SaR: Self-adaptive Refinement on Pseudo Labels for Multiclass-Imbalanced Semi-supervised Learning

Zhengfeng Lai[1*] Chao Wang[2*] Sen-ching Cheung[3] Chen-Nee Chuah[1]
[1]University of California, Davis [2]Southern University of Science and Technology [3]University of Kentucky
[1]{lzhengfeng, chuah}@ucdavis.edu [2] chaowang.hk@gmail.com [3] sccheung@ieee.org

Abstract

Class-imbalanced datasets can severely deteriorate the performance of semi-supervised learning (SSL). This is due to the confirmation bias especially when the pseudo labels are highly biased towards the majority classes. Traditional resampling or reweighting techniques may not be directly applicable when the unlabeled data distribution is unknown. Inspired by the threshold-moving method that performs well in supervised learning-based binary classification tasks, we provide a simple yet effective scheme to address the multiclass imbalance issue of SSL. This scheme, named SaR, is a Self-adaptive Refinement of soft labels before generating pseudo labels. The pseudo labels generated post-SaR will be less biased, resulting in higher quality data for training the classifier. We show that SaR can consistently improve recent consistency-based SSL algorithms on various image classification problems across different imbalanced ratios. We also show that SaR is robust to the situations where unlabeled data have different distributions as labeled data. Hence, SaR does not rely on the assumptions that unlabeled data share the same distribution as the labeled data.

1. Introduction

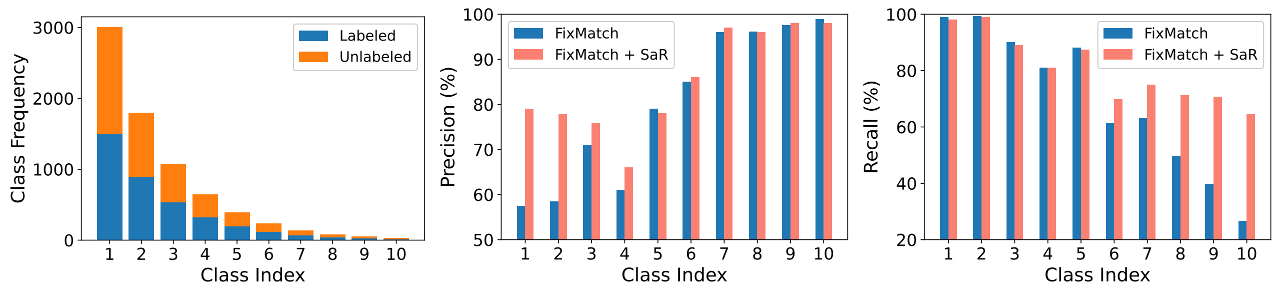
To circumvent the heavy reliance on a well-curated and large-scale labeled dataset, semi-supervised learning (SSL) has been actively studied by leveraging a large amount of unlabeled data for training the classifier in addition to a limited labeled set [47]. Recent SSL algorithms have shown promising performance on standard image classification benchmarks by combining both pseudo labeling and consistency regularization to leverage unlabeled data effectively [4, 5, 37]. Specifically, they predict pseudo labels on unlabeled data, then use them to train the model with the consistency loss as a regularization item. However, these SSL algorithms [4, 5, 37, 41] assume that the class distributions are balanced for both the labeled and unlabeled sets.

One challenge for the applicability of SSL in real-world scenarios is that their performance can suffer from the potential imbalance issue among the unlabeled data [8, 22, 24, 34]. Real-world datasets typically have skewed distributions, here we take "long-tailed" distribution [17, 27, 28, 42] as one example: as shown in Figure 1(a), the *majority classes* have the most of the data points while the *minority classes* have very few samples. In supervised learning (SL), recent approaches including re-sampling [11, 19, 20, 48] and re-weighting [9, 10, 14, 39] based techniques have shown their effectiveness. However, they are designed for SL and relying on the distribution information of the entire dataset. Hence they may not be directly applicable to SSL settings where the majority samples are unlabeled. A few recent works [22, 44] have tried to alleviate this issue under SSL settings, but they require either complicated sampling-based techniques or distribution alignment of the pseudo labels or a combination of both. These processes can be computationally expensive, resulting in longer training time. Besides, they rely on the assumptions about the distribution of the unlabeled set [26, 44], which may not hold true in practice. Therefore, we seek a computation-efficient solution to address the multiclass-imbalanced issue under SSL settings *without* any reliance on the knowledge about the distribution of unlabeled sets.

