to obtain with probability $1-\delta/c$, for all choices of $\gamma_j > 0$ and $f\in\mathcal{F}$,

$$L_j[f] \le \hat{L}_{\gamma_j,j} + \frac{4}{\gamma_j}\hat{\mathfrak{R}}_j(\mathcal{F}) + \sqrt{\frac{\log\log_2(\frac{2\max_{x\in\mathcal{X},f\in\mathcal{F}}|f(x)|}{\gamma_j})}{n_j}} + \sqrt{\frac{\log\frac{2c}{\delta}}{n_j}} \tag{14}$$

Now since $L_{\text{bal}} = \frac{1}{k}\sum_{j=1}^{k} L_j$, we can union bound over all classes and average (14) to get the desired result. □

We will now show that in the case of $k=2$ classes, it is always possible to shift the margins in order to optimize the generalization bound of Theorem 2 by adding bias terms.

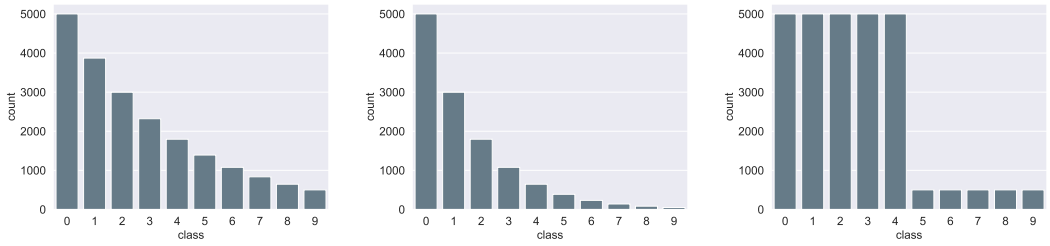
Theorem 3. *For binary classification, let $\mathcal{F}$ be a hypothesis class of neural networks with a bias term, i.e. $\mathcal{F} = \{f+b\}$ where f is a neural net function and $b\in\mathbb{R}^2$ is a bias, with Rademacher complexity upper bound $\hat{\mathfrak{R}}_j(\mathcal{F}) \le \sqrt{\frac{\mathrm{C}(\mathcal{F})}{n_j}}$. Suppose some classifier $f\in\mathcal{F}$ can achieve a total sum of margins $\gamma_1' + \gamma_2' = \beta$ with $\gamma_1',\gamma_2' > 0$. Then there exists a classifier $f^\star\in\mathcal{F}$ with margins*

$$\gamma_1^\star = \frac{\beta n_2^{1/4}}{n_1^{1/4}+n_2^{1/4}}\ ,\gamma_2^\star = \frac{\beta n_1^{1/4}}{n_1^{1/4}+n_2^{1/4}}$$

which with probability $1-\delta$ obtains the optimal generalization guarantees for Theorem 2:

$$L_{\text{bal}}[f^\star] \le \min_{\gamma_1+\gamma_2=\beta}\left(\frac{2}{\gamma_1}\sqrt{\frac{\mathrm{C}(\mathcal{F})}{n_1}} + \frac{2}{\gamma_2}\sqrt{\frac{\mathrm{C}(\mathcal{F})}{n_2}}\right) + \epsilon(\gamma_1^\star) + \epsilon(\gamma_2^\star)$$

where ϵ is defined in Theorem 2. Furthermore, this $f^\star$ is obtained via $f+b^\star$ for some bias $b^\star$.



(a) $\rho = 10$ (b) $\rho = 100$ (c) $\rho = 10,\ \mu = 0.5$

Figure 4: Number of training examples per class in artificially created imbalanced CIFAR-10 datasets. Fig. 4a and Fig. 4b belong to long-tailed imbalance type and Fig. 4c is a step imbalance distribution.

Proof. For our bias $b^\star$, we simply choose $b_1^\star = (\gamma_1^\star - \gamma_1')/2$, $b_2^\star = -(\gamma_1^\star - \gamma_1')/2$. Now note that adding a bias term simply shifts the margins for class 1 by $b_1^\star - b_2^\star$, giving a new margin of $\gamma_2^\star$. Likewise, the margin for class 2 becomes

$$b_2^\star - b_1^\star + \gamma_2' = \gamma_2' - \gamma_1^\star + \gamma_1' = \beta - \gamma_1^\star = \gamma_2^\star$$

