Attentional-Biased Stochastic Gradient Descent

Anonymous authors
Paper under double-blind review

Abstract

In this paper, we present a simple yet effective systematic method (named ABSGD) for addressing the data imbalance or label noise problem in deep learning. Our method is a simple modification to momentum SGD where we assign an individual importance weight to each sample in the mini-batch. The individual-level weight of a sampled data is systematically proportional to the exponential of a scaled loss value of the data, where the scaling factor is interpreted as the regularization parameter in the framework of distributionally robust optimization (DRO). Depending on whether the scaling factor is positive or negative, ABSGD is guaranteed to converge to a stationary point of an information-regularized min-max or min-min DRO problem, respectively. Compared with existing class-level weighting schemes, our method can capture the diversity between individual examples within each class. Compared with existing individual-level weighting methods using meta-learning that require three backward propagations for computing mini-batch stochastic gradients, our method is more efficient with only one backward propagation at each iteration as in standard deep learning methods. ABSGD is flexible enough to combine with other robust losses without any additional cost. Our empirical studies on several benchmark datasets demonstrate the effectiveness of the proposed method.

1 Introduction

Deep Learning (DL) has emerged as the most popular machine learning technique in recent years. It has brought transformative impact in industries and quantum leaps in the quality of a wide range of everyday technologies including face recognition (1; 2; 3; 4; 5), speech recognition (6; 7; 8; 9; 10) and machine translation (11; 12; 13; 14; 15). Most of these systems are built based on learning a deep neural network (DNN) model from a huge amount of data. However, it has been observed that these deep learning systems could fail in some cases caused by undesirable data distribution, such as data imbalance (16; 17; 18; 19; 20) and noisy labels in the dataset (21; 22; 23). To be more specific, for example, Apple's FaceID (a face recognition system) is much less accurate for recognizing a child than an adult (24), and an autonomous driving car might fail at night under the same road condition (25). The key factors that cause these problems are (i) the training data sets collected from the real-world are usually follows a highly skewed distribution (e.g., the number of facial images of children are much less than that of adults), and/or contain noisily labelled samples due to the inaccurate annotation process (26); (ii) current deep learning systems are not robust enough to overcome negative influence incurred by the real-world imperfect data as most existing deep learning techniques in the literature are crafted and evaluated on well-designed benchmark datasets with balanced distributions among different classes (e.g., ImageNet data for image classification).

Extensive studies have been explored to learn a robust model to overcome the above deficiencies. Those stuides can be divided into two directions, data manipulation and robust learning. Popular data manipulation methods include under/over-sampling based approaches (27; 28; 29) for data imbalance problem and label correction methods (30; 31; 32; 33) for label noise problem, etc. Existing studies in these methods are not very successful for deep learning with big data. For example, several studies have found that over-sampling yields better performance than using under-sampling (16). But over-sampling will add more examples to the training data, which will lead to increased training time. While the label correction methods (30; 31; 32; 33) usually require extra clean data that are expensive to collect.

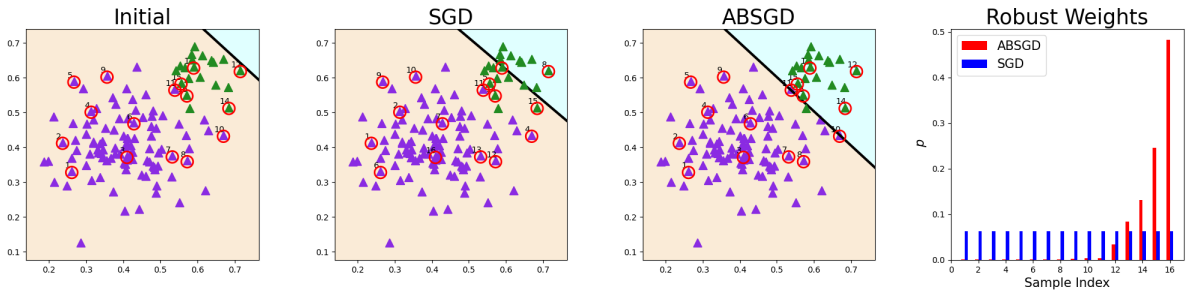

Figure 1: A two-dimensional linear binary classification toy example with logistic loss. We optimize the model for one iteration using SGD with different weighting schemes for the same stochastic minibatch. The classifier generated in iteration 1 vividly demonstrates the effectiveness of attentional biased SGD, compared to not doing it, on handling data imbalance.

Robust learning methods include robust weighting and robust loss, where robust weighting assigns weights to different losses of individual data which are either hand-crafted or learned, and robust loss refers to new loss functions that are heuristic-driven or theoretically inspired for addressing data imbalance or label noise issues. The existing robust weighting methods either require significant tuning or suffer from significant computational burden. In this paper, we propose a simple yet systematic **attentional-biased stochastic gradient descent** (ABSGD) method for addressing the class imbalance or the label noise problem in a unified framework, which falls in the category of robust weighting methods. ABSGD is a simple modification of the popular momentum SGD method for deep learning by injecting individual-level importance weights to stochastic gradients in the mini-batch. These importance weights allow our method either to focus on examples from the minority classes for the data imbalance problem or the clean samples for the label noise problem. This idea is illustrated in Figure 1 on a toy imbalanced dataset by comparing it with the standard momentum SGD method for deep learning. Unlike existing meta-learning methods for learning individual-level weights, our individual-level weights are self-adaptive that are computed based on the loss value of each individual data. In particular, the weight for each example is proportional to exponential of a scaled loss value on that example. The weighting scheme is **grounded in the theoretically justifiable distributionally robust optimization (DRO) framework**.

Specifically, our method can be considered as a stochastic momentum method for solving an information-regularized distributionally robust optimization (IR-DRO) problem defined on all possible data (34). From this perspective, our method has several unique features. (i) The weights for all examples in the mini-batch have a proper normalization term to ensure the method optimizes the IR-DRO problem, which is updated online. We prove a theorem to show that our method converges to a stationary solution of the non-convex IR-DRO problem (with a certain convergence rate). (ii) The scaling factor before the loss value in the exponential function is interpreted as the regularization parameter in the DRO framework. In addition, our method has two benefits: (i) it is applicable in online learning, where the data is received sequentially; (ii) it is loss independent, and can be combined with all existing loss functions crafted for tackling data imbalance and label noise. Finally, we summarize **our contributions** below:

- We propose a simple robust stochastic gradient descent method with momentum and self-adaptive importance weighting to tackle deep learning tasks with imbalanced data or label noise, which is named as **ABSGD**. ABSGD can be generalized to a broader family of AB methods that employ other updating methods, e.g., AB-ADAM that uses the ADAM scheme to update the model parameter.

- We prove that ABSGD finds a stationary solution of a non-convex IR-DRO problem for learning a deep neural network, and establish its convergence rate.

- We compare ABSGD with a variety of existing techniques for addressing the data imbalance and label noise problems, including crafted loss functions, class-balancing weighting methods, individual-level weighting meta-learning methods, and demonstrate superb performance of ABSGD.

2 Related Work

Class-level Weighting. The idea of class-level weighting is to introduce weights to examples at the class level to balance the contributions from different classes. This idea is rooted in cost-sensitive classification in machine learning (35; 36; 37; 38; 39; 40; 41; 42). Traditional cost-sensitive methods typically tune the class-level weights. Recently, a popular approach is to set the class-wise weights to be proportional to the inverse of class sizes (43; 44). (45) proposed an improved class-level weighting scheme according to inverse of the "effective number" of examples per class. It is also notable that over/under-sampling methods have the same effect of introducing the class-level weighting to the training algorithm. We can see that these class-level weighting schemes usually require certain knowledge about the size (distribution) of each class, which makes them not suitable to online learning where the size of each class is not known beforehand. These methods also neglect the differences between different examples from the same class (cf. Figure 1).

Individual-weighting by Meta-Learning. The individual-level weights learning methods typically use meta-learning to learn the individual-level weights along with updating the model parameters (46; 47). The idea is to learn individual-level weights by solving a two-level optimization problem. In particular,

$$\min_{\theta} \frac{1}{|\mathcal{C}|} \sum_{\mathbf{z}_i \in \mathcal{C}} L(\mathbf{w}(\theta); \mathbf{z}_i), \quad \text{where } \mathbf{w}(\theta) = \arg\min_{\mathbf{w}} \frac{1}{|\mathcal{D}|} \sum_{\mathbf{z}_i \in \mathcal{D}} \theta_i L(\mathbf{w}; \mathbf{z}_i)$$

where $\mathcal{D}$ denotes the training dataset, $\mathcal{C}$ denotes a balanced validation dataset, $\mathbf{w}$ denotes the model parameter, $\mathbf{z}_i$ denotes a data, $L(\mathbf{w}; \mathbf{z})$ denotes the loss value of model $\mathbf{w}$ on data $\mathbf{z}$, and $\theta = (\theta_1, \ldots, \theta_{|\mathcal{D}|})$ denotes the weights on the training examples. (47) directly optimized the individual weights in the framework of meta-learning with a heuristic trick by normalizing the weights of all examples in a training batch so that they sum up to one. (46) considered the problem from the perspective of domain adaptation and decomposed the individual weight into sum of a non-learnable class-level weight and a learnable individual-level weight. One issue of these meta-learning methods is that they require three back-propagations at each iteration, which is computationally more expensive than our method that is about the same cost of standard SGD for DL.