To tackle this issue, we first take a deeper look into the key component of these recent SSL algorithms [5, 31, 37, 41]: one-hot pseudo labels. Figure 1(b), (c) show the precision and recall of the pseudo labels for the unlabeled data after 200 epochs of training with FixMatch [37]: the precision of majority classes are relatively lower while the recall of minority classes are underestimated [44]. The model is highly biased in this example, which deteriorates the quality of pseudo labels and further hurts the performance: incorrect pseudo-label assignments may lead to misclassifications in subsequent iterations, resulting in a vicious cycle of self-reinforcing errors [24, 38].

*Equal contributions.

This work was supported by the NSF HDR:TRIPODS grant CCF-1934568.



(a) (b) (c)

Figure 1. Experimental results on CIFAR-10 under the imbalanced ratio $\gamma = 100$. (a) Class distribution of the labeled and unlabeled data. (b) The precision of pseudo labels generated from the unlabeled set. (c) The recall of pseudo labels generated from the unlabeled set.

Therefore, it is crucial to improve the quality of pseudo labels so that more of them can be predicted for representing the minority classes instead of biasing towards the majority classes. This motivates us to take a step backward and observe the pseudo-labeling process: before generating the one-hot pseudo labels, the soft labels that are represented for the predicted probability for different classes are already biased. Inspired by threshold-moving based methods in SL [13, 49], we propose Self-adaptive Refinement (SaR) to refine the soft labels before generating pseudo labels so that the minority classes can be easier to be assigned with the correct pseudo labels. Moreover, following [47], we define the biased degree by estimating the learning difficulty and propose *Mitigating Vector* to achieve the same effect of threshold-moving [13, 49].

Contribution. In this work, we propose SaR (Self-adaptive Refinement) to refine the soft labels before generating the one-hot pseudo labels (as shown in Figure 2) to alleviate the confirmation bias so that both the recall of minority classes and the precision of majority classes can be improved. We demonstrate the compatibility of SaR with several recently proposed pseudo-label based SSL algorithms. SaR only introduces less than 5% additional running time compared to the original SSL but it improves both ReMixMatch [4] and FixMatch [37] by up to 28.7% and 32.5% on the geometric mean (GM) score of class-wise accuracies [6] respectively. We show that the proposed scheme outperforms the recent state-of-the-art imbalanced algorithms designed for SSL on all the datasets and conditions tested: unlabeled data can be of both either similar or different distributions compared to the labeled data; hold-out test sets can also have different distributions.

2. Related works

Semi-supervised learning. Semi-supervised learning (SSL) aims to leverage the unlabeled data to improve the performance of supervised learning. One type of SSL is pseudo labeling based algorithms [25]: use the model itself to generate predictive probability on unlabeled data, which will be converted into a "soft" pseudo-label by temperature scaling or a "hard" pseudo-label by a manual threshold [4, 5, 37, 45]. To promote the prediction consistency on unlabeled data, consistency regularization [3, 35] can be applied on different perturbed versions of the same image. The recent state-of-the-art SSL algorithms, FixMatch [37] and ReMixMatch [4], combine these two techniques and achieve promising results. However, all of the aforementioned algorithms assume that the labeled and unlabeled data are of the same uniform distribution. In practice, they may face performance degradation by the imbalanced scenarios since the pseudo labels can be significantly biased [8, 22, 34].

Class-imbalanced supervised learning. Supervised learning (SL) on class-imbalanced datasets has been widely studied due to its wide applicability. Recent approaches include re-weighting the loss function [2, 9, 10, 14, 21, 39] and re-sampling the training data [7, 11, 19, 20, 33, 48]. Both of them require a pre-defined distribution to re-balance the training data or the loss function. Another direction is the use of threshold-moving based techniques in cost-sensitive learning in which output thresholds are adjusted to handle both class imbalance and different costs associated with misclassification of different classes [13, 49]. Although the aforementioned methods are effective, they are designed for SL and heavily rely on the knowledge of the distribution of the labeled data. Hence their applicability to SSL is questionable due to unknown distributions of the unlabeled data.

