Unifying Distribution Alignment as a Loss for Imbalanced Semi-supervised Learning

Justin Lazarow[†], [*] Kihyuk Sohn[‡], Chen-Yu Lee[‡], Chun-Liang Li[‡], Zizhao Zhang[‡], Tomas Pfister[‡]
UC San Diego[†], Google Cloud AI Research[‡]
jlazarow@eng.ucsd.edu, {kihyuks,chenyulee,chunliang,zizhaoz,tpfister}@google.com

Abstract

While remarkable progress has been made in imbalanced supervised learning, less attention has been given to the setting of imbalanced semi-supervised learning (SSL) where not only are few labeled data provided, but the underlying data distribution can be severely imbalanced. Recent work requires both complicated sampling strategies of pseudo-labeled unlabeled data and distribution alignment of the pseudo-label distribution to accommodate this imbalance. We present a novel approach that relies only on a form of a distribution alignment but no sampling strategy where rather than aligning the pseudo-labels during inference, we move the distribution alignment component into the respective cross entropy loss computations for both the supervised and unsupervised losses. This alignment compensates for both imbalance in the data and the eventual distributional shift present during evaluation. Altogether, this provides a unified strategy that offers both significantly reduced training requirements and improved performance across both low and richly labeled regimes and over varying degrees of imbalance. In experiments, we validate the efficacy of our method on SSL variants of CIFAR10-LT, CIFAR100-LT, and ImageNet-127. On ImageNet-127, our method shows 1.6% accuracy improvement over CReST with an 80% training time reduction and is competitive with other SOTA methods. Code is available at `https://github.com/google-research/crest`

1. Introduction

Semi-supervised learning (SSL) leverages a large pool of unlabeled data to learn a classifier despite having access to only a small amount of labeled data. Recently, techniques have been introduced [4, 3, 27], which simplify the process while pushing accuracy to new levels. However, they have focused on the cases where the class distributions are balanced for both the labeled and unlabeled data.

Meanwhile, supervised learning community has shown renewed focus on imbalanced learning [15, 24, 26, 12], owing to the fact that most real world data is not well-balanced. Many have observed that ordinary supervised learning techniques suffer from the bias favoring head classes over tail classes. To mitigate the bias, several methods have been introduced, including data resampling [8, 11, 5, 23, 6], loss reweighting [17, 9, 16, 7], representation and classifier decoupling [15], and logit adjustment [24, 26, 12]. However, these methods require data labels and are not readily applicable when majority of the data remains unlabeled.

In this work, we study an important and practical problem of imbalanced semi-supervised learning, where class distributions of both labeled and unlabeled are imbalanced. We develop our method upon state-of-the-art consistency-based semi-supervised learners, such as FixMatch [27] and MixMatch [4], which employ two losses on the labeled data and unlabeled data with pseudo labels. Therefore, vulnerability to bias from the imbalance can happen in three ways: the supervised loss, the quality of pseudo-labels derived from the classifier on unlabeled data, and the unsupervised

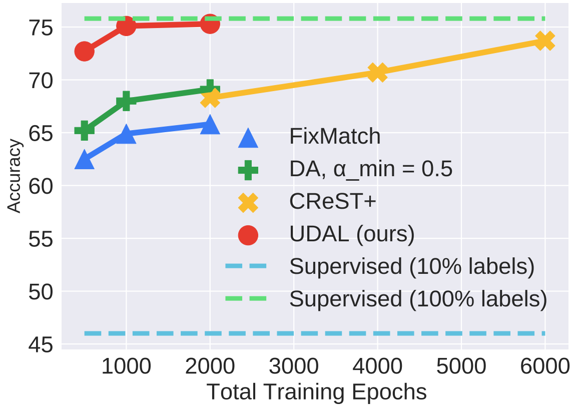


Figure 1: **Classification accuracy on ImageNet-127 with 10% of labels**. The proposed UDAL trained for only $1/6$ of the epochs of CReST+ [29] further closes the accuracy gap to a fully supervised baseline and heavily improves on a supervised baseline trained on 10% of the labels.

[*] Work done during internship at Google Cloud AI Research.

loss using pseudo-labels even if they are correctly predicted.

CReST [29] offers a solution to reduce the bias caused by issues from unlabeled data. Observing the high precision of unlabeled data from the tail classes, CReST has proposed a class-balanced sampling strategy that samples more from the tail classes than head classes. This not only improves the quality of pseudo-labels, but also results in more balanced label distribution. Distribution alignment [3], which modifies the prediction by the ratio of the desired distribution to model distribution, is at its core to progressively rebalance the pseudo-label distribution. On the other hand, it also requires a generational approach that accumulates a relatively balanced subset of confident pseudo-labels over a multiple steps. Each generation re-initializes the network and therefore the process is costly with respect to training time.

Is this disjoint methodology truly necessary? Can a single, central approach be devised to address class imbalance in semi-supervised learning? We identify an affirmative answer to these questions. First, we connect the two separate ideas of progressive distribution alignment [3, 29] from semi-supervised learning and logit adjustment [24, 26, 12] from imbalanced supervised learning. Based on this observation, we propose a *progressive logit adjustment*, a unified approach of the two aforementioned techniques for imbalanced semi-supervised learning. Compared to CReST [29], our approach is more efficient in training time as it does not require resampling or many steps with re-initialization, while achieving competitive or higher accuracy, as in Figure 1. Moreover, our method is extremely simple, requiring only a few lines of change from the existing consistency-based semi-supervised learning implementations.

2. Prerequisites

We present prerequisites on both problem settings. First, we formally define the problem setting of Imbalanced Semi-Supervised Learning. Second, we outline the idea of distribution alignment [3, 29] that improves pseudo-label quality within both the balanced and imbalanced settings of SSL. We revisit a recent method, CReST [29], which attempts to address imbalanced semi-supervised learning.