Crafted Individual Loss Functions. Some crafted individual loss functions have been proposed for tackling data imbalance or label noise. A popular loss function is known as the focal loss (17), which is a modification of the standard cross-entropy loss. Specifically, it is defined as $-(1-p_t)^{\gamma}\log(p_t)$ where $\gamma > 0$ is a tuning parameter, p_t is the estimated probability for the ground-truth class. The focal loss has been observed to be effective for dense object detection and is also widely used for classification with imbalanced data due to its simplicity (48). However, the focal loss lacks theoretical foundation. To complement this, (49) proposed a theoretically-principled label-distribution-aware margin loss, which injects uneven margins into the cross-entropy loss, where the margin for each class is proportional to inverse of each class size to the power of 2/5. For tackling label noise, symmetric losses have been proposed, e.g., symmetric cross entropy loss (SCE) (50) and generalized cross entropy loss (TCE) (51). Our method is loss independent and hence can be combined with these existing crafted individual loss functions.

Optimization of DRO. DRO is a useful technique for domain adaptation, which has been shown both theoretically and empirically promising for learning with imbalanced data (52; 53; 54). However, most existing optimization algorithms for DRO are not practical for deep learning, which dims the usefulness of DRO. In the literature, DRO is formulated as (55; 53) :

$$\min_{\mathbf{w}\in\mathbb{R}^d} \max_{\mathbf{p}\in\Delta_n} \sum_{i=1}^{n} p_i L(\mathbf{w}; \mathbf{z}_i) - h(\mathbf{p}, \mathbf{1}/n) + r(\mathbf{w}), \tag{1}$$

where $\Delta_n = \{\mathbf{p} \in \mathbb{R}^n : \sum_i p_i = 1, p_i \geq 0\}$ denotes an n-dimensional simplex, $h(\mathbf{p}, \mathbf{1}/n)$ is a divergence measure or constraint between $\mathbf{p}$ and uniform probabilities $\mathbf{1}/n$, $r(\mathbf{w})$ is a standard regularizer on $\mathbf{w}$. We can see DRO aims to minimize the worst-case loss over all the underlying distribution $\mathbf{p}$ in an uncertainty set specified by $h(\mathbf{p}, \mathbf{1}/n)$. Many primal-dual optimization algorithms have been designed for solving the above problem for DL (56; 57). However, the dual variable $\mathbf{p}$ in the above min-max form is an n-dimensional variable restricted to a simplex, which makes existing primal-dual optimization algorithms computationally expensive and not applicable for the online setting where the data is coming sequentially. Our method can be considered as a solution to addressing these issues by considering a specific information-oriented

regularizer $h(\mathbf{p}, \mathbf{1}/n) = \lambda \sum_i p_i \log(np_i)$ that is the KL divergence between $\mathbf{p}$ and uniform probabilities $\mathbf{1}/n$, which allows us to transform the min-max formulation into an equivalent minimization formulation with a compositional objective. From this perspective, our method resembles a recently proposed dual-free algorithm RECOVER (58). However, RECOVER requires computing stochastic gradients at two different points in each iteration, which causes their GPU cost to double ours. In addition, RECOVER is a variance-reduction method, which might have poor generalization performance. Li et al. (59) investigated the effectiveness of the KL regularization DRO objective in different applications, including boosting the fairness between subgroups and dealing with class imbalance. Several recent studies also proposed stochastic algorithms for DRO (60; 61; 62; 63; 62), which are arguably more complicated than our methods.

3 Attentional-biased SGD with Momentum (ABSGD)

In this section, we present the proposed method ABSGD and its analysis. We first describe the algorithm and then connect it to the DRO framework. Then we present the convergence result of our method for solving IR-DRO. Throughout this paper, we let $\mathbf{z} = (\mathbf{x}, y)$ denote a random sample that includes an input $\mathbf{x} \in \mathbb{R}^d$ and the class label $y \in \{1, \ldots, K\}$, $\mathbf{w} \in \mathbb{R}^d$ denote the weight of the underlying DNN to be learned. Let $\mathbf{f}(\mathbf{x}) \in \mathbb{R}^K$ be the prediction score of the DNN on data $\mathbf{x}$, and $\ell(\mathbf{f}; y)$ denote a loss function. For simplicity, we let $L(\mathbf{w}; \mathbf{z}) = \ell(\mathbf{f}(\mathbf{x}); y)$. A standard loss function is the cross-entropy loss where $\ell(\mathbf{f}; y) = -\log \frac{\exp(f_y(\mathbf{x}))}{\sum_{k=1}^K \exp(f_k(\mathbf{x}))}$. We emphasize that our method is loss independent, and can be applied with any loss functions $\ell(\mathbf{f}; y)$. Specifically, ABSGD can employ the class-level weighted loss functions such as the class-balanced loss (45), crafted individual loss functions such as label-distribution aware margin loss (49).

3.1 Algorithm

The proposed algorithm ABSGD is presented in Algorithm 1. The key steps are described in Step 2 to Step 6, and the key updating step for $\mathbf{w}_{t+1}$ is given by

$$\textbf{ABSGD:} \quad \mathbf{w}_{t+1} = \mathbf{w}_t - \eta\left(\frac{1}{B}\sum_{i=1}^{B} \widetilde{p}_i \nabla L(\mathbf{w}_t; \mathbf{z}_i) + \nabla r(\mathbf{w}_t)\right) + \beta(\mathbf{w}_t - \mathbf{w}_{t-1}) \tag{2}$$

where $r(\mathbf{w}) \propto 1/2\|\mathbf{w}\|_2^2$ denotes a standard ℓ_2 norm regularization (i.e., for contributing weight decay in the update). The above update is a simple modification of the standard momentum method (64), where the last term $\beta(\mathbf{w}_t - \mathbf{w}_{t-1})$ is a momentum term. The modification lies at the introduced weight $\widetilde{p}_i$ for each data $\mathbf{z}_i$ in the mini-batch. The individual weight $\widetilde{p}_i$ is computed in Step 7 and is proportional to $\exp(L(\mathbf{w}_t; \mathbf{z}_i)/\lambda)$, where λ is a scaling parameter that $\lambda \in \{\mathbb{R} \backslash 0\}$. Intuitively, we can see that a sample with a large loss value tends to get a higher weight with $\lambda > 0$. It makes sense for learning with imbalanced data since the model tends to fit the data from the majority class while making the loss value larger for the minority class. Hence, the data from the minority class tends to get a larger weight $\widetilde{p}_i$. This phenomenon is demonstrated on a toy dataset in Figure 1. Similarly, if $\lambda < 0$, large value losses have smaller weights. As the noisy samples incurs larger losses than the clean samples, $\widetilde{p}_i$ would further emphasize more on the clean samples with larger weights, hence $\lambda < 0$ is preferred in the presence of label noise.

Algorithm 1 ABSGD $(\lambda, \eta, \gamma, \beta, s_0, \mathbf{w}_0, T)$

1: **for** $t = 0, \cdots, T-1$ **do**
2: Sample/Receive a mini-batch of B samples $\{\mathbf{z}_1, \cdots, \mathbf{z}_B\}$
3: Compute $\tilde{g}(\mathbf{w}_t) = \frac{1}{B}\sum_{i=1}^{B} \exp(L(\mathbf{w}_t, \mathbf{z}_i)/\lambda)$
4: Compute $s_{t+1} = (1-\gamma)s_t + \gamma \tilde{g}(\mathbf{w}_t)$
5: Compute $\widetilde{p}_i = \frac{\exp(\frac{L(\mathbf{w}_t; \mathbf{z}_i)}{\lambda})}{s_{t+1}}$, for $i = 1, \ldots, B$
6: Update $\mathbf{w}_{t+1}$ by Equation (2)
7: **end for**
8: **Return** $\mathbf{w}_T$

It is notable that the weight $\widetilde{p}_i$ is properly normalized dividing a quantity s_{t+1} that is updated online. In particular, s_{t+1} maintains a moving average of exponential of the scaled loss value on the sampled data

(Step 4). It is notable that the normalization does not make the sum of $\widetilde{p}_i$ in the mini-batch equal to 1. We emphasize that this normalization is essential in twofold: (i) it stabilizes the update without causing a significant numerical issue; (ii) it ensures the algorithm converge to a meaningful solution as presented in next subsection.