Class-imbalanced semi-supervised learning. Semi-supervised AUC (area under a receiver operating characteristic curve) optimization [16, 36, 46] has been proposed to address the imbalanced issue but these methods are specifically designed for *binary* classification tasks. For the multi-class classification problem, Wei et al. [44] found that the pseudo labels for the minor classes from FixMatch [37] are of high precision and low recall. Based on this observation, they proposed CReST to expand the labeled data by incorporating pseudo-labeled samples from the unlabeled set.

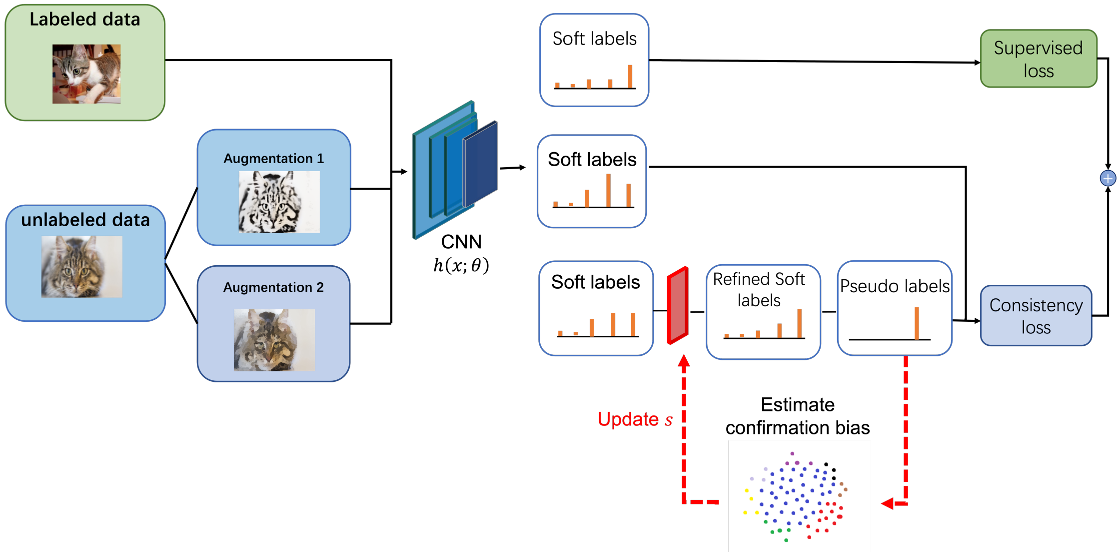


Figure 2. Illustration of the SaR framework.

However, they only consider the situations where unlabeled data have similar distribution as labeled data. Such assumption is also used in [26]. Besides, CReST [44] can introduce up to 20 times of computation cost compared to the original SSL algorithms. DARP [22] proposed a technique called distribution aligning refinery to refine the highly biased pseudo labels. DARP [22] also made an implicit assumption that the confusion matrix of unlabeled data must be the same as that of the labeled data. However, such an assumption about the unlabeled data does not hold in many practical applications such as medical data collected over different subgroups [29, 40]. In this paper, we aim at alleviating the imbalance issue without any pre-defined information on the unlabeled set, which means the unlabeled data can also have totally different distribution as the labeled data.

3. Methodology

3.1. Problem setup

We first set up the problem in class-imbalanced semi-supervised learning (SSL). In an C-class classification task, an input vector and its corresponding one-hot label are denoted as $x \in \mathbb{R}^d$ and $y \in \{0,1\}^C$, respectively. Here d is the dimension of the input. There are a labeled set $\mathcal{D}_1 = \left\{(x_i^{(1)}, y_i^{(1)})\right\}_{i=1}^{m}$ and an unlabeled set $\mathcal{D}_{\mathrm{u}} = \left\{x_j^{(\mathrm{u})}\right\}_{j=1}^{n}$, with $m \ll n$. Let m_k be the number of training labeled instances in $\mathcal{D}_1$ of class k, i.e., $\sum_{k=1}^{C} m_k = m$. Similarly, n_k is for unlabeled instances in $\mathcal{D}_{\mathrm{u}}$ of class k with $\sum_{k=1}^{C} n_k = n$ where n_k is *unknown*. Without loss of generality, we assume $\{m_k\}$ and $\{n_k\}$ are sorted by cardinality in descending order, i.e., $m_1 \geq m_2 \geq \cdots \geq m_C$, and $n_1 \geq n_2 \geq \cdots \geq n_C$. Here the skewed class distribution is considered, i.e., $m_1 \gg m_C$ or $n_1 \gg n_C$. The fraction $\gamma_l = \frac{m_1}{m_C}$ and $\gamma_u = \frac{n_1}{n_C}$ measure the degree of class imbalance in labeled and unlabeled data, respectively.