2.1. Class-imbalanced Semi-Supervised Learning

Semi-supervised learning relies on two sources of data: a *labeled* set $\mathcal{X} = \{(x_i, y_i)\}_{i=1}^N$ where each x_i is a training example and y_i is the corresponding target. Since classification is the focus of this work, we consider y_i as a class label within $\mathcal{C} = \{1, \ldots, C\}$ with a total number of C classes. In imbalanced learning, we expect varying numbers of training examples across classes. Therefore, we denote the number of examples in our labeled set corresponding to class $c \in \mathcal{C}$ as N_c such that $\sum_{c=1}^C N_c = N$. We assume that the classes are ordered with respect to frequencies and in a descending manner *i.e.*, $N_c \geq N_{c+1}$. It is often useful to characterize the degree of imbalance by the ratio N_1/N_C, and we refer to this as the *imbalance ratio* of the dataset. We use $p_{\text{data}}(y)$ and $q(y)$ to denote the marginal distributions of the data and the model. When there is no ambiguity, we drop y as p_{data} and q to simplify presentations.

In addition, we have an *unlabeled* set $\mathcal{U} = \{u_i\}_{i=1}^M$ for which we have no corresponding target. We follow the task setup in [18, 22, 29] and utilize the same imbalance ratio for both labeled and unlabeled sets, but also perform experiments when the marginal class distribution of unlabeled data is different from that of labeled data and unknown in Section 4.3.4. $\beta = \frac{N}{N+M}$ denotes the labeled data ratio.

2.2. Distribution alignment

Distribution alignment (DA) was introduced for the SSL setting within ReMixMatch [3] to reduce the confirmation bias [2] via regularization added to the pseudo-label inference. In particular, if we assume the labeled and unlabeled data both come from the same distribution p_{data}, we would expect that the model should produce pseudo-labels that follow the same distribution. [3] has proposed to estimate the marginal distribution of the model $q(y)$ by the moving average, which we denote as $\hat{q}(y)$ or $\hat{q}$. If we denote our current model's predictions on unlabeled examples as $q(y|x_u)$, these predictions can be re-scaled through dividing by $\hat{q}$ and multiplying by p_{data}. After normalization, we have:

$$\tilde{q}(y|x) = \text{Normalize}\left(q(y|x)\,\frac{p_{\text{data}}}{\hat{q}}\right). \tag{1}$$

In (1), we assume element-wise operations beween $q(y|x)$, p_{data} and $\hat{q}$. Normalize(p) ensures p as a probability distribution which sums to 1.

As noted in [29], it is not always optimal to align the predictions directly to p_{data} when p_{data} is imbalanced. Rather, a smoothed form which (elementwise) exponentiates the distribution by a factor of α before normalization:

$$\tilde{p}_\alpha = \text{Normalize}\left(p_{\text{data}}^\alpha\right),\ 0 \leq \alpha \leq 1. \tag{2}$$

is used instead of p_{data} in Equation (1) and found to both regularize predictions and combat bias. As $\alpha \rightarrow 0$, this approaches an alignment against a more uniform distribution.

2.3. Logit Adjustment

While DA is clearly applicable to pseudo-label *inference* during training, it has no direct impact on the labeled portion. Since a semi-supervised approach relies on both labeled and unlabeled losses, it is critical to address the problem of imbalance at the supervised level as well. For this, we examine a popular technique within supervised learning, often known as logit adjustment [24], balanced softmax [26], or LADE [12]. These methods modify the *loss computation* to compensate for the class imbalance found in the data distribution. Notably, when a data distribution is class-imbalanced, we attempt to minimize the classification loss with respect to this data distribution. *However*, at evaluation time, we either evaluate on a class-balanced dataset or produce a class-balanced error by averaging the per-class accuracies. This is a shift in distribution which can cause poor performance. Therefore, this shift is integrated into the cross entropy loss:

$$\begin{aligned}\mathcal{L}_{LA}\left(y, f(x)\right) &= \mathcal{L}_{CE}\big(y, f(x) + \log p_{\text{data}} - \log\left(\text{Unif}(C)\right)\big) \\ &\equiv \mathcal{L}_{CE}\big(y, f(x) + \log p_{\text{data}}\big)\end{aligned} \tag{3}$$

where Unif(C) is the discrete uniform distribution over C classes, y is the true label of x, $f(x)$ is the vector-valued output of the classifier, and p_{data} is the marginal class distribution of the data as a vector. As elaborated within [24], this has the effect that instead of optimizing $f(x)$ directly, we optimize $h(x) = f(x) + \log p_{\text{data}}$ which aligns the source distribution of $f(x)$ correctly to the uniform class distribution. [24] also discusses an inference time procedure which attempts to account for this shift without modifications to the training procedure. Since this is "for free", we include experimental results combined with it in Table 1 as "LA (Inf)" for the inference time logit adjustment.

2.4. CReST

We briefly examine CReST [29], an approach to imbalanced semi-supervised learning using FixMatch and MixMatch as a base semi-supervised learner. They re-introduce a generational self-training approach where after each generation (usually 64 epochs), confidently pseudo-labeled examples from the unlabeled set $\mathcal{U}$ are added to $\mathcal{X}$. Ordinarily, this would exacerbate class imbalance. [29] introduce a resampling strategy according to a more balanced form of the marginal class distribution of the data to overcome. However, it requires to *reinitialize and train the network from scratch* after each generation.

Additionally, for optimal performance, CReST requires distribution alignment to improve pseudo-label quality. Specifically, as defined in Section 2.2, a *schedule* for α is chosen so that the strength of re-balancing can be altered over the course of training. Given a hyperparameter α_{min} which defines the minimal value α should take (corresponding to the largest re-balancing of the data distribution), they choose a linear schedule such that for generation iteration t:

$$\alpha_t = 1.0 - \left(1.0 - \alpha_{\text{min}}\right)\frac{t}{T} \tag{4}$$

where T is the number of generations over the course of training. This is referred to as **progressive distribution alignment** (PDA) and is essential to strong performance.