3.2 Connection with Min-max or Min-min Robust Optimization

In the next subsection, we will show that ABSGD converges to a stationary solution of two robust optimization problems depending on whether λ is positive or negative. In particular, given n training samples $\{\mathbf{z}_1, \ldots, \mathbf{z}_n\}$ we consider the following min-max and min-min robust optimization:

$$\min_{\mathbf{w}\in\mathcal{R}^d} \underbrace{\max_{\mathbf{p}\in\Delta_n} \sum_{i=1}^n p_i L(\mathbf{w};\mathbf{z}_i) - \tau \sum_i^n p_i \ln(np_i)}_{F_\tau^{(1)}(\mathbf{w})} + r(\mathbf{w}). \tag{3}$$

$$\min_{\mathbf{w}\in\mathcal{R}^d} \underbrace{\min_{\mathbf{p}\in\Delta_n} \sum_{i=1}^n p_i L(\mathbf{w};\mathbf{z}_i) + \tau \sum_i^n p_i \ln(np_i)}_{F_\tau^{(2)}(\mathbf{w})} + r(\mathbf{w}) \tag{4}$$

where $\tau > 0$ and Δ_n is a simplex. In the Appendix, we show that $F_\tau^{(1)}(\mathbf{w}) = \tau \log \frac{1}{n}\sum_i \exp(L(\mathbf{w};\mathbf{z}_i)/\tau) + r(\mathbf{w})$ and $F_\tau^{(2)}(\mathbf{w}) = -\tau \log \frac{1}{n}\sum_i \exp(-L(\mathbf{w};\mathbf{z}_i)/\tau) + r(\mathbf{w})$. Similar min-max and min-min formulations have been considered in the literature under the framework of tilting log-likelihood (65). Recently, there is some renaissance of solving the min-max and min-min formulation in machine learning. For example, the min-max formulation (3) is also closely related to distributionally robust optimization (53) with a difference that a regularization is imposed on $\mathbf{p}$ instead of a constraint function. The min-min formulation has been considered in (66) for tackling noisy data. Recently, the titled risk functions $F_\tau^{(1)}(\mathbf{w})$ and $F_\tau^{(2)}(\mathbf{w})$ have been also studied in (59). We describe the difference between ABSGD and the algorithm in (59) in detail in section 3.5.

By considering the explicit $\tau \sum_i p_i \log(np_i)$ regularizer in the two DRO formulations, our algorithm is applicable to solving the min-max objective (3) by setting $\lambda = \tau$ and the min-min objective (4) by setting $\lambda = -\tau$. When $\tau = +\infty$, $p_i = 1/n$ according to the close form solution derived in Eqn (5). Then above DRO objectives, Eqn (3) and (4), become the standard empirical risk minimization problem: $\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^n L(\mathbf{w};\mathbf{z}_i) + r(\mathbf{w})$. When $\tau = 0$, then $\mathbf{p}$ has only one component equal to 1 that corresponds to the data with largest loss value for Eqn (3) and the data with smallest loss value for Eqn (4). Hence, when $\tau \to 0$, DRO objective (3) becomes the maximal loss minimization: $\min_{\mathbf{w}\in\mathbb{R}^d} \max_i L(\mathbf{w};\mathbf{z}_i) + r(\mathbf{w})$. And when $\tau \to 0$, DRO objective (4) becomes the minimal loss minimization: $\min_{\mathbf{w}\in\mathbb{R}^d} \min_i L(\mathbf{w};\mathbf{z}_i) + r(\mathbf{w})$. The above maximal loss minimization has been studied for learning with imbalanced data (52). It was shown theoretically to yield better generalization performance than empirical risk minimization for imbalanced data. However, the maximal loss minimization is sensitive to outliers. Hence, by varying the value of τ we can enjoy the balanced robustness between the imbalanced data and outliers.

3.3 Optimization Analysis

It is nice that the DRO formulation is robust to imbalanced data (Eqn (3)) and noisy data (Eqn (4)). However, the min-max/min formulation of DRO is not friendly to the design of efficient optimization algorithms, especially given the constraint $\mathbf{p} \in \Delta_n$. To this end, we transform the min-max/min formulation (3) and (4) into an equivalent minimization formulation following (58). In particular, we first compute the optimal solution of dual variable $\mathbf{p}^*$ for the inner maximization/minimization problem given $\mathbf{w}$. By taking the first-derivation of equation (3) and (4) in terms of $\mathbf{p}$ and setting it to zero, *i.e.* $\nabla_{\mathbf{p}} F(\mathbf{w},\mathbf{p}) = 0$, we have $\mathbf{p}^*$:

$$p_i^* = \frac{\exp(\frac{L(\mathbf{w};\mathbf{z}_i)}{\lambda})}{\sum_{i=1}^n \exp(\frac{L(\mathbf{w};\mathbf{z}_i)}{\lambda})}, \quad i = 1, \ldots, n \tag{5}$$

where $\lambda = \tau$ for equation (3) and $\lambda = -\tau$ for equation (4). By substituting $\mathbf{p}^*$ back, we obtain the following equivalent minimization formulation:

$$\min_{\mathbf{w}\in\mathbb{R}^d} F_\lambda(\mathbf{w}) = \lambda \ln\left(\frac{1}{n}\sum_{i=1}^{n}\exp\left(\frac{L(\mathbf{w};\mathbf{z}_i)}{\lambda}\right)\right) + r(\mathbf{w}). \tag{6}$$

In an online learning setting, we can further generalize the above formulation as

$$\min_{\mathbf{w}\in\mathbb{R}^d} F_\lambda(\mathbf{w}) = \lambda \ln\left(\mathbb{E}_{\mathbf{z}}\exp(L(\mathbf{w};\mathbf{z})/\lambda)\right) + r(\mathbf{w}). \tag{7}$$

Given the above minimization formulations, our method ABSGD can be considered as a stochastic algorithm for optimizing (6) in offline learning or optimizing (7) in online learning. Our method is rooted in stochastic optimization for compositional optimization that has been studied in the literature (67; 68; 69; 58). Intuitively, we can understand our weighting scheme $\widetilde{\mathbf{p}}$ as following. In offline learning with a big data size where n is huge, it is impossible to calculate the $\mathbf{p}^*$ as in (5) at every iteration due to computation and memory limits. As a result, we need to approximate $\mathbf{p}^*$ in a systematic way.

In our method, we use moving average estimate s_{t+1} to approximate the denominator in $\mathbf{p}^*$, i.e., $\frac{1}{n}\sum_{i=1}^{n}\exp(\frac{L(\mathbf{w};\mathbf{z}_i)}{\lambda})$, and use it to compute a scaled weight of data in the mini-batch by Step 5, i.e.,

$$\widetilde{p}_i = \frac{\exp(\frac{L(\mathbf{w}_t;\mathbf{z}_i)}{\lambda})}{s_{t+1}}, \quad i \in \{1,\cdots B\}. \tag{8}$$

More rigorously, our method ABSGD is a stochastic momentum method for solving a compositional problem in the form $f(g(\mathbf{w}))+r(\mathbf{w})$. To this end, we write the objective in (7) as $f(g(\mathbf{w}))+r(\mathbf{w})$, where $f(g) = \lambda\log(g)$ and $g(\mathbf{w}) = \mathbb{E}_{\mathbf{z}}[\exp(L(\mathbf{w};\mathbf{z})/\lambda)]$. The difficulty of stochastic optimization for the compositional function $f(g(\mathbf{w}))$ lies on computing an approximate gradient at $\mathbf{w}_t$. By the chain rule, its gradient is given by $\nabla f(g(\mathbf{w}_t))\nabla g(\mathbf{w}_t) = \frac{\lambda}{g(\mathbf{w}_t)}\nabla g(\mathbf{w}_t)$. To approximate $\nabla f(g(\mathbf{w}_t)) = \frac{\lambda}{g(\mathbf{w}_t)}$, we use a moving average to estimate $g(\mathbf{w}_t)$ inspired by (67), which is updated in Step 4 of Algorithm 1, i.e.,

$$s_{t+1} = (1-\gamma)s_t + \gamma\tilde{g}(\mathbf{w}_t), \quad \text{where } \tilde{g}(\mathbf{w}_t) = \frac{1}{B}\sum_{i=1}^{B}\exp(\frac{L(\mathbf{w}_t,\mathbf{z}_i)}{\lambda}), \{\mathbf{z}_i\} \text{ are random samples.}$$

And $\nabla g(\mathbf{w}_t)$ can be estimated by mini-batch stochastic gradient, i.e.,

$$\nabla\tilde{g}(\mathbf{w}_t) = \frac{1}{B}\sum_{i=1}^{B}\exp(L(\mathbf{w}_t;\mathbf{z}_i)/\lambda)\frac{\nabla L(\mathbf{w}_t;\mathbf{z}_i)}{\lambda}.$$

Hence, the true gradient $\nabla f(g(\mathbf{w}_t))\nabla g(\mathbf{w}_t)$ is able to be approximated by

$$\frac{\lambda}{s_{t+1}}\nabla\tilde{g}(\mathbf{w}_t) = \frac{1}{B}\sum_{i=1}^{B}\frac{1}{s_{t+1}}\exp(L(\mathbf{w}_t;\mathbf{z}_t)/\lambda)\nabla L(\mathbf{w}_t;\mathbf{z}_i),$$

which is exactly the approximate gradient used in the update of $\mathbf{w}_{t+1}$ as in equation (2). However, instead of directly using this gradient estimator to update the model parameter following (67), we employ a momentum update. The reason is that the algorithm in (67) has a larger sample complexity, which is $O(1/\epsilon^5)$ for finding an ϵ-stationary point of the objective function (cf. Section 3.5). By using a momentum update as in (2), we are able to establish an optimal sample complexity. It is notable that the momentum update can be seen as a simplification of the NASA method proposed in (68), which was designed to address the constrained compositional optimization.