SSL aims to train a classifier $h(x;\theta) : \mathbb{R}^d \rightarrow [0,1]^C$ using the above training data. Here the k-component $h(x;\theta)_k \in [0,1]$ is the predictive probability for the k-th class given an input x. To utilize $\mathcal{D}_{\mathrm{u}}$ effectively, many state-of-the-art SSL algorithms infer their labels using pseudo labeling schemes denoted by

$$\left\{\hat{y}^{(\mathrm{u})} \in [0,1]^C \;\middle|\; \sum_{k=1}^{C} \hat{y}^{(\mathrm{u})}(k) = 1\right\}_{j=1}^{n}.$$

Pseudo labels are generated based on the classifier $h(x;\theta)$ with a stochastic data augmentation function α. Specifically, one can obtain a one-hot vector as pseudo label by

$$\hat{y}^{(\mathrm{u})}(k) = \delta(h(\alpha(x^{(\mathrm{u})});\theta)_k), k = 1, \cdots C, \qquad (1)$$

where δ is an operator to transform a soft label $h(\alpha(x^{(\mathrm{u})});\theta)$ into a one-hot vector. With these pseudo labels, the training process is to minimize the loss in both labeled and unlabeled data

$$\min_{\theta \in \Theta} \sum_{i=1}^{m} \mathcal{L}(h(x_i^{(1)};\theta), y_i^{(1)}) + \sum_{j=1}^{n} \hat{\mathcal{L}}(h(\hat{\alpha}(x_j^{(\mathrm{u})});\theta), \hat{y}_j^{(\mathrm{u})}), \qquad (2)$$

where $\mathcal{L}$ and $\hat{\mathcal{L}}$ are the per-sample losses (e.g., cross-entropy). The first term works as a supervised loss function for labeled data while the latter one is a regularization term. Note that some SSL algorithms use different distance-based loss function for labeled and unlabeled data. One of the widely used regularization is consistency loss [30, 37, 41]. This type of regularization forces SSL to predict consistency on different views of the same sample. Here $\hat{\alpha}$ refers to another augmentation different from α in (1). For simplicity, in the following content, we will only consider consistency loss as the regularization term.

3.2. Self-adaptive refinement on soft labels

Although traditional reweighting [2, 9, 10, 14, 39] and resampling [11, 19, 20, 48] are popular techniques dealing with imbalanced datasets and show their effectiveness on supervised learning (SL), they cannot be directly applicable to SSL since the majority of samples are unlabeled. Instead, we focus on the threshold-moving method [49] to improve the quality of pseudo labels in class-imbalanced SSL. One drawback of consistency loss-based SSL methods is that the imperfect class distribution is used to generate pseudo labels and the over-reliance on pseudo labels makes it difficult to correctly update the class distribution. It motivates us to take one step back and focus on refining the soft labels and adjusting the thresholds for pseudo label generation. Threshold-moving is a classical method for the binary-classification tasks. It has been extended to multi-class problems for supervised learning. For example, Zhou et al. [49] adjusted the output of neural network to move the output threshold toward classes with higher misclassification costs. This is very different from re-sampling [7, 19, 20, 33, 48] as it relies on the manipulation of the outputs of the classifier instead of re-balancing training data. Threshold-moving is more applicable to SSL than traditional re-sampling or re-weighting schemes since it does not rely on the pre-defined distribution of the unlabeled data.