3. Proposed Method

We describe our approach, Unifying Distribution Alignment as a Loss (UDAL). First, we connect the ideas behind distribution alignment [3] and logit adjustment [24] in Section 3.1. Notably, in Section 3.2, we introduce a progressive form of logit adjustment and examine how it applies a form of distribution alignment *at the loss level*. Subsequently, we examine how this allows us to apply the same mitigation to both the supervised and unsupervised components of modern consistency-based SSL methods [4, 27] – providing a unified manner in which imbalance can be addressed with respect to labeled and unlabeled data. Figure 2 shows the comparison between UDAL and previous work. Finally, in Section 3.3, we discuss variants of our method.

3.1. Connection between DA and LA

As in Section 2.2 and 2.4, applying distribution alignment (DA) modifies predictions of unlabeled data by aligning them to a *target distribution*. DA modifies the pseudo-label generation process for unsupervised branch, but does not affect the known labels of the supervised branch.

[24] demonstrates that the logit adjustment can be used to align a supervised classifier to the uniform distribution. On the other hand, we argue that the logit adjustment can be used to align to an arbitrary target distribution like in DA. For example, if we replace $\text{Unif}(C)$ into $\tilde{p}_\alpha$ in Equation 3, we get the following form of logit adjustment:

$$\mathcal{L}_{CE}\left(y, f(x) + \log p_{\text{data}} - \log \tilde{p}_\alpha\right) \tag{5}$$

What kind of learner does this form attempt to produce? Defining $h(x) = f(x) + \log\left(p_{\text{data}}/\tilde{p}_\alpha\right)$ and solving for $f(x)$, this implies that we are optimizing the following:

$$f(x) = h(x) + \log\left(\frac{\tilde{p}_\alpha}{p_{\text{data}}}\right) \tag{6}$$

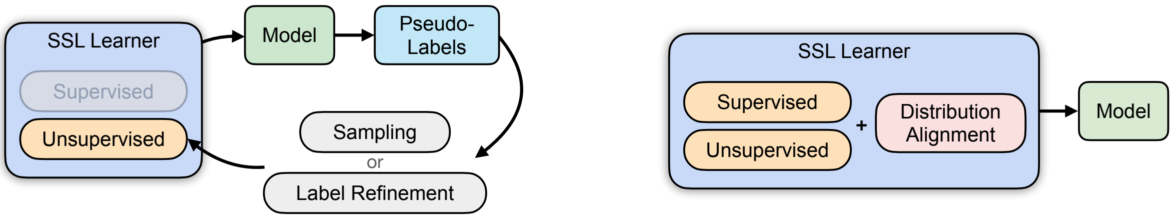
or rather, in the softmax space, we get

$$\text{Softmax}(f(x)) = \text{Softmax}\left(h(x)\right)\left(\frac{\tilde{p}_\alpha}{p_{\text{data}}}\right) \tag{7}$$

which is exactly the form of applying distribution alignment to $f(x)$ using $\tilde{p}_\alpha$. In other words, distribution alignment [3] and logit adjustment [24] are just two instantiations of the same idea, one as a sampling method for unlabeled data in semi-supervised learning and another as a training loss for labeled data in supervised learning, respectively.

3.2. A unified approach

Based on our finding in Section 3.1, we are now ready to introduce our method that unifies the two into one framework. We will use FixMatch [27] as a base semi-supervised learner for presentation, but our method is also applicable to



(a) Previous work

(b) Ours

Figure 2: Given an off-the-shelf Semi-Supervised Learning (SSL) learner, previous works address the data imbalanced issue on the unsupervised loss by *iterative* (a) class-rebalancing sampling [29] or pseudo-label refinement [18]. In this work, we propose to tackle the imbalance issue on (b) both supervised and unsupervised losses by *directly* performing distribution-aligned learning.

other consistency-based SSL methods, including MixMatch [4], derived in Section 3.3. FixMatch consists of cross entropy losses on the labeled data and on unlabeled data with respect to inferred pseudo-labels:

$$\begin{aligned}\mathcal{L}_{\text{FixMatch}} =&\mathbb{E}_{x,y\sim p_l}\left[\mathcal{L}_{\text{CE}}(y, f(x))\right] + \\ &\mathbb{E}_{u\sim p_u}\left[\mathcal{L}_{\text{CE}}\left(\text{PL}(f(u^{\text{w}})), f(u^{\text{s}})\right)\right]\end{aligned} \tag{8}$$

where $f(x)$ is the network's outputs, p_l is labeled data distribution, p_u is the unlabeled data distribution, PL produces hard pseudo-labels for confident predictions, while u^{s} and u^{w} are strong and weakly augmented versions of the unlabeled example u, respectively.

First, we introduce a **progressive logit adjustment** that extends logit adjustment to align supervised classifier to a progressively changing target distribution $\tilde{p}_{\alpha_t}$:

$$\mathcal{L}_{\text{CE}}\big(y, f(x) + \log\left(p_{\text{data}}\right) - \log\left(\tilde{p}_{\alpha_t}\right)\big) \tag{9}$$