3.4 Other AB methods

In light of the discussion about the connection between ABSGD and optimization of IR-DRO, we can generalize ABSGD to employ other updating schemes, e.g., AdaGrad (70), RMSProp (71; 72), Adam (73). Below, we present the ABAdam method. The key idea is to replace the standard mini-batch gradient estimator in Adam by the weighted mini-batch gradient estimator. The key steps of ABAdam are presented below.

$$
\begin{aligned}
G(\mathbf{w}_t) &= \frac{1}{B}\sum_{i=1}^{B} \widetilde{p}_i \nabla L(\mathbf{w}_t; \mathbf{z}_i) \\
\textbf{ABAdam:} \quad \mathbf{v}_{t+1} &= \beta_1 \mathbf{v}_t + (1-\beta_1) G(\mathbf{w}_t) \\
\mathbf{u}_{t+1} &= \beta_2 \mathbf{u}_t + (1-\beta_2)(G(\mathbf{w}_t))^2 \\
\mathbf{w}_{t+1} &= \mathbf{w}_t - \eta\left(\frac{\mathbf{v}_{t+1}}{\sqrt{\mathbf{u}_{t+1}} + G_0} + \nabla r(\mathbf{w}_t)\right)
\end{aligned} \tag{9}
$$

where G_0 is a constant to increase stability, β_1, β_2 are the constant hyperparameters that are usually set as 0.9 and 0.999, respectively. ABAdam could potentially benefit the applications that Adam has better generalization performance than the SGD (74; 75). In the appendix, we provide a theoretical analysis for ABAdam for optimizing IR-DRO, and leave the experimental exploration for the future.

3.5 Convergence Analysis

In this subsection, we provide a convergence result of ABSGD for solving the min-max or the min-min objective under some standard assumptions in non-convex optimization. For presentation simplicity, we use the notations $g(\mathbf{w}) = \mathbb{E}_{\mathbf{z}}[\exp(L(\mathbf{w};\mathbf{z})/\lambda)]$ and $g(\mathbf{w};\mathbf{z}) = \exp(L(\mathbf{w};\mathbf{z})/\lambda)$. We first state a standard assumption (58; 67) and then present our main theorem.

Assumption 1. *Let V_g, L_l are constant scalars,*

- *For a fixed λ, there exists $V_g > 0$ such that $\mathbb{E}_{\mathbf{z}}[\|g(\mathbf{w};\mathbf{z}) - g(\mathbf{w})\|^2] \leq V_g, \mathbb{E}_{\mathbf{z}}[\|\nabla g(\mathbf{w};\mathbf{z}) - \nabla g(\mathbf{w})\|^2] \leq V_g$ and $L(\mathbf{w};\mathbf{z})$ for any $\mathbf{z}$ is an L_l-smooth function, i.e., $\|\nabla L(\mathbf{w};\mathbf{z}) - \nabla L(\mathbf{w}';\mathbf{z})\| \leq L_l\|\mathbf{w} - \mathbf{w}'\|, \forall \mathbf{w}, \mathbf{w}'$*
- *For a given τ, there exists Δ_0 such that $F_\tau^{(1)}(\mathbf{w}_1) - \min F_\tau^{(1)}(\mathbf{w}) \leq \Delta_0$ or $F_\tau^{(2)}(\mathbf{w}_1) - \min F_\tau^{(2)}(\mathbf{w}) \leq \Delta_F$.*

Theorem 1. *Assume assumption 1 holds and there exists C_0, C_1 such that $\exp(L(\mathbf{w}_t;\mathbf{z}_i)/\lambda) \leq C_0, \|\nabla L(\mathbf{w}_t;\mathbf{z}_i)\| \leq C_1$, for all $\mathbf{w}_t$ and any $\mathbf{z}_i$, Then, For $\lambda = \tau > 0$ with appropriate η, γ, β, ABSGD ensures that $E\left[\frac{1}{T}\sum_{t=1}^{T}\|\nabla F_\tau^{(1)}(\mathbf{w}_t)\|^2\right] \leq \epsilon^2$ after $T = O(1/\epsilon^4)$ iterations, and for $\lambda = -\tau < 0$ with appropriate η, γ, β, ABSGD ensures that $\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}\|\nabla F_\tau^{(2)}(\mathbf{w}_t)\|^2\right] \leq \epsilon^2$ after $T = O(1/\epsilon^4)$ iterations, where we exhibit the constant in the big O in Appendix.*

Remark: Before ending this section, we present some remarks. First, we notice that in a concurrent work (59), the authors proposed a similar algorithm to ABSGD without the momentum term, i.e., $\gamma = 1$. However, they only prove the convergence for the algorithm with independent mini-batches for $L(\mathbf{w};\mathbf{z})$ and $\nabla L(\mathbf{w};\mathbf{z}')$. In our experiments, we show that the momentum term is important for speeding up the convergence. In another concurrent work (66) the authors proposed an algorithm for solving the min-min objective (4). The difference is that in their algorithm the normalization for computing the weight is computed only from the current mini-batch while that in ABSGD depends on all historical data. In addition, (66) provides no convergence analysis for solving the min-min robust optimization problem.

3.6 Two-stage Training Strategy for λ

Since λ can be interpreted as the regularization parameter in IR-DRO, we can understand its impact on the learning of model. With a larger $|\lambda|$, the IR-DRO is getting closer to ERM and ABSGD is getting close to the standard momentum SGD method without robust weighting. When $|\lambda| = \infty$, the update becomes exactly the same as the standard momentum SGD method. When $|\lambda|$ becomes smaller, the update will focus more on data with larger loss values (e.g., from the minority class). This uneven weighting is helpful to learn a robust classifier. However, it might harm the learning of feature extraction layers. This phenomenon has been also observed in previous works (49; 76).

To address this issue, we employ a two-stage training method following the existing literature (76), where in the first stage we employ momentum SGD to learn a basis network, and in the second stage we employ ABSGD to learn the classifier and finetune the feature layers. As momentum SGD is a special case of ABSGD with $|\lambda| = \infty$, the two-stage method is equivalent to running ABSGD with $|\lambda| = \infty$ first and then restarting

it with a decayed $|\lambda| < \infty$. In the ablation study, we will show that damping $|\lambda|$ is critical for balancing the learning of feature extraction layers and classifier layers. Finally, it is notable that in the second stage, we can fix some lower layers and only fine-tune upper layers using ABSGD.

4 Experimental Results on Data Imbalance Problem

We conduct experiments on multiple imbalanced benchmark datasets, including CIFAR-10 (LT), CIFAR-10 (ST), CIFAR-100 (LT), CIFAR-100 (ST), ImagetNet-LT (5), Places-LT (77), and iNaturelist2018 (78), and compare ABSGD with several state-of-the-art (SOTA) methods, including meta-learning (46), class-balanced weighting (49), and two-stage decoupling methods (76). We use the ResNets (79) as the main backbone in our experiments. For fair comparison, ABSGD is implemented with the same hyperparameters such as momentum parameter, initial step size, weight decay and step size decaying strategy, as the baseline momentum SGD method. For ABSGD, the moving average parameter γ are tuned in $[0.1 : 0.1 : 1]$ by default. Without additional mentions, we directly use the results of baselines from their original papers by default.