Here we refine the pseudo labels (1) as

$$\hat{y}^{(\mathrm{u})}(k) = \delta(s_k \cdot h(\alpha(x^{(\mathrm{u})}); \theta)_k), k = 1, \cdots, C. \tag{3}$$

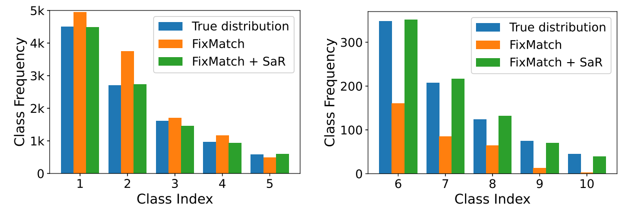
where $s = [s_k, k = 1, \cdots, C]$ is a positive scaling vector to mitigate the confirmation bias [1]. We refer to this vector as *Mitigating Vector* and will describe how it is calculated in the next subsection. Note that some SSL algorithms directly use soft labels as pseudo labels. In that circumstance, (3) becomes

$$\hat{y}^{(\mathrm{u})}(k) = \frac{s_k \cdot h(\alpha(x^{(\mathrm{u})}); \theta)_k}{\sum_{j=1}^{C} s_j \cdot h(\alpha(x^{(\mathrm{u})}); \theta)_j}, \tag{4}$$

with the normalization.

3.3. Mitigating Vector

Inspired by the cost-sensitive classification [49], we turn our attention to the learning effect of unlabeled data during the training process. Consider a long-tailed version of the CIFAR-10 with the imbalance ratio (the ratio between



(a) Majority Classes (b) Minority Classes

Figure 3. Confirmation bias on CIFAR10-LT under imbalanced ratio $\gamma = 100$: (a) the estimated distribution of majority classes; (b) the estimated distribution of minority classes.

the number of samples of the most frequent class and that of the least frequent class) as $\gamma := \gamma_l = \gamma_u = 100$, as shown in Figure 3 (blue bars). In the same figure, we see that pseudo labels generated from FixMatch [37] are highly biased towards the majority classes. The original imbalance ratio can be further accentuated from 100 to 1000 [22]. To address this issue, we propose Self-adaptive Refinement (SaR) to take the confirmation bias of each class in SSL into account. Specifically, we adjust the soft labels so as to rebalance the contributions of samples from both the majority and the minority classes. To achieve this, one can make the value of s_k proportional to the biased degree. In [47], the degree of bias is estimated by counting the number of samples $\hat{n}_k$ whose predictions fall into class k and above the threshold, i.e.,

$$\hat{n}_k = \sum_{j=1}^{n} \mathbf{1}(h(\alpha(x_j^{(\mathrm{u})}); \theta)_k > \tau) \cdot \mathbf{1}(\arg\max(\hat{y}_j^{(\mathrm{u})}) = k), \tag{5}$$

where τ is pre-defined threshold and $\mathbf{1}(\cdot)$ is the indicated function. In other words, the class that has more samples with prediction confidence exceeds τ is considered to exhibit a lower bias [47]. Hence we can design our Mitigating Vector as follows:

$$s_k = b(\hat{n}_k), k = 1, \cdots, C. \tag{6}$$

Here $b(\cdot)$ is a monotone decreasing function. Unlike the FlexMatch [47], we fix the threshold τ during the training and focus on refining the soft labels. Besides, blindly matching the distribution of the pseudo labels to the true distribution could bias the classifier [22]. We design a mitigating vector to smooth the distribution of the pseudo labels by balancing the needs of representing the true distribution and emphasizing the minority classes. Specifically, we use the function b, which is inspired by the effective samples [14] proposed for SL, i.e.,

$$b(\hat{n}_k) = (1 - \beta)/(1 - \beta^{\hat{n}_k}). \tag{7}$$

The degree of flattening is determined by the value of $\beta \in (0, 1)$. Here $\beta = 0$ nullifies the impact on the sample counts $\hat{n}_k$ making all s_k's equal to 1. When β goes

Algorithm 1 The SaR Framework

Data: Labeled data $\{(x_i^{(1)}, y_i^{(1)})\}_{i=1}^m$, unlabeled data $\{x_j^{(\mathrm{u})}\}_{j=1}^n$, number of classes C, learning rate η, max epoch T
Initialize model parameter θ, Mitigating Vector s
while $t < T$ **do**
  for `1, ` $\ldots K$ **do**
    Sample batches label data $\{x^{(1)}, y^{(1)}\}$
    Sample batches unlabeled data $\{x^{(\mathrm{u})}\}$
    Refine the soft labels from unlabeled data with s
    Compute pseudo labels $p(x^{(\mathrm{u})}; \theta)$ and save them
    Update θ by stochastic gradient decent with fixed s
  end for
  Estimate $(\hat{n}_1, \ldots, \hat{n}_C)$ based on (5)
  Update $s_k = b(\hat{n}_k), k = 1, \cdots, C$
end while
Return: θ

to 1, $b(\hat{n}_k)$ returns a larger value for a smaller $\hat{n}_k$, thereby putting more emphasis on the minority classes. We select β as 0.999 and fix it as suggested by [14]. After we refine the soft labels, we will generate a new set of pseudo labels that alleviate the confirmation bias: as shown in Figure 3(b), the estimated distribution of the minority classes are closer to the true distribution. The entire SaR framework is summarized in Figure 2 and Algorithm 1.