Moreover, we hypothesize that not only can we align the supervised branch in this manner, but that we can also apply this exact same form to the unsupervised branch:

$$\mathcal{L}_{\text{CE}}\big(\text{PL}(f(u^{\text{w}})), f(u^{\text{s}}) + \log\left(\hat{q}\right) - \log(\tilde{p}_{\alpha_t})\big) \tag{10}$$

where we divide by the estimated model's class distribution $\hat{q}$ through moving average as explained in Section 2.2, rather than p_{data} since the unsupervised branch is learned from pseudo-labels drawn directly from the model rather than the data distribution. *This allows us to use the same, unified approach for each branch* by simply replacing the distribution we are aligning from: the marginal class distribution of the data for the supervised branch and the moving average of pseudo-labels for the unsupervised branch. Altogether, this results in a loss:

$$\begin{aligned}&\mathcal{L}_{\text{UDAL}} = \\ &\mathbb{E}_{x,y\sim p_l}\left[\mathcal{L}_{\text{CE}}(y, f(x) + \log\left(p_{\text{data}}\right) - \log\left(\tilde{p}_{\alpha_t}\right))\right] + \\ &\mathbb{E}_{u\sim p_u}\left[\mathcal{L}_{\text{CE}}\left(\text{PL}(f(u^{\text{w}})), f(u^{\text{s}}) + \log\left(\hat{q}\right) - \log\left(\tilde{p}_{\alpha_t}\right)\right)\right]\end{aligned} \tag{11}$$

where α_t is generalized from Equation (4) as:

$$\alpha_t = 1.0 - (1.0 - \alpha_{\min})\left(\frac{t}{T}\right)^k \tag{12}$$

to allow for an alignment rate k. $\tilde{p}_{\alpha_t}$ is computed as in Equation (2). We provide the pseudocode of our approach in Algorithm 1. This constitutes of *only a few lines of code* with no alterations to the training scheme or time compared to the base FixMatch and MixMatch learner. Moreover, no resampling is required as in [29].

3.3. Discussion

Simple Baselines. While our method may seem like a simple combination of two existing methods, distribution alignment and logit adjustment, we argue that UDAL is an elegant formulation inspired by the novel insight that connects the two. For example, one may consider a simple combination by applying the progressive logit adjustment in Equation (9) to the supervised branch alongside progressive distribution alignment from [29] to the unsupervised branch. However, we find that, through an extensive ablation study in Section 4.4.1, such naive combinations do not fully leverage their benefits and fall short of UDAL.

Unknown Unlabeled Data Distribution. The formulation of UDAL contains an estimate of the label distribution p_{data}, which is estimated from a small labeled data. While our assumption on the label distribution of labeled and unlabeled sets being equivalent might be true in certain scenarios (*e.g.*, labeled data is randomly drawn from the entire data population), there are also cases where the label distributions of labeled and unlabeled are different.

In other words, $\tilde{p}_{\alpha_t}$ is unknown. On the other hand, there are methods that estimate the class distribution of unlabeled data using a small labeled set, such as DARP [18], which is directly applicable to UDAL as well. Let $\hat{p}_{\text{data}}$ an estimate of class distribution of unlabeled data. Finally, we replace p_{data} of Equation (1) with $\hat{p}_{\text{data}}$ when aligning the unsupervised branch and optimize Equation (11). Empirical validation

Algorithm 1 UDAL Pseudocode, TensorFlow-ish

```
# p_data: class distribution of labeled data
# p_model: moving average of model’s predictions on
    unlabeled data
# current_epoch (g): current epoch of training (out
    of max_epoch total)
# k: rate at which a_min is approached
# a_min: minimum value of alpha for dist. alignment
def compute_adjustment_dist(current_dist):
   factor = 1.0 - (1.0 - a_min) * (current_epoch /
       max_epoch) ** k
   # normalize ensures the argument sums to 1
   target_dist = normalize(p_data ** factor)
   return current_dist / (target_dist + 1e-9)

# supervised (labeled) loss on examples x_l
# f: classifier, y_l: true labels
loss_l = SoftmaxCE(y_l, f(x_l) + log(
    compute_adjustment_dist(p_data)))

# unsupervised (unlabeled) loss on examples x_u
# y_u: PLs predicted from weakly-augmented x_u after
     confidence thresholding
loss_u = SoftmaxCE(y_u, f(x_s) + log(
    compute_adjustment_dist(p_model)))
```

of UDAL for unknown class distribution of unlabeled set is provided in Section 4.3.4.

UDAL on MixMatch [4]. As mentioned earlier, UDAL is not specific to FixMatch, but generally applicable to modern consistency-based SSL learners. Here, we derive its application to MixMatch, another state-of-the-art consistency-based SSL method based on MixUp [30], instead of strong data augmentation specific to image domains.

Let $(x, y) \sim p_l$ and $u \sim p_u$. Let $\text{SPL}(f(u))$ a soft pseudo label of an unlabeled data u with model f. Following [4], let $(\tilde{x}, \tilde{y})$ and $(\tilde{u}, \widetilde{\text{SPL}}(f(u)))$ MixUp augmented labeled and unlabeled data, respectively. The training objective of MixMatch is given as follows:

$$\begin{aligned}\mathcal{L}_{\text{MixMatch}} = &\mathbb{E}_{\tilde{x},\tilde{y}}\left[\mathcal{L}_{\text{CE}}(\tilde{y}, f(\tilde{x})\right] + \\ &\mathbb{E}_{\tilde{u}}\left[\|\widetilde{\text{SPL}}(f(u)) - \text{Softmax}\left(f(\tilde{u})\right)\|_2^2\right]\end{aligned} \tag{13}$$

and applying UDAL, we get the following training loss:

$$\begin{aligned}&\mathbb{E}_{\tilde{x},\tilde{y}}\left[\mathcal{L}_{\text{CE}}(\tilde{y}, f(\tilde{x}) + \log\left(p_{\text{data}}\right) - \log\left(\tilde{p}_{\alpha_t}\right)\right] + \\ &\mathbb{E}_{\tilde{u}}\left[\|\widetilde{\text{SPL}}(f(u)) - \text{Softmax}\left(f(\tilde{u}) + \log\left(\hat{q}\right) - \log\left(\tilde{p}_{\alpha_t}\right)\right)\|_2^2\right]\end{aligned} \tag{14}$$

4. Experiments

We follow the experimental settings outlined in CReST [29]. First, we conduct experiments under the assumption that marginal class distributions of labeled and unlabeled data are known to be equal. Then, we present results of our method evaluated when the marginal class distribution of unlabeled data is different from that of labeled data and unknown in Section 4.3.4. We test the efficacy of UDAL over various long-tailed versions of CIFAR10 and CIFAR100.