Datasets: The original CIFAR-10 and CIFAR-100 data contain 50,000 training images and 10,000 validation with 10 and 100 classes, respectively. We construct the imbalanced version of training set of CIFAR10, CIFAR100 following the two strategies: Long-Tailed (LT) imbalance (49) and Step (ST) imbalance (80) with two different imbalance ratio $\rho = 10, \rho = 100$, and keep the testing set unchanged. The imbalance ratio ρ is defined as the ratio between sample sizes of the most frequent and least frequent classes. The LT imbalance follows the exponentially decayed sample size between different categories. In ST imbalance, the number of examples are both equal within minority classes and majority classes but differs between the majority and minority classes. We denote the imbalanced versions of CIFAR10, CIFAR100 as CIFAR10-LT/ST, CIFAR100-LT/ST according the imbalanced strategies. ImageNet-LT (5) is a long-tailed subset of the original ImageNet-2012 by sampling a subset following the Pareto distribution with the power value 6. It has 115.8K images from 1000 categories, which include 4980 for the head class and 5 images for the tail class. The Places-LT dataset was also created by sampling from Places-2 (77) using the same strategy as ImageNet-LT. It contains 62.5K training images from 365 classes with an imbalance ratio $\rho = 4980/5$. iNaturalist 2018 is a real world dataset whose class-frequency follows a heavy-tail distribution (78). It contains 437K images from 8142 classes. The long-tail and step imbalance label distribution of the datasets are shown in Figure 2.

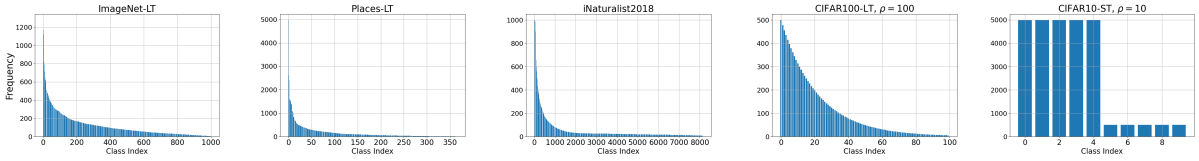

Figure 2: Long-tail label distributions of ImageNet-LT, Places-LT, iNaturalist2018 and CIFAR100 with imbalance ratio $\rho = 100$, and Step imbalance label distribution of CIFAR10 with imbalance ratio $\rho = 10$.

Label-Distribution Independent Losses. We first compare the effectiveness of our ABSGD method with standard momentum SGD method for DL. In particular, we consider two loss functions, cross-entropy (CE) loss and focal loss. The baseline method is the momentum SGD optimizing these losses, denoted by SGD (CE) and SGD (Focal). Our methods are denoted by ABSGD (CE) and ABSGD (Focal) that employ the two losses in our framework. This comparison is meaningful as in the online learning setting the prior knowledge of class-frequency is not known.

Label-distribution Dependent Losses. Next, we compare ABSGD with baseline methods that use label-distribution dependent losses. In particular, we consider class-balanced (CB) versions of three individual losses, including CE loss, focal loss, label-distribution-aware margin (LDAM) loss (49). The class-balanced weighing strategy is from (45), which uses the effective number of samples to define the weight. As a result, there are three categories of CB losses, i.e., CB-CE, CB-Focal, CB-LDAM. We use our method ABSGD with these different losses. In particular, ABSGD + CB-CE/Focal/LDAM uses a combination of class-level weighting and instance-level weighting, which is expected to have outstanding performance as it considers

diversity between examples at both class level and individual level. For each of these losses, we consider two baseline optimization methods. The first method is the standard momentum SGD method with a practical useful trick (49) that defers adding the class-level weighting after a number of pre-training steps with no class-level weights to improve the performance. We denote the first method by SGD (XX), where XX denotes the loss function. The second method is the meta learning method (46) that uses meta-learning on a separate validation data to learn individual weights and combines them with class-balanced weights. The meta learning method has been observed with SOTA results on these benchmark imbalanced datasets. We let META (XX) denote the second method. Our method is denoted by ABSGD (XX).

In the following, we compare ABSGD, SGD, and meta-learning methods by optimizing the same label-dependent and label-independent losses on imbalanced CIFAR datasets, and including more baselines on ImageNet-LT, Places-LT, and iNaturalist-LT.

Table 1: Top-1 testing accuracy (%), mean (std), of ResNet32 on imbalanced CIFAR-10 and CIFAR-100 trained with label-distribution independent losses. The results are reported over 3 independent runs.

Dataset	Imbalance Type	long-tailed (LT)		step (ST)	
	Imbalance Ratio	100	10	100	10
Cifar10	SGD (CE)	71.75 (± 0.75)	87.64 (± 0.45)	63.12 (± 0.63)	85.63 (± 0.41)
	ABSGD (CE)	**72.43** (± 0.31)	**87.93** (± 0.25)	**66.24** (± 0.35)	**85.84** (± 0.27)
	SGD (Focal)	70.86 (± 0.68)	87.10 (± 0.41)	63.31 (± 0.61)	85.55 (± 0.46)
	ABSGD (Focal)	**72.48** (± 0.28)	**87.26** (± 0.35)	**65.03** (± 0.33)	**85.67** (± 0.30)
Cifar100	SGD (CE)	38.35 (± 0.63)	56.91 (± 0.44)	39.23 (± 0.58)	55.09 (± 0.35)
	ABSGD (CE)	**39.77** (± 0.34)	**57.44** (± 0.25)	**39.76** (± 0.37)	**55.15** (± 0.29)
	SGD (Focal)	39.05 (± 0.71)	56.89 (± 0.41)	39.32 (± 0.61)	54.45 (± 0.43)
	ABSGD (Focal)	**39.37** (± 0.38)	**57.08** (± 0.29)	**39.75** (± 0.39)	**55.40** (± 0.33)

4.1 Experimental Results on CIFAR Datasets

Setups Following the experimental setting in the literature, the initial learning rate is 0.1 and decays by a factor of 100 at the 160-th, 180-th epoch for both ABSGD and SGD in our experiments, respectively. The value of λ in ABSGD tuned in $[1:1:10]$.

Results. We report the results with label independent losses in Table 1 and with label dependent losses in Table 2. We can see that ABSGD consistently outperforms SGD with a noticeable margin regardless of imbalance strategies and imbalance ratio ρ. In particular, we have more than 2% improvements on the CIFAR10-ST and CIFAR100-LT, respectively with $\rho = 100$. For the label dependent losses, we have the following observations, comparing ABSGD with SGD, we can see that our method that incorporates the self-adaptive robust weighting scheme performs consistently better in all imbalanced settings. This verifies that the proposed self-adaptive weighting scheme is also effective even when applied on top of the class-level weighting strategy. It is notable that META requires a separate validation data and is more computationally expensive than our method. Hence, our method is a strong choice even compared with the SOTA meta learning method, especially for highly imbalanced tasks. Also, the improvements of ABSGD with CB losses over ABSGD with label independent losses verify the importance of prior label information in addressing the data imbalance problem.

4.2 Experimental Results on ImageNet-LT, Places-LT and iNaturalist2018

Setups and baselines. Next, we conduct experiments on large-scale datasets and compare ABSGD with more baselines. We conduct experiment on two different architectures, ResNet50 for ImageNet-LT and iNaturalist2018, and ResNet152 for Places-LT and iNaturalist2018. We compare ABSGD with several methods, which include single-stage methods such as momentum SGD for optimizing LDAM loss, CB-CE loss and CB-Focal loss, two-stage methods such as τ-normalized (CB-CE), LWS (CB-CE) proposed in (76), and meta-learning method (META) (46). For the two-stage decoupling strategy baseline methods (76), the first stage uses the standard uniform sampling to train the model with the CE loss, and the second stage fine tunes part of parameters in higher layers such as the fully connected (FC) layers and last block of (LB) feature layers. META also uses the two-stage strategy to improve the performance.

Table 2: Top-1 testing accuracy (%) of ResNet32 on imbalanced CIFAR-10 and CIFAR-100 trained with label-distribution dependent losses. The red numbers indicate the best in each category of class-weighted loss. The **bold red** numbers indicate the best in each imbalanced setting. The original paper of META does not include the results on the ST imbalanced setting, hence their missing results are marked by −.