4. Experiments

In this section, we evaluate our proposed SaR across CIFAR-10, CIFAR-100 [23], and STL-10 [12] under various class-imbalanced situations, e.g., when 1) labeled and unlabeled data share similar distribution; and 2) unlabeled data are of different distribution from labeled data. We use *balanced accuracy* (bACC) [15, 43] and *geometric mean scores* (GM) [6] as the performance measuring metrics. To have a fair comparison, all SSL algorithms use the same encoder (Wide ResNet-28-2 [32]) and batch size of 64. All of the following results are collected via running three random trials for each setting. Each random trial also randomizes the classes in generating the long-tailed dataset.

4.1. CIFAR10-LT

Setup. We first consider the situations where unlabeled data share similar distribution as labeled data [44]. Based on this assumption, we construct a "synthetically long-tailed" variant of CIFAR-10 [23], denoted as CIFAR10-LT. In the training set, both labeled and unlabeled images are randomly sampled based on the pre-defined imbalance ratios: γ_l denotes for the labeled set and γ_u denoted for the unlabeled set. We follow the same settings of DARP [22] where $m_1 = 1500$ and $n_1 = 3000$, then we have $m_k = m_1 \cdot \gamma_l^{\epsilon_k}$ and $n_k = n_1 \cdot \gamma_l^{\epsilon_k}$, where $\epsilon_k = -\frac{k-1}{C-1}$. We vary γ to construct various datasets of different imbalanced degree. The hold-out test set remains balanced following similar settings in previous studies [4, 9, 18, 20, 22, 32, 44].

Main results under $\gamma_l = \gamma_u$. When the unlabeled data have similar distribution with the labeled data, we select the following baselines for the comparison: 1) Supervised learning using Wide ResNet-28-2 [32] on the labeled data without re-balancing algorithms; 2) Re-sampling [18] the labeled data based on its distribution; 3) Label distribution-aware margin (LDAM-DRW) [9]; 4) classifier Re-Training cRT [20]. To evaluate the efficacy, we apply SaR to two recent state-of-the-art SSL algorithms: ReMixMatch [4] and FixMatch [37].

We compare SaR with recent class-imbalanced techniques designed for SSL, DARP [22] and CReST [44], which outperform other methods. Hence we do not include other methods into our comparison. Note that DARP [22] used two measuring metrics (bACC and GM) while CReST [44] only used one metric (bACC). Besides, we do not consider applying CReST [44] on ReMixMatch [4] because the computational overhead is too large (20x original run time), which renders it impractical. The main results are summarized in Table 1. We show that SaR can consistently improve both ReMixMatch [4] and FixMatch [37] with at least 3% absolute gain on bACC and at least 4.4% on GM for all settings. The absolute gain on GM can even be up to 9% with FixMatch [37]. Moreover, SaR outperforms the recent imbalance technique (DARP [22]) on both bACC and GM for all settings, and outperforms CReST [44] on bACC. Specifically, SaR achieves up to 4.6% increase on GM and 6.1% increase on bACC compared to DARP [22].

Main results under $\gamma_l \neq \gamma_u$. To make SSL more applicable to various settings, we also consider the situations where the distributions of labeled and unlabeled set do not match each other. We set γ_l as 100 but vary γ_u from 1, 50, to 150. We even consider the situation where unlabeled data are not only imbalanced but the distribution is the reverse of that of labeled data ($\gamma_u = 100$ reversed). Under this setting, since CReST [44] requires that labeled and unlabeled data share similar distribution, which does not hold true in this setting, we exclude it for our comparison. Our main results are summarized in Table 2. Due to the unknown distribution of the unlabeled data, the algorithms designed for SL such as re-sampling [18], LDAM-DRW [9], cRT [20] will only re-balance the labeled data.