Now we apply Theorem 2 to get with probability $1 - \delta$ the generalization error bound

$$L_{\text{bal}}[f^\star] \leq \frac{2}{\gamma_1^\star}\sqrt{\frac{\mathrm{C}(\mathcal{F})}{n_1}} + \frac{2}{\gamma_2^\star}\sqrt{\frac{\mathrm{C}(\mathcal{F})}{n_2}} + \epsilon(\gamma_1^\star) + \epsilon(\gamma_2^\star)$$

To see that $\gamma_1^\star, \gamma_2^\star$ indeed solve

$$\min_{\gamma_1+\gamma_2=\beta} \frac{1}{\gamma_1}\sqrt{\frac{1}{n_1}} + \frac{1}{\gamma_2}\sqrt{\frac{1}{n_2}}$$

we can substitute $\gamma_2 = \beta - \gamma_1$ into the expression and set the derivative to 0, obtaining

$$\frac{1}{(\beta-\gamma_1)^2\sqrt{n_2}} - \frac{1}{\gamma_1^2\sqrt{n_1}} = 0$$

Solving gives $\gamma_1^\star$. □

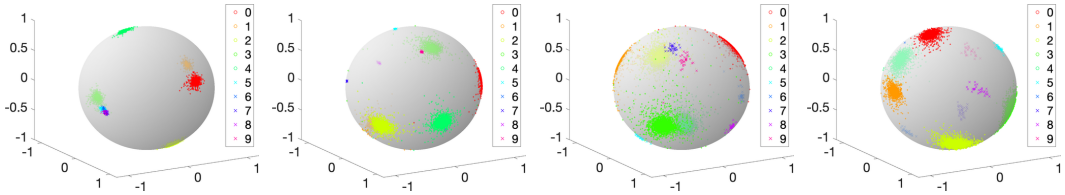
B Implementation details

Label distributions. Some example distributions of our artificially created imbalance are shown in Figure 4.

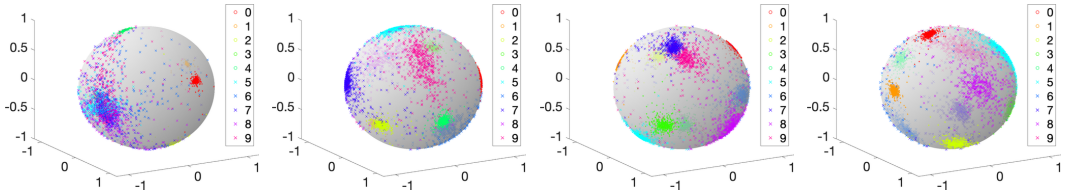
Implementation details for CIFAR. For CIFAR-10 and CIFAR-100, we follow the simple data augmentation in [He et al., 2016] for training: 4 pixels are padded on each side, and a 32×32 crop is randomly sampled from the padded image or its horizontal flip. We use ResNet-32 [He et al., 2016] as our base network, and use stochastic gradient descend with momentum of 0.9, weight decay of 2×10^{-4} for training. The model is trained with a batch size of 128 for 200 epochs. For fair comparison, we use an initial learning rate of 0.1, then decay by 0.01 at the 160th epoch and again at the 180th epoch. We also use linear warm-up learning rate schedule [Goyal et al., 2017] for the first 5 epochs for fair comparison. Notice that the warm-up trick is essential for the training of re-weighting, but it won't affect other algorithms in our experiments. We tune C to normalize Δ_j so that the largest enforced margin is 0.5.

Implementation details for Tiny ImageNet. For Tiny ImageNet, we perform simple horizontal flips, taking random crops of size 64×64 from images padded by 8 pixels on each side. We perform 1 crop test with the validation images. We use ResNet-18 [He et al., 2016] as our base network, and use stochastic gradient descend with momentum of 0.9, weight decay of 2×10^{-4} for training. We train the model using a batch size of 128 for 120 epochs with a initial learning rate of 0.1. We decay the learning rate by 0.1 at epoch 90. We tune C to normalize Δ_j so that the largest enforced margin is 0.5.

Implementation details for iNaturalist 2018. On iNaturalist 2018, we followed the same training strategy used by [He et al., 2016] and trained ResNet-50 with 4 Tesla V100 GPUs. Each image is first



(a) ERM train (b) Re-sampling train (c) Re-weighting train (d) LDAM train



(e) ERM val (f) Re-sampling val (g) Re-weighting val (h) LDAM val

Figure 5: Visualization of feature distribution of different methods. We constrain the feature dimension to be three and normalize it for better illustration. The top row has the feature distribution on the training set and the second row the feature distributions on the validation set. We can see that LDAM appears to have more separate training features compared to the other methods. We note this visualization is only supposed to provide qualitative intuitions, and the differences between our methods and other methods may be more significant for harder tasks with higher feature dimension. (For example, here the accuracies of re-weighting and LDAM are very similar, whereas for large-scale datasets with higher feature dimensions, the gap is significantly larger.)