Datasets	Imbalanced CIFAR-10				Imbalanced CIFAR-100			
Imbalance Type	long-tailed		step		long-tailed		step	
Imbalance Ratio	100	10	100	10	100	10	100	10
Resampling (CE)	71.78	86.99	61.16	84.59	38.87	56.92	38.84	54.35
SGD (CB-CE) (45)	72.37	86.77	61.84	83.80	38.70	57.56	21.31	53.39
META (CB-CE) (46)	76.41	**88.85**	-	-	43.35	59.58	-	-
ABSGD (CB-CE)	79.34	88.57	72.93	**88.42**	**45.54**	**61.12**	**45.89**	**60.77**
SGD (CB-Focal) (45)	74.57	87.10	60.27	83.46	36.02	57.99	19.76	50.02
META (CB-Focal) (46)	78.90	88.37	-	-	44.70	59.59	-	-
ABSGD (CB-Focal)	79.53	88.76	76.33	85.90	44.11	59.14	45.41	59.75
SGD (LDAM) (49)	73.35	86.69	66.58	85.00	39.60	56.91	39.58	56.27
SGD (CB-LDAM) (49)	77.03	88.12	76.92	87.81	42.04	58.71	45.36	59.46
META (CB-LDAM) (46)	80.00	87.40	-	-	44.08	58.80	-	-
ABSGD (CB-LDAM)	**80.45**	88.27	**78.33**	88.40	44.71	59.21	45.65	58.74

Here, to achieve the SOTA results, we investigate two-stage decoupling strategy for ABSGD. Hence, the two-stage decay λ training scheme can be automatically applied. For ImageNet-LT, we jointly train the feature representation and classifier using momentum SGD in the first stage for 90 epochs from scratch, and finetune the FC layer for 90 epochs of ABSGD in the second stage. For Places-LT, we train the Last Block (LB) of the convolutions layer and Fully Connected (FC) layer for 90 epochs in the first stage using SGD with momentum from an ImageNet pretrained model, and finetune the FC and LB layer for 30 epochs in second stage using ABSGD. For iNaturalist2018, we run momentum SGD ($\beta = 0.9$) for 200 epochs in the first stage from the ImageNet pretrained model, and in the second stage we only finetune FC layer and LB of the neural network using ABSGD with $\lambda = 10$ for 30 epochs. λ is tuned in $\{10, 20, 30\}$ for all datasets. The initial learning rates and learning scheme are described in Table 8. All of our results are reported based on 3 independent runs.

Table 3: Test top-1 accuracy(%) of different baseline methods on ImageNet-LT with Resnet50.

Methods	Sampling	Loss	Stage-1 TV	Stage-2 TV	Results
Vanilla Model (46)	None	CE	All	-	41.0
CB-CE (45)	None	CE	All	-	41.8
Joint (76)	CB	CE	All	All	41.6
NCM (76)	CB	CE	All	FC	44.3
cRT (76)	CB	CE	All	FC	43.3
τ-normalizer (76)	CB	CE	All	FC	46.7
META† (46)	None	CE	All	FC	<u>48.0</u>
ABSGD	None	CE	All	FC+LB	**48.2**

Table 4: Test top-1 accuracy(%) of different baseline methods on Places-LT using ResNet50.

Methods	Sampling	Loss	Stage-1 TV	Stage-2 TV	Results
Vanilla Model (46)	-	CE	FC/LB+FC	-	27.9/30.3
Vanilla Model (81)	-	Range	FC		35.1
Joint (76)	CB	CE	LB+FC	LB+FC	30.2
NCM (76)	CB	CE	LB+FC	FC	36.3
cRT (76)	CB	CE	LB+FC	FC	36.7
τ-normalized (76)	None	CE	LB+FC	FC	<u>37.9</u>
OLTR* (5)	CB	CE	LB+FC	FC	35.9
META † (46)	None	CE	LB+FC	FC	37.1
ABSGD	None	CE	LB+FC	FC	**38.7**

Results Table 3, 4 5 are the experimental results of ImageNet-LT, Places-LT and iNaturalist2018, respectively. To better understanding results, we make some notes in the table. *TV* represents Trainable Variable. *All* represents standard training process that optimizes all the parameters of the backbone. *FC* represents fully connected layer, LB represents the last block of feature layers in the backbone. The CB in the Sampling column denotes Class-Balanced Sampling (45) in the second stage. † represents an additional balanced data set is required in the second stage. * denotes an additional memory is required in the second stage. The bold numbers and the numbers with underline in the table represent the best and the second best the numbers with underline on each dataset, respectively.

We can see that ABSGD combining with the two stage training strategy achieves best on all three datasets. For the ImageNet-LT dataset in Table 3, ABSGD has 0.2% improvements over the next best META method while has no requirements on the additional balanced validation datasets. For Places-LT, ABSGD has 0.9% improvements over than the best baseline, τ-normalized. For the iNaturalist2018-LT in Table 5, ABSGD outperforms all other baselines by a large margin 0.3% and 0.6% for using both ResNet50 and ResNet152, respectively. To the best of our knowledge, 73.1% is the SOTA result on iNaturalist2018 dataset. In addition,

it is worth to mention that all the other baselines takes the advantage of the Class-Balanced Sampling or additional balanced validation datasets (META), which makes ABSGD more favorable than the baselines.

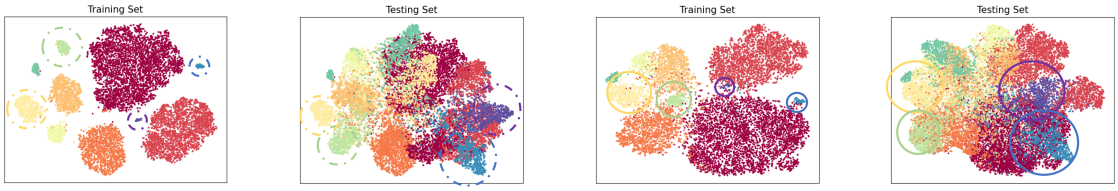

Figure 3: t-SNE visualization of feature representations of training & testing set on CIFAR10-LT ($\rho = 100$) with different λ strategies. Left two figures: Two-stage decay of λ: first phase $\lambda = 100$ and second phase $\lambda = 1$. Right two figures: Fixed $\lambda = 1$.

Table 5: Top-1 testing accuracy(%) of different methods on iNaturalist2018 using ResNet50, ResNet152.

Methods Network	Stage-2 TV	Results ResNet50	Results ResNet152
CE (45)	-	65.8	69.0
LDAM (49)	-	68.0	-
CB-Focal (45)	-	61.1	-
NCM (CE) (76)	FC	63.1	67.3
cRT (CB-CE) (76)	FC	68.2	71.2
τ-Normalized (CE) (76)	FC	69.3	<u>72.5</u>
LWS (CB-CE) (76)	FC	<u>69.5</u>	72.1
META† (CB-CE) (46)	All	67.6	-
META† (CB-Focal) (46)	All	67.7	-
ABSGD (CB-CE)	FC	**69.8**	**73.1**

4.3 Ablation Studies on CIFAR Datasets

In the ablation study, we first study ABSGD from different perspectives: a) the stagewise decay λ; b) the influence of the moving average parameter γ on the testing accuracy. Then we plot the average instance robust weights for each class to show the attention of ABSGD towards the minority class.

Two-stage decay of λ. To verify the model enjoys the benefits of stagewise decay λ the same as the learning rate η, we compare the feature representations in both training and testing data between adopting the two-stage λ decay training strategy and using a fixed value of λ during the training. For two-stage strategy, we use $\lambda = 100$ in the first phase and decay it to 1 in the second phase. For fixed values of λ, we use $\lambda = 1$. The results are plotted in the second column of Figure 3 on CIFAR10-LT. It is clear to see using the stagewise strategy on λ yields much better feature representations that are well separated between different classes. In contrast, the learned feature representations with a fixed value $\lambda = 1$ are more cluttered. Thus the stagewise decay λ strategy is better than using a fixed value of λ, which verifies our algorithmic choice. We also provide the convergence curves of different λ strategies in Appendix.

The sensitivity of the moving average parameter γ In the derivation of Theorem 1, $\gamma = O(\frac{1}{\sqrt{T}})$, which decreases to 0 when the total number of iterations increases. In practical training, we tune the $\gamma \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$. We report the testing accuracy over 3 independent runs in Figure 4 (left two) and compare it with standard SGD training, the green dashed line. Here we can see that ABSGD achieves highest testing accuracy with $\gamma = 0.5$ on both CIFAR10-LT and CIFAR100-LT. All the results of ABSGD with different γ are better or comparable than momentum SGD verifies the effectiveness of the moving average estimator Step 4 in Algorithm 1.

The average instance weights per-class ABSGD is an instance-level weighting method. For each sample, ABSGD assigns a robust weight that is proportional to the scaled loss value. For ABSGD (CE), we plot the average robust weights for the samples in the minority and majority class in Figure 4 (right two). It can be clearly seen that the average weights of samples in minority class is greater than the average weights of samples in majority class, which verifies the intuition behind ABSGD.

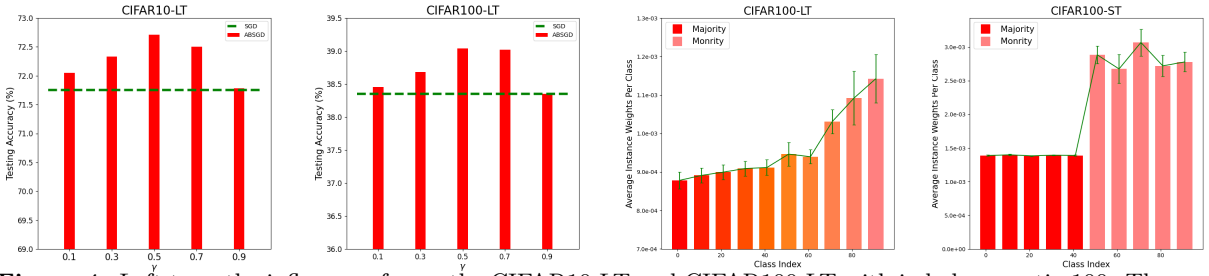

Figure 4: Left two: the influence of γ on the CIFAR10-LT and CIFAR100-LT with imbalance ratio 100. The results are reported over 3 independent runs. The green error bar is the stand deviation of each results. Right two: the average instance weights for difference classes during the training process on CIFAR100-LT and CIFAR100-ST with imbalanced ratio 100 on ResNet32.