Method	$\gamma = 50$	$\gamma = 100$	$\gamma = 200$
MixMatch [4]	$69.1_{\pm 1.18}$	$60.4_{\pm 2.24}$	$54.5_{\pm 1.87}$
w/ CReST+ [29]	$76.7_{\pm 0.35}$	$66.1_{\pm 0.79}$	$57.6_{\pm 1.30}$
w/ UDAL (ours)	**77.8**$_{\pm 0.88}$	**68.4**$_{\pm 1.48}$	**58.6**$_{\pm 1.10}$
FixMatch [27]	$80.1_{\pm 0.44}$	$67.3_{\pm 1.19}$	$59.7_{\pm 0.63}$
w/ DA [3] ($\alpha_{\text{min}} = 0.5$)	$82.4_{\pm 0.33}$	$73.6_{\pm 0.63}$	$63.7_{\pm 1.17}$
w/ DA ($\alpha_{\text{min}} = 0.5$) + LA (Sup)	$83.5_{\pm 0.19}$	$75.7_{\pm 1.56}$	$65.7_{\pm 1.87}$
w/ LA (Inf) [24]	$83.2_{\pm 0.87}$	$70.4_{\pm 2.90}$	$62.4_{\pm 1.24}$
w/ CReST+ [29]	$84.2_{\pm 0.39}$	$78.1_{\pm 0.84}$	$67.7_{\pm 1.39}$
w/ CReST+ & LA (Inf) [29]	$85.6_{\pm 0.36}$	$81.2_{\pm 0.70}$	$71.9_{\pm 2.24}$
w/ UDAL (ours)	$85.3_{\pm 0.34}$	$80.2_{\pm 0.59}$	$68.6_{\pm 1.32}$
w/ UDAL & LA (Inf) (ours)	**86.3**$_{\pm 0.37}$	**82.1**$_{\pm 0.37}$	**72.9**$_{\pm 1.21}$

Table 1: Accuracy on CIFAR10-LT at $\beta = 10\%$ with various γ. We test from the simplest approaches that involve no accommodations for class imbalance to stronger baselines which attempt to address it. We include two types of logit adjustment [24], “LA (Sup)”, that modifies the loss to the supervised branch and “LA (Inf)” that does not modify training but applies an adjustment at inference time only.

Finally, we perform large-scale experiments on the naturally long-tailed ImageNet-127 dataset.

4.1. Dataset creation

CIFAR10-LT and CIFAR100-LT [7, 9] are modifications of CIFAR10 and CIFAR100 with a long-tailed (Zipfian) distribution. Given a desired imbalance ratio γ and some class ordering C_i, $1 \leq i \leq C$, we sample N_i examples from the dataset for class C_i according to $N_i = N_1 * \gamma^{\frac{C_i - 1}{C - 1}}$. For CIFAR10, $N_1 = 5000$, $C = 10$ while $N_1 = 500$ and $C = 100$ for CIFAR100. To create suitable splits for SSL, labeled subsets are randomly sampled according to $\beta = 10\%$ and 30%. Imbalance ratios of $\gamma \in \{50, 100, 200\}$ are explored for CIFAR10-LT while $\gamma \in \{50, 100\}$ for CIFAR100-LT. We test with respect to the original, balanced test set which ensures that we can use ordinary accuracy metrics.

4.2. Training

We follow the model guidelines in CReST [29]. We use Wide ResNet-28-2 as a backbone. FixMatch [27] and MixMatch [4] are used as base semi-supervised learners. For FixMatch, the unlabeled ratio (7) and confidence threshold (0.95) are untouched from their original settings. The same cosine learning rate schedule is adopted as in [29]. Notably different from [29], however, is our ability to use vastly shorter training times. For example, we find only 64 epochs (2^{16} iterations at a batch size of 64) is needed to reach optimal performance, and therefore all CIFAR models are trained for 64 epochs — constituting *a 5× reduction in training time* compared to CReST. Like CReST [29], we find it useful to train MixMatch models slightly longer and use a less aggressive α_{min}, therefore, MixMatch models are

	CIFAR10-LT						CIFAR100-LT			
	$\beta = 10\%$			$\beta = 30\%$			$\beta = 10\%$		$\beta = 30\%$	
Method	$\gamma = 50$	$\gamma = 100$	$\gamma = 200$	$\gamma = 50$	$\gamma = 100$	$\gamma = 200$	$\gamma = 50$	$\gamma = 100$	$\gamma = 50$	$\gamma = 100$
FixMatch [27]	79.4$_{\pm 0.65}$	66.3$_{\pm 1.74}$	59.7$_{\pm 0.74}$	81.9$_{\pm 0.30}$	73.1$_{\pm 0.58}$	64.7$_{\pm 0.69}$	33.7$_{\pm 0.94}$	28.3$_{\pm 0.66}$	43.1$_{\pm 0.24}$	38.6$_{\pm 0.45}$
w/ CReST+ [29]	84.2$_{\pm 0.39}$	78.1$_{\pm 0.84}$	67.7$_{\pm 1.39}$	84.9$_{\pm 0.27}$	79.2$_{\pm 0.20}$	70.5$_{\pm 0.56}$	38.8$_{\pm 1.03}$	**34.6**$_{\pm 0.74}$	46.7$_{\pm 0.34}$	42.0$_{\pm 0.44}$
w/ UDAL (ours)	**85.3**$_{\pm 0.34}$	**80.2**$_{\pm 0.59}$	**68.6**$_{\pm 1.32}$	**86.7**$_{\pm 0.34}$	**82.4**$_{\pm 0.43}$	**74.5**$_{\pm 1.13}$	**39.8**$_{\pm 0.88}$	34.3$_{\pm 0.85}$	**48.0**$_{\pm 0.56}$	**43.7**$_{\pm 0.41}$

Table 2: Classification accuracy (%) over CIFAR10-LT and CIFAR100-LT under a variety of label fractions β and imbalance ratios γ, each averaged over 5 folds.

trained for 128 epochs and use $\alpha_{\text{min}} = 0.5$.