From Table 2, when the unlabeled data have different distribution from the labeled data, we find SaR improves both ReMixMatch [4] and FixMatch [37] even more significantly. The absolute gain is up to 5% for ReMixMatch [4] and 17% for FixMatch [37]. When compared to DARP [22], SaR also achieves noticeable improvements for all settings,

Table 1. Comparison of classification performance on CIFAR10-LT under the situations where the unlabeled data are of similar distribution to the labeled data $\gamma = \gamma_l = \gamma_u$ (hold-out test set is balanced). The evaluation criterion is bACC/GM.

Algorithm	SSL	RB	CIFAR10-LT $\gamma = 50$	$\gamma = 100$	$\gamma = 150$
Wide ResNet-28-2 [32]	-	-	65.2±0.05 / 61.1±0.09	58.8±0.13 / 51.0±0.11	55.6±0.43 / 44.0±0.98
Re-sampling [18]	-	✓	64.3±0.48 / 60.6±0.67	55.8±0.47 / 45.1±0.30	52.2±0.05 / 38.2±1.49
LDAM-DRW [9]	-	✓	68.9±0.07 / 67.0±0.08	62.8±0.17 / 58.9±0.60	57.9±0.20 / 50.4±0.30
cRT [20]	-	✓	67.8±0.13 / 66.3±0.15	63.2±0.45 / 59.9±0.40	59.3±0.10 / 54.6±0.72
ReMixMatch [4]	✓	-	81.5±0.26 / 80.2±0.32	73.8±0.38 / 69.5±0.84	69.9±0.47 / 62.5±0.35
ReMixMatch + DARP [22]	✓	-	82.1±0.14 / 80.8±0.09	75.8±0.09 / 72.6±0.24	71.0±0.27 / 64.5±0.68
ReMixMatch + SaR	✓	-	**85.1**±0.25 / **84.6**±0.37	**77.2**±0.71 / **75.5**±0.30	**72.9**±0.45 / **69.1**±0.23
FixMatch [37]	✓	-	79.2±0.33 / 77.8±0.36	71.5±0.72 / 66.8±1.51	68.4±0.15 / 59.9±0.43
FixMatch + DARP [22]	✓	-	81.8±0.24 / 80.9±0.28	75.5±0.05 / 73.0±0.09	70.4±0.25 / 64.9±0.17
FixMatch + CReST [44]	✓	✓	83.0±0.39 / 81.5±0.17	75.7±0.38 / 72.7±0.85	70.8±0.25 / 64.5±0.31
FixMatch + SaR	✓	-	**83.9**±0.09 / **82.9**±0.15	**77.6**±0.42 / **75.9**±0.76	**71.5**±0.23 / **66.9**±0.25

Table 2. Comparison of classification performance on CIFAR10-LT under the situations where the unlabeled data are of different distribution from the labeled data $\gamma_u \neq \gamma_l$ (hold-out test set is balanced). The evaluation criterion is bACC/GM.

Algorithm	SSL	RB	CIFAR10-LT ($\gamma_l = 100$) $\gamma_u = 1$	$\gamma_u = 50$	$\gamma_u = 150$	$\gamma_u = 100$ (reversed)
Wide ResNet-28-2 [32]	-	-	58.8±0.13 / 51.0±0.11	58.8±0.13 / 51.0±0.11	58.8±0.13 / 51.0±0.11	58.8±0.13 / 51.0±0.11
Re-sampling [18]	-	✓	55.8±0.47 / 45.1±0.30	55.8±0.47 / 45.1±0.30	55.8±0.47 / 45.1±0.30	55.8±0.47 / 45.1±0.30
LDAM-DRW [9]	-	✓	62.8±0.17 / 58.9±0.60	62.8±0.17 / 58.9±0.60	62.8±0.17 / 58.9±0.60	62.8±0.17 / 58.9±0.60
cRT [20]	-	✓	63.2±0.45 / 59.9±0.40	63.2±0.45 / 59.9±0.40	63.2±0.45 / 59.9±0.40	63.2±0.45 / 59.9±0.40
ReMixMatch [4]	✓	-	48.3±0.14 / 19.5±0.85	75.1±0.43 / 71.9±0.77	72.5±0.10 / 68.2±0.32	49.0±0.55 / 17.1±1.48
ReMixMatch*	✓	-	85.0±1.35 / 84.3±1.55	77.0±0.12 / 74.7±0.04	72.8±0.10 / 68.8±0.21	75.3±0.03 / 72.3±0.04
ReMixMatch* + DARP [22]	✓	-	89.7±0.15 / 89.4±0.17	77.4±0.22 / 75.0±0.25	73.2±0.11 / 69.2±0.31	80.1±0.11 / 78.5±0.17
ReMixMatch* + SaR	✓	-	**90.2**±0.21 / **89.9**±0.34	**80.1**±0.15 / **78.9**±0.27	**75.1**±0.56 / **73.5**±0.33	**85.4**±0.09 / **83.1**±0.18
FixMatch [37]	✓	-	68.9±1.95 / 42.8±8.11	73.9±0.25 / 70.5±0.52	69.6±0.60 / 62.6±1.11	65.5±0.05 / 26.0±0.44
FixMatch + DARP [22]	✓	-	85.4±0.55 / 85.0±0.65	77.3±0.17 / 75.5±0.21	72.9±0.24 / 69.5±0.18	74.9±0.51 / 72.3±1.13
FixMatch + SaR	✓	-	**85.9**±0.68 / **85.3**±0.53	**80.7**±0.15 / **79.7**±0.20	**75.9**±0.11 / **73.7**±0.27	**78.3**±0.34 / **76.1**±0.21