5 Experimental Results on Label Noise Problem

To show the effectiveness of ABSGD for handling noisy labels, we provide empirical studies on the noisy label datasets in this section. We conduct experiments on CIFAR10, CIFAR100, and Clothing1M (30) datasets. The noise rate is defined as the portion of samples whose ground truth label are randomly flipped. We follow the same setting as (50) and consider both the symmetric label noise and asymmetric label noise on CIFAR10 and CIFAR100 with the noisy rates $\{0.2, 0.4\}$ (50; 82; 51) in our experiments. The symmetric (uniform) noisy labels are generated by flipping the labels of a given proportion of training samples to one of the other class labels uniformly. The asymmetric noisy labels are class-dependent noise, in which the flipping of labels only occur within a specific set of classes. Please refer to the Noise setting section in (50) for details. The Clothing1M is a real-world large-scale label noisy dataset and includes 14 classes with 1M training images in total.

Baselines We compare ABSGD with SGD and a mini-batch based method for solving the min-min DRO formulation (4) named EG (66) with three losses. The first is the standard CE loss. Then a theoretically grounded generalized cross entropy loss, named as TCE, has been proposed later on (51). Furthermore, (50) proposed a symmetric loss, named SCE, to address the under learning and overfitting problem that widely exists in the noisy labels. For crafting loss hyperparameters, we tune the symmetric parameters in SCE $\alpha, \beta \in \{0.1, 1, 0.5, 1, 5\}$ and the truncated parameter q in TCE is tuned in $\{0.1, 0.5, 0.7\}$. The momentum parameter γ for ABSGD is tuned in $\{0.1, 0.5, 0.9\}$.

5.1 Experimental Results on CIFAR Datasets

Experimental Setting. Following the setting in (50), we use a 4-layer CNN proposed in (50) for CIFAR10 data. For the CIFAR100, we use the ResNet18 for the symmetric noisy labels and ResNet34 for the asymmetric noisy labels. We report the results of using CE and TCE losses optimized by SGD, and SCE optimized by SGD, EG, and ABSGD, respectively. The weight decay for different methods are tuned in $\{1e\text{-}4, 5e\text{-}4, 1e\text{-}3, 5e\text{-}3\}$. We train the model for 120 epochs and the batch size is fixed as 128 for all settings. The initial learning rates are tuned in $\{1e\text{-}3, 1e\text{-}2, 1e\text{-}1\}$ and decayed at the epoch of 40, and 80 epochs by a factor of 10. The ABSGD hyper-parameter λ is tuned in $\{-0.1, -0.5, -1, -2, -3\}$.

Results. The results are reported in Table 6. Among the three baselines, SCE achieves better/comparable experimental results in most of the different models, settings and datasets. Then the testing accuracy is consistently improved further by optimizing SCE with the proposed ABSGD. By comparing the results across different noisy rates, we can see that our ABSGD(SCE) improves more when the noisy rate increases.

5.2 Experimental Results on Clothing1M

Experimental Setting. We train the ResNet50 starting from the ImageNet pretrained model for all the baselines following the same setting as (50) on the Clothing1M dataset. The training phase includes 10 epochs, and the initial learning rates is fixed as 0.002 and decayed by a factor of 10 at the 5th epoch for all

Table 6: Top-1 testing accuracy (%) on noisy labelled CIFAR10 and CIFAR100 data of different methods. Results are reported over 5 independent runs.

		Symmetric		Asymmetric	
	Noisy Rate	0.2	0.4	0.2	0.4
CIFAR10	SGD(CE)	88.59 (± 0.21)	85.75 (± 0.31)	86.62 (± 0.27)	80.81 (± 0.29)
	SGD(TCE)	89.87 (± 0.27)	86.84 (± 0.32)	88.97 (± 0.31)	80.85 (± 0.27)
	SGD(SCE)	90.05 (± 0.23)	87.83 (± 0.33)	90.25 (± 0.34)	81.91 (± 0.42)
	EG(SCE)	90.25 (± 0.21)	88.13 (± 0.29)	90.55 (± 0.32)	82.41 (± 0.23)
	ABSGD(SCE)	**91.15** (± 0.18)	**88.65** (± 0.21)	**91.04** (± 0.21)	**83.10** (± 0.23)
CIFAR100	SGD(CE)	68.21 (± 0.39)	56.29 (± 0.31)	61.93 (± 0.30)	0.45 (± 0.34)
	SGD(TCE)	65.12 (± 0.43)	59.61 (± 0.35)	64.98 (± 0.27)	47.20 (± 0.21)
	SGD(SCE)	68.28 (± 0.31)	58.82 (± 0.27)	63.20 (± 0.23)	46.18 (± 0.31)
	EG(SCE)	69.53 (± 0.18)	65.36 (± 0.22)	65.24 (± 0.19)	48.10 (± 0.19)
	ABSGD(SCE)	**71.73** (± 0.16)	**66.36** (± 0.21)	**66.32** (± 0.18)	**49.60** (± 0.22)

Table 7: Top-1 testing accuracy (%) on Clothing1M data of different methods. Results are reported over 3 independent runs.

loss	SGD	EG	ABSGD
CE	69.05 (± 0.21)	69.42 (± 0.20)	**69.79** (± 0.18)
SCE	69.31 (± 0.31)	69.32 (± 0.21)	**69.93** (± 0.11)
TCE	68.28 (± 0.23)	68.66 (± 0.19)	**68.69** (± 0.18)

the methods. The weight decay is set as $1e$-2. The λ for ABSGD is tuned in $\{-1, -5, -10, -15\}$. We report the results on CE, TCE, SCE optimized by standard SGD, EG and ABSGD, respectively.

Results. The results are reported in Table 7. We can see that ABSGD has better testing accuracy than SGD and EG for all losses.

6 Conclusion

In this paper, we propose a unified framework, ABSGD, for addressing the data imbalance and noisy label problem. We provide the theoretical analysis both for the SGD-style and Adam-style updates. Empirical studies on multiple benchmark datasets with different models show the outstanding performance of ABSGD compared with several strong baselines.

References

[1] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[2] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1701–1708, 2014.

[3] Omkar M Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. 2015.

[4] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *European conference on computer vision*, pages 499–515. Springer, 2016.

[5] 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.

[6] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. IEEE, 2013.

[7] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[8] Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

[9] Alex Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.

[10] Mirco Ravanelli, Philemon Brakel, Maurizio Omologo, and Yoshua Bengio. Light gated recurrent units for speech recognition. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2(2):92–102, 2018.

[11] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[12] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[13] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[14] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015.

[15] Ashish Vaswani, Samy Bengio, Eugene Brevdo, Francois Chollet, Aidan N Gomez, Stephan Gouws, Llion Jones, Łukasz Kaiser, Nal Kalchbrenner, Niki Parmar, et al. Tensor2tensor for neural machine translation. *arXiv preprint arXiv:1803.07416*, 2018.

[16] Justin M Johnson and Taghi M Khoshgoftaar. Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1):27, 2019.

[17] 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.

[18] Robin Chan, Matthias Rottmann, Fabian Hüger, Peter Schlicht, and Hanno Gottschalk. Application of decision rules for handling class imbalance in semantic segmentation. *arXiv preprint arXiv:1901.08394*, 2019.

[19] Alberto Fernández, Salvador Garcia, Francisco Herrera, and Nitesh V Chawla. Smote for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. *Journal of artificial intelligence research*, 61:863–905, 2018.

[20] Chao Huang, Xian Wu, Xuchao Zhang, Suwen Lin, and Nitesh V Chawla. Deep prototypical networks for imbalanced time series classification under data scarcity. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 2141–2144, 2019.

[21] Yuanrui FAN, Xin XIA, Daniel A COSTA, David LO, Ahmed E HASSAN, and Shanping LI. The impact of changes mislabeled by szz on just-in-time defect prediction.(2019). *IEEE Transactions on Software Engineering*, pages 1–26.

[22] Kim Herzig, Sascha Just, and Andreas Zeller. It's not a bug, it's a feature: how misclassification impacts bug prediction. In *2013 35th international conference on software engineering (ICSE)*, pages 392–401. IEEE, 2013.

[23] Sunghun Kim, Hongyu Zhang, Rongxin Wu, and Liang Gong. Dealing with noise in defect prediction. In *2011 33rd international conference on software engineering (ICSE)*, pages 481–490. IEEE, 2011.