Table 4: Validation error on imbalanced Tiny ImageNet with different loss functions and training schedules.

Imbalance Type		long-tailed				step			
Imbalance Ratio		100		10		100		10	
Loss	Schedule	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
ERM	SGD	66.19	42.63	50.33	26.68	63.82	44.09	50.89	27.06
CB SM	SGD	72.72	52.62	51.58	28.91	74.90	59.14	54.51	33.23
ERM	DRW	64.57	40.79	50.03	26.19	62.36	40.84	49.17	25.91
LDAM	SGD	64.04	40.46	48.08	24.80	62.54	39.27	49.08	24.52
LDAM	DRW	**62.53**	**39.06**	**47.22**	**23.84**	**60.63**	**38.12**	**47.43**	**23.26**

resized by setting the shorter side to 256 pixels, and then a 224×224 crop is randomly sampled from an image or its horizontal flip. We train the network for 90 epochs with an initial learning rate of 0.1. We anneal the learning rate at epoch 30 and 60. For our two-stage training schedule, we rebalance the training data starting from epoch 60. We tune C to normalize Δ_j so that the largest enforced margin is 0.3.

C Additional Results

C.1 Feature visualization

To have a better understanding of our proposed LDAM loss, we use a toy example to visualize feature distributions trained under different schemes. We train a 7-layer CNN as adopted in [Liu et al., 2017b] on MNIST [LeCun et al., 1998] with step imbalance setting ($\rho = 100, \mu = 0.5$). For a more intuitive visualization, we constrain the feature dimension to 3 and normalize the feature before feeding it into the final fully-connected layer, allowing us to scatter the features on a unit hyper-sphere in a 3D frame. The visualization is shown in Figure 5 with additional discussion in the caption.

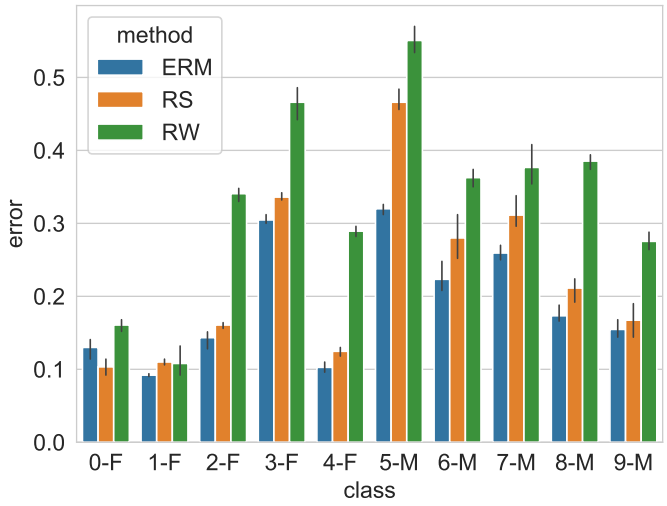


Figure 6: In the setting of training mbalanced CIFAR-10 dataset with step imbalance of $\rho = 100, \mu = 0.5$, to test the quality of the features obtained by the ERM, RW and RS before annealing the learning rate, we use a subset of the *balanced* validation dataset to train linear classifiers on top of the features, and evaluate the per-class validation error on the rest of the validation data. (Little over-fitting in training the linear classifier is observed.) The left-5 classes are frequent and denoted with -F. The features obtained from ERM setting has the strongest performance, confirming our intuition that the second stage of DRW starts from better features. In the second stage, DRW re-weights the example again, adjusting the decision boundary and locally fine-tuning the features.

C.2 Visual Recognition on imbalanced Tiny ImageNet

In addition to artificial imbalanced CIFAR, we further verify the effectiveness of our method on artificial imbalanced Tiny ImageNet. The Tiny ImageNet dataset has 200 classes. Each class has 500 training images and 50 validation images of size 64×64. We use the same strategy described above to create long-tailed and step imbalance versions of Tiny ImageNet. The results are presented in Table 4. While Class-Balanced Softmax performs worse than the ERM baseline, the proposed LDAM and DRW demonstrate consistent improvements over ERM.