Apart from those of the base semi-supervised learner, UDAL introduces only two hyperparameters: α_{min} which controls the final strength of the re-balancing for $\tilde{p}_{\alpha_t}$ and k, which controls the rate at which we approach α_{min}. We use $\alpha_{\text{min}} = 0.10$ and $k = 2$ which allows most of training to be aligned to p_{data} and spend the very last stages of training aligning to a relatively balanced class distribution. This is supported by other work [15] which empirically finds that when training long-tailed, fully-supervised models, the bulk of training should be done with respect to random sampling, and only a small amount of time of class-balanced sampling is necessary near the end of training. We provide an ablation of these hyperparameters in Section 4.4.2.

Training is carried out over 5 random folds of the data. We report final test accuracy along with standard deviation using the exponential moving average of the model's parameters. We use the same underlying codebase, written in TensorFlow [1], and data splits as CReST to prevent framework or dataset dependent uncertainty in performance.

4.3. Results

4.3.1 CIFAR.

CIFAR10-LT and CIFAR100-LT results are in Table 2. Across the board, we find UDAL outperforms CReST+ [29] and is worse in only a single setting. Notably, we find that CReST+ *especially struggles* in the more label rich regimes. In particular, despite a heavy imbalance of $\gamma \in \{100, 200\}$, UDAL significantly outpeforms CReST+ when $\beta = 30\%$ for both CIFAR10-LT (up to a 4 point increase) and CIFAR100-LT (up to a 1.7 point increase). We hypothesize that CReST+ is unable to efficiently use more labeled data since it only weakly addresses imbalance in the supervised branch. The imbalance in the supervised branch can only be indirectly alleviated by adding pseudo-labeled to the dataset under some amount of resampling. While this could have a high impact when there is scarce labeled data to begin with, it appears to have much less of an effect when more labeles are available. UDAL, however, directly aligns the supervised branch to a more balanced distribution over the course of training. Not only would this improve the balanced performance of the classifier in the fully supervised setting, but it additionally aids the feedback loop to provide more balanced pseudo-labels in the unsupervised branch.

To understand where the improvement comes from, we report the per-class recall of CReST+ and UDAL in Table 3. We find that, compared to CReST+, UDAL appears to be better at re-balancing the few-shot classes. We think that this could be due to UDAL having a more direct way of mitigating the contribution of imbalance from the loss rather than sampling as in CReST+.

Method / Freq.	1- 3 (many)	4-6 (med)	7-10 (few)
CReST+	**92.9**	**77.9**	71.0
UDAL	90.1	77.1	**74.1**

Table 3: Per-class recall (%) on the balanced test set of CIFAR10-LT ($\gamma = 100$, $\beta = 10\%$).

4.3.2 DARP.

We show results within the DARP [18] setting in Table 4. While DARP also produces an imbalanced dataset from CIFAR10, it has slight differences. While the CReST [29] setting assigns every example from the original CIFAR dataset to *either* the labeled or unlabeled dataset, DARP uses an unlabeled to labeled ratio of 2:1, which may not utilize all data points at training. One can consider the DARP setting as a $\beta = 33\%$ but with only 95% of the original dataset.

Nonetheless, UDAL continues to significantly outperforms CReST+ in all settings and shows the largest gains of all in the most imbalanced settings – capable of taking advantage of the increased amount of labeled data.

Method	$\gamma = 50$	$\gamma = 100$	$\gamma = 150$
FixMatch [27]	79.2$_{\pm 0.33}$	71.5$_{\pm 0.72}$	68.4$_{\pm 0.15}$
w/ DARP [18]	81.8$_{\pm 0.24}$	75.5$_{\pm 0.05}$	70.4$_{\pm 0.25}$
w/ CReST+ [29]	83.9$_{\pm 0.14}$	77.4$_{\pm 0.36}$	72.8$_{\pm 0.58}$
w/ DASO* [25]	-	79.1$_{\pm 0.75}$	75.1$_{\pm 0.77}$
w/ ABC* [22]	-	81.5$_{\pm 0.29}$	-
w/ UDAL (ours)	**86.5**$_{\pm 0.29}$	81.4$_{\pm 0.39}$	**77.9**$_{\pm 0.33}$

Table 4: Classification accuracy (%) under DARP's protocol [18] for CIFAR10. Numbers with * are taken from the original papers.

	Epochs			
Method	1000	2000	4000	6000
Supervised (100% labels)	75.8			
Supervised (10% labels)	46.0			
Supervised (10% labels, RandAug)	50.0			
Supervised (10% labels, LA)	54.0			
Supervised (10% labels, RandAug, LA)	60.0			
FixMatch (10% labels)	64.9	65.8	-	-
w/ DA ($\alpha_{\min} = 0.5$)	68.0	69.1	-	-
w/ CReST+	-	68.3	70.7	73.7
w/ UDAL (ours, $\alpha_{\min} = 0.5$)	74.5	74.3	-	-
w/ UDAL (ours, $\alpha_{\min} = 0.55$)	**75.1**	74.8	-	-
w/ UDAL (ours, $\alpha_{\min} = 0.6$)	73.2	**75.3**	-	-

Table 5: Evaluating UDAL on ImageNet127 with imbalance factor $\gamma = 286$ where $\beta = 10\%$ samples are labeled. Supervised models are trained for 300 epochs with 100% labeled data or 3000 epochs with 10% labeled data. Logit adjustment (LA) is applied at inference time.