[24] Andrew Bud. Facing the future: The impact of apple faceid. *Biometric Technology Today*, 2018(1):5–7, 2018.

[25] Daisuke Wakabayashi. Self-driving uber car kills pedestrian in arizona, where robots roam. *The New York Times*, 19, 2018.

[26] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *CoRR*, abs/1611.03530, 2016.

[27] 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.

[28] Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. Borderline-smote: a new over-sampling method in imbalanced data sets learning. In *International conference on intelligent computing*, pages 878–887. Springer, 2005.

[29] Nitesh V Chawla. Data mining for imbalanced datasets: An overview. In *Data mining and knowledge discovery handbook*, pages 875–886. Springer, 2009.

[30] Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2691–2699, 2015.

[31] Arash Vahdat. Toward robustness against label noise in training deep discriminative neural networks. *Advances in neural information processing systems*, 30, 2017.

[32] Kuang-Huei Lee, Xiaodong He, Lei Zhang, and Linjun Yang. Cleannet: Transfer learning for scalable image classifier training with label noise. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5447–5456, 2018.

[33] Andreas Veit, Neil Alldrin, Gal Chechik, Ivan Krasin, Abhinav Gupta, and Serge Belongie. Learning from noisy large-scale datasets with minimal supervision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 839–847, 2017.

[34] Dixian Zhu, Zhe Li, Xiaoyu Wang, Boqing Gong, and Tianbao Yang. A robust zero-sum game framework for pool-based active learning. In *The 22nd international conference on artificial intelligence and statistics*, pages 517–526, 2019.

[35] Zhi-Hua Zhou and Xu-Ying Liu. Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Transactions on knowledge and data engineering*, 18(1):63–77, 2005.

[36] Yanmin Sun, Mohamed S Kamel, Andrew KC Wong, and Yang Wang. Cost-sensitive boosting for classification of imbalanced data. *Pattern Recognition*, 40(12):3358–3378, 2007.

[37] Clayton Scott. Surrogate losses and regret bounds for cost-sensitive classification with example-dependent costs. In *Proceedings of International Conference of Machine Learning (ICML)*, pages 153–160, 2011.

[38] Shameem Puthiya Parambath, Nicolas Usunier, and Yves Grandvalet. Optimizing f-measures by cost-sensitive classification. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2123–2131, 2014.

[39] Harikrishna Narasimhan, Purushottam Kar, and Prateek Jain. Optimizing non-decomposable performance measures: A tale of two classes. In *International Conference on Machine Learning*, pages 199–208, 2015.

[40] Charles Elkan. The foundations of cost-sensitive learning. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence - Volume 2*, pages 973–978, 2001.

[41] Róbert Busa-Fekete, Balázs Szörényi, Krzysztof Dembczynski, and Eyke Hüllermeier. Online f-measure optimization. In *Advances in Neural Information Processing Systems*, pages 595–603, 2015.

[42] Yan Yan, Tianbao Yang, Yi Yang, and Jianhui Chen. A framework of online learning with imbalanced streaming data. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI)*, 2017.

[43] 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.

[44] Xi Yin, Xiang Yu, Kihyuk Sohn, Xiaoming Liu, and Manmohan Chandraker. Feature transfer learning for deep face recognition with under-represented data. *arXiv preprint arXiv:1803.09014*, 2018.

[45] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9268–9277, 2019.

[46] Muhammad Abdullah Jamal, Matthew Brown, Ming-Hsuan Yang, Liqiang Wang, and Boqing Gong. Rethinking class-balanced methods for long-tailed visual recognition from a domain adaptation perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7610–7619, 2020.

[47] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. *arXiv preprint arXiv:1803.09050*, 2018.

[48] Priyal Goyal and He Kaiming. Focal loss for dense object detection. *IEEE transactions on pattern analysis and machine intelligence*, 39:2999–3007, 2018.

[49] 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*, pages 1567–1578, 2019.

[50] Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey. Symmetric cross entropy for robust learning with noisy labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 322–330, 2019.

[51] Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in neural information processing systems*, 31, 2018.

[52] Shai Shalev-Shwartz and Yonatan Wexler. Minimizing the maximal loss: How and why. In *ICML*, pages 793–801, 2016.

[53] Hongseok Namkoong and John C Duchi. Variance-based regularization with convex objectives. In *Advances in neural information processing systems*, pages 2971–2980, 2017.

[54] Qi Qi, Yan Yan, Zixuan Wu, Xiaoyu Wang, and Tianbao Yang. a simple and effective framework for pairwise deep metric learning. *arXiv preprint arXiv:1912.11194*, 2019.

[55] Hamed Rahimian and Sanjay Mehrotra. Distributionally robust optimization: A review. *arXiv preprint arXiv:1908.05659*, 2019.

[56] Hassan Rafique, Mingrui Liu, Qihang Lin, and Tianbao Yang. Non-convex min-max optimization: Provable algorithms and applications in machine learning. *arXiv preprint arXiv:1810.02060*, 2018.

[57] Yan Yan, Yi Xu, Qihang Lin, Wei Liu, and Tianbao Yang. Sharp analysis of epoch stochastic gradient descent ascent methods for min-max optimization. *arXiv preprint arXiv:2002.05309*, 2020.

[58] Qi Qi, Zhishuai Guo, Yi Xu, Rong Jin, and Tianbao Yang. A practical online method for distributionally deep robust optimization. *arXiv preprint arXiv:2006.10138*, 2020.

[59] Tian Li, Ahmad Beirami, Maziar Sanjabi, and Virginia Smith. On tilted losses in machine learning: Theory and applications, 2021.

[60] Qi Qi, Jiameng Lyu, Er Wei Bai, Tianbao Yang, et al. Stochastic constrained dro with a complexity independent of sample size. *arXiv preprint arXiv:2210.05740*, 2022.

[61] John C Duchi and Hongseok Namkoong. Learning models with uniform performance via distributionally robust optimization. *The Annals of Statistics*, 49(3):1378–1406, 2021.

[62] Daniel Levy, Yair Carmon, John C Duchi, and Aaron Sidford. Large-scale methods for distributionally robust optimization. *Advances in Neural Information Processing Systems*, 33:8847–8860, 2020.

[63] Jikai Jin, Bohang Zhang, Haiyang Wang, and Liwei Wang. Non-convex distributionally robust optimization: Non-asymptotic analysis. *Advances in Neural Information Processing Systems*, 34:2771–2782, 2021.

[64] Boris T Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

[65] E Choi, P Hall, and B Presnell. Rendering parametric procedures more robust by empirically tilting the model. *Biometrika*, 87(2):453–465, 06 2000.

[66] Negin Majidi, Ehsan Amid, Hossein Talebi, and Manfred K. Warmuth. Exponentiated gradient reweighting for robust training under label noise and beyond. *CoRR*, abs/2104.01493, 2021.

[67] Mengdi Wang, Ethan X Fang, and Han Liu. Stochastic compositional gradient descent: algorithms for minimizing compositions of expected-value functions. *Mathematical Programming*, 161(1-2):419–449, 2017.

[68] Saeed Ghadimi, Andrzej Ruszczynski, and Mengdi Wang. A single timescale stochastic approximation method for nested stochastic optimization. *SIAM Journal on Optimization*, 30(1):960–979, 2020.

[69] Tianyi Chen, Yuejiao Sun, and Wotao Yin. Solving stochastic compositional optimization is nearly as easy as solving stochastic optimization. *arXiv preprint arXiv:2008.10847*, 2020.

[70] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.

[71] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

[72] Zhishuai Guo, Yi Xu, Wotao Yin, Rong Jin, and Tianbao Yang. A novel convergence analysis for algorithms of the adam family. *arXiv preprint arXiv:2112.03459*, 2021.

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

[74] Prakash M Nadkarni, Lucila Ohno-Machado, and Wendy W Chapman. Natural language processing: an introduction. *Journal of the American Medical Informatics Association*, 18(5):544–551, 2011.

[75] Yue Kang, Zhao Cai, Chee-Wee Tan, Qian Huang, and Hefu Liu. Natural language processing (nlp) in management research: A literature review. *Journal of Management Analytics*, 7(2):139–172, 2020.

[76] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. *arXiv preprint arXiv:1910.09217*, 2019.

[77] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017.

[78] iNatrualist. The inaturalist 2018 competition dataset. 2018.

[79] 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.

[80] 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.

[81] Xiao Zhang, Zhiyuan Fang, Yandong Wen, Zhifeng Li, and Yu Qiao. Range loss for deep face recognition with long-tailed training data. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5409–5418, 2017.

[82] Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1944–1952, 2017.

[83] Zhuoning Yuan, Yan Yan, Rong Jin, and Tianbao Yang. Stagewise training accelerates convergence of testing error over sgd. In *Advances in Neural Information Processing Systems*, pages 2604–2614, 2019.

[84] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.