C.3 Comparing feature extractors trained by different schemes

As discussed in Section 4.4, we train a linear classifier on features extracted by backbone filters pretrained under different schemes. We could conclude that for highly imbalanced settings (step imbalance with $\rho = 100, \mu = 0.5$), backbone networks trained by ERM learns the most expressive feature embedding compared with the other two methods, as shown in Figure 6.

C.4 Comparing DRW and DRS

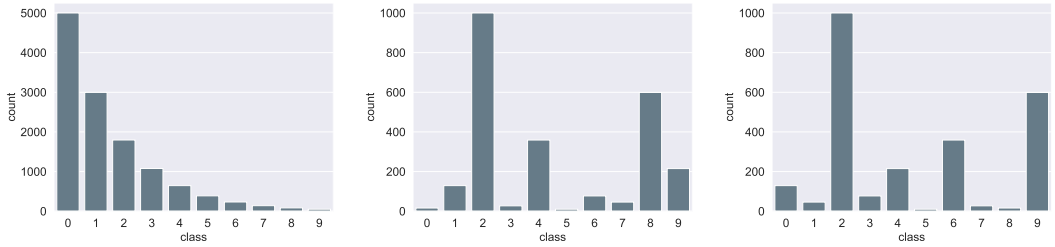
Our proposed deferred re-balancing optimization schedule can be combined with either re-weighting or re-sampling. We use re-weighting as the default choice in the main paper. Here we demonstrate through Table 5 that re-weighting and re-sampling exhibit similar performance when combined with deferred re-balancing scheme. This result could be explained by the fact that the second stage does not move the weights far. Re-balancing in the second stage mostly re-adjusts the decision boundary and thus there is no significant difference between using re-weighting or re-sampling for the second stage.

C.5 Imbalanced Test Label Distributions

Though the majority of our experiments follow the uniform test distribution setting, it could be extended to imbalanced test distribution naturally. Suppose the number of training examples in class i is denoted by n_i and the number of test examples in class i is denoted by n'_i, then we could adapt

Table 5: Top-1 validation error of ResNet-32 trained with different training schedules on imbalanced CIFAR-10 and CIFAR-100.

Dataset Name	Imbalanced CIFAR-10				Imbalanced CIFAR-100			
Imbalance Type	long-tailed		step		long-tailed		step	
Imbalance Ratio	100	10	100	10	100	10	100	10
ERM	29.64	13.61	36.70	17.50	61.68	44.30	61.05	45.37
DRW	25.14	13.12	28.40	14.49	59.34	42.68	58.86	42.78
DRS	25.50	13.28	27.97	14.83	59.67	42.74	58.65	43.21



(a) train set distribution (b) val set distribution 1 (c) val set distribution 2

Figure 7: Example distributions when train and test distributions are both imbalanced. As discussed in C.5 we run two random seeds for generating test distributions. Here Figure 7b denotes the left column in Table 6.

the LDAM simply by encouraging the margin Δ_i for class i with

$$\Delta_j \propto \left(\frac{n'_i}{n_i}\right)^{1/4} \tag{15}$$

To complement our main result, In Table 6, we demonstrate that this extended algorithm can also work well when the test distribution is imbalanced. We use the same rule as described in Section 4 to generate imbalanced test label distribution and then permute randomly the frequency of the labels (so that the training label distribution is very different from the test label distribution.). For example, in the experiment shown in Figure 6, the training label distribution of the column of "long-tailed with $\rho = 100$" follows Figure 7a (which is the same as Figure 4b) whereas the test label distribution is shown in Figure 7b and Figure 7c. For each of the settings reported in Table 6, we have run it with two different random seeds for generating the test label distribution, and we see qualitatively similar results. We refer to our code for the precise label distribution generated in the experiments.[3]

[3]Code available at https://github.com/kaidic/LDAM-DRW.

Table 6: Top-1 validation error of ResNet-32 on imbalanced training and imbalanced validation scheme for CIFAR-10. See Section C.5 for details.

Imbalance Type	long-tailed				step			
Imbalance Ratio Train	100		10		100		10	
Imbalance Ratio Val	100	100	10	10	100	100	10	10
ERM	30.99	28.45	13.08	13.12	24.55	28.63	10.34	11.67
CB-RW	20.86	26.19	10.70	11.93	35.76	31.35	9.82	11.02
LDAM-DRW	**14.40**	**12.95**	**10.12**	**10.62**	**10.30**	**9.54**	**7.51**	**7.82**