4.3.3 ImageNet-127.

We provide experimental results on ImageNet-127 in Table 5. As presented in [29], ImageNet-127 is a coarser version of ImageNet [19], containing the same number of examples but with 127 class groupings rather than 1000. This grouping results in an imbalance ratio of $\gamma = 286$. We conduct experiments with $\beta = 10\%$ along with all base training settings identical to CReST+, with exception to the duration of training, where we observe that only 1000 epochs ($1/6$ the total training time of CReST+) is necessary for convergence (Figure 1). For UDAL, we continue to find $k = 2$ to be optimal, however, a less aggressive $\alpha_{\min} \in [0.5, 0.6]$ was found to be more effective in this setting. Overall, we find UDAL provides quite healthy increases in performance over CReST and is only 0.7 absolute points worse than a fully supervised baseline with access to 100% of the labels.

4.3.4 When γ_u is unknown.

As noted in Section 2.2, the *de facto* assumption generally made in semi-supervised imbalanced learning is that the unlabeled data is generated from the same underlying distribution as the labeled data, *i.e*., the imbalance factors of the labeled (γ_l) and unlabeled (γ_u) sets are the same as γ.

While this assumption has statistically sound properties under the idea that we label a sufficiently large set of unlabeled data in order to produce the labeled set, it can be interesting to consider how UDAL would perform without this assumption *i.e*., $\gamma_l \neq \gamma_u$. As outlined in Section 3.3, we estimate the class distribution of unlabeled set using a small amount of labeled data following [18] and train a model with UDAL loss. We report the performance evaluated with known (γ_u) and estimated (γ'_u) in Table 6. We observe that UDAL is quite robust to the noisy measurements of the distribution given by the estimation procedure – even in the extreme case that the unlabeled data is uniform. In all cases, we see that UDAL outperforms DARP [18].

				Accuracy	
Method	γ_l	γ_u	γ'_u	γ_u	γ'_u
DARP [18]	100	1	1.7	-	85.4
UDAL (ours)	100	1	1.7	89.8	**89.4**
DARP [18]	100	50	69	-	77.3
UDAL (ours)	100	50	69	83.6	**83.4**
DARP [18]	100	150	82	-	72.9
UDAL (ours)	100	150	82	79.9	**79.2**

Table 6: Evaluating when γ_u is known versus estimated as γ'_u on CIFAR10-LT.

Method	$\alpha_{\min}$	k	Accuracy
Progressive LA + DA	0.0	3	77.9
Progressive LA + DA	0.1	1	77.9
Progressive LA + DA	0.2	2	77.9
Progressive LA + DA	0.1	2	77.5
UDAL (ours)	0.1	2	**80.2**

Table 7: Evaluating the simple combination baseline on CIFAR10-LT at $\gamma = 100$ which falls short of UDAL.

4.4. Ablation studies

We carry out ablation studies for our method using the CIFAR10-LT dataset with $\gamma = 100$ and $\beta = 10\%$. We plot all graphs with respect to the same scale so that they may be readily compared.

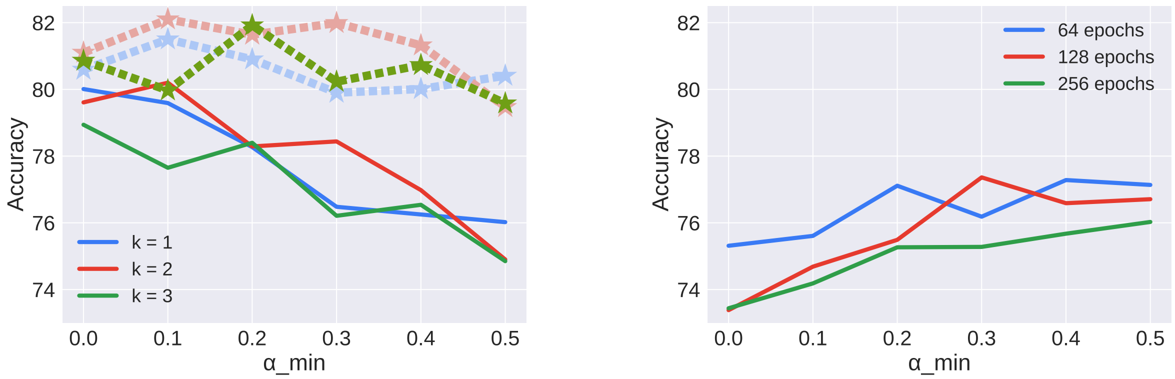
4.4.1 A simpler combination?

Given the introduction of progressive distribution alignment in Section 2.2 and progressive logit adjustment in Equation (9), it might appear tempting to produce a "simple combination" of these by applying the progressive form of distribution alignment to the unsupervised branch and the progressive form of logit adjustment in to the unsupervised branch. This is as opposed to the form of UDAL which applies a similar form of Equation (9) to both branches. For this simple combination, which we denote as "Progressive LA + DA", we explore the best performing hyperparameter settings to understand its performance in Table 7.

We find a large gap between the best settings of this simple combination and UDAL. We hypothesize that while logit adjustment (LA) and distribution alignment (DA) are similar in spirit, they are very different in mechanism:

1. Logit adjustment adjusts the loss computation which directly affects the bias of the learned classifier

2. Distribution alignment adjusts the pseudo-label distribution computation which must first go through *thresholding* to produce pseudo-labels and indirectly affect the classifier



(a) **Analysis of sensitivity to schedule hyperparameters**. We plot the classification performance of our method as a function of schedule parameters k and α_{min}.

(b) **Removing the progressive nature of our alignment** shows a degradation in the performance of UDAL when α_{min} is fixed for the entirety of training (*i.e.*, $k = 0$).

Figure 3: Ablation on design choices for our method. $\star$ denotes with inference time logit adjustment (LA)

UDAL, however, uses a unified approach by applying progressive logit adjustment to *both* branches and therefore avoids this mismatch.

4.4.2 Schedule parameters

Since UDAL uses a form of progressive distribution alignment in a similar way to CReST, it introduces two hyperparameters: k and α_{min} which dictates the extent and rate at which the alignment approaches the uniform distribution. An ablation of these values on CIFAR10-LT is shown in Figure 3a which supports that a quadratic schedule and relatively low α_{min} are optimal.

4.4.3 Is a schedule necessary?

We investigate whether a schedule is necessary for good performance. This is equivalent to a "schedule" with the distribution alignment hyperparameter $k = 0$. We include a variety of settings for α_{min} and the duration of training in Figure 3b to ensure we adequately explore the trends in performance. We observe a significant loss in performance compared to the progressive version of UDAL presented in the paper. We attribute this to the fact that the *supervised branch* is immediately pushed to produce a marginal distribution of pseudo-labels that are more balanced. This, however, goes against empirical evidence [15] that representations are best learned from data sampled according to the data distribution, even if it is imbalanced.

5. Related Work

Semi-supervised learning [10, 21, 20, 4, 3, 27] has recently seen strong advances in performance. This can be attributed to the success of pseudo-labeling [21] combined with consistency of predictions [4, 27] among varying types of augmentations of unlabeled data.

While supervised learning has progressed significantly, a large amount of work has tackled the setting of class imbalanced learning [15, 14, 9, 24, 26, 12, 28, 17]. These range from modifications to the loss formulation [14, 24, 26, 12, 17] to decoupling representation from the classifier [15] and even modifying the optimization process itself [28].

By combining the two previous settings, we consider imbalanced, semi-supervised learning. Although still relatively unexplored, previous attempts [13, 18, 29] to combat imbalance in the semi-supervised setting have been made. Works like DARP [13, 18] modify the loss formulation of a base semi-supervised learner to combat bias within majority classes, while CReST [29] involves a hybrid, generational approach that progressively aligns pseudo-labels predictions as well as augmenting the labeled set with rebalanced, confident pseudo-labels. DASO [25] provides a modification to blend pseuo-label generation, while ABC [22] uses an auxiliary classifier to better balance the learned representation . Compared to these, we believe our formulation is simpler and connects common techniques from the supervised and semi-supervised literature while maintaining comparable performance.

6. Conclusion

We present Unifying Distribution Alignment as a Loss (UDAL) that addresses the issue of class-imbalance within semi-supervised learning. By connecting the ideas of distribution alignment to logit adjustment, we provide a loss that can be applied to both the supervised and unsupervised branches rather than previous disjoint approaches. Our approach incurs no additional training time on top of the underlying semi-supervised learner, achieves improved performance across multiple imbalanced settings and datasets, and scales to larger, more realistic datasets like ImageNet, while requiring only a few lines of code.

References

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] Eric Arazo, Diego Ortego, Paul Albert, Noel E O'Connor, and Kevin McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *International Joint Conference on Neural Networks (IJCNN)*, 2020.

[3] David Berthelot, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. ReMixMatch: Semi-supervised learning with distribution matching and augmentation anchoring. In *ICLR*, 2020.

[4] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. MixMatch: A holistic approach to semi-supervised learning. In *NeurIPS*, 2019.

[5] 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.

[6] Jonathon Byrd and Zachary Lipton. What is the effect of importance weighting in deep learning? In *ICML*, 2019.

[7] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in neural information processing systems*, 32, 2019.

[8] 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.

[9] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *CVPR*, 2019.

[10] Yves Grandvalet, Yoshua Bengio, et al. Semi-supervised learning by entropy minimization. *CAP*, 367:281–296, 2005.

[11] Haibo He and Edwardo A Garcia. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9):1263–1284, 2009.

[12] Youngkyu Hong, Seungju Han, Kwanghee Choi, Seokjun Seo, Beomsu Kim, and Buru Chang. Disentangling label distribution for long-tailed visual recognition. In *CVPR*, 2021.

[13] Minsung Hyun, Jisoo Jeong, and Nojun Kwak. Class-imbalanced semi-supervised learning. *arXiv preprint arXiv:2002.06815*, 2020.

[14] 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 *CVPR*, 2020.

[15] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *ICLR*, 2020.

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

[17] Salman H Khan, Munawar Hayat, Mohammed Bennamoun, Ferdous A Sohel, and Roberto Togneri. Cost-sensitive learning of deep feature representations from imbalanced data. *IEEE transactions on neural networks and learning systems*, 29(8):3573–3587, 2017.

[18] Jaehyung Kim, Youngbum Hur, Sejun Park, Eunho Yang, Sung Ju Hwang, and Jinwoo Shin. Distribution aligning refinery of pseudo-label for imbalanced semi-supervised learning. In *NeurIPS*, 2020.

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *NIPS*, 2012.

[20] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *ICLR*, 2017.

[21] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *ICML Workshop*, 2013.

[22] Hyuck Lee, Seungjae Shin, and Heeyoung Kim. Abc: Auxiliary balanced classifier for class-imbalanced semi-supervised learning. *Advances in Neural Information Processing Systems*, 34, 2021.

[23] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens Van Der Maaten. Exploring the limits of weakly supervised pretraining. In *Proceedings of the European conference on computer vision (ECCV)*, pages 181–196, 2018.

[24] Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. In *ICLR*, 2021.

[25] Youngtaek Oh, Dong-Jin Kim, and In So Kweon. Daso: Distribution-aware semantics-oriented pseudo-label for imbalanced semi-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9786–9796, 2022.

[26] Jiawei Ren, Cunjun Yu, Shunan Sheng, Xiao Ma, Haiyu Zhao, Shuai Yi, and Hongsheng Li. Balanced meta-softmax for long-tailed visual recognition. In *NeurIPS*, 2020.

[27] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. FixMatch: Simplifying semi-supervised learning with consistency and confidence. In *NeurIPS*, 2020.

[28] Kaihua Tang, Jianqiang Huang, and Hanwang Zhang. Long-tailed classification by keeping the good and removing the bad momentum causal effect. In *NeurIPS*, 2020.

[29] Chen Wei, Kihyuk Sohn, Clayton Mellina, Alan Yuille, and Fan Yang. Crest: A class-rebalancing self-training framework for imbalanced semi-supervised learning. In *CVPR*, 2021.

[30] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.