especially when the unlabeled data have reversed distribution as the labeled data: the absolute improvement can be up to 5.3%. More results where the labeled data are of different imbalanced ratios can be seen in Appendix B.1. Hence we can conclude that SaR has the potential to widen the applicability of SSL in various real-world settings where the distribution of unlabeled data is unknown.

4.2. CIFAR100-LT and STL-10

Setup. To make a more comprehensive comparison, we also experiment with CIFAR-100 [23] and STL-10 [12]. We follow the similar methodology described in Section 4.1 to construct a "synthetically long-tailed" CIFAR-100 denoted as CIFAR100-LT, where $m_1 = 150$ and $n_1 = 300$. For STL-10, we set m_1 as 450 to construct an imbalanced labeled set and add all unlabeled samples into the unlabeled set since STL-10 does not provide annotations for them. Hence STL-10 can be considered as a real-world example.

Main results. We summarize The main results of CIFAR100-LT and STL-10 in Table 3. The results are similar to Section 4.1 in that SaR outperforms original SSL and DARP consistently in all settings. We show that the improvement of SaR on the real-world example, STL-10, is significant. Specifically, SaR can achieve up to 28.7% of absolute gain on bACC and 32.5% on GM with ReMixMatch [4]; 11.2% of absolute gain on bACC and 20.1% on

Table 3. Comparison of classification performance on CIFAR100-LT under the situations where unlabeled data share similar distribution to the labeled data, and STL-10 under the situations where the distribution of unlabeled data is purely unknown. Hold-out test set is balanced. The evaluation criterion is bACC/GM.

Algorithm	CIFAR100-LT ($\gamma = \gamma_l = \gamma_u$)		STL-10	
	$\gamma = 10$	$\gamma = 20$	$\gamma_l = 10$	$\gamma_l = 20$
ReMixMatch [4]	59.2$\pm$0.03 / 52.1$\pm$0.13	53.5$\pm$0.03 / 42.3$\pm$0.13	67.8$\pm$0.45 / 61.1$\pm$0.92	60.1$\pm$1.18 / 44.9$\pm$1.52
ReMixMatch* + DARP [22]	59.8$\pm$0.20 / 52.9$\pm$0.41	54.4$\pm$0.07 / 44.2$\pm$0.07	79.4$\pm$0.07 / 78.2$\pm$0.10	70.9$\pm$0.44 / 67.0$\pm$1.62
ReMixMatch* + SaR	**62.1**$\pm$0.05 / **57.2**$\pm$0.24	**55.8**$\pm$0.69 / **46.7**$\pm$0.56	**83.1**$\pm$0.21 / **82.5**$\pm$0.31	**78.8**$\pm$0.15 / **77.6**$\pm$0.11
FixMatch [37]	60.1$\pm$0.05 / 54.4$\pm$0.11	54.0$\pm$0.04 / 44.4$\pm$0.17	72.9$\pm$0.09 / 69.6$\pm$0.01	63.4$\pm$0.21 / 52.6$\pm$0.09
FixMatch + DARP [22]	61.1$\pm$0.23 / 56.4$\pm$0.28	54.9$\pm$0.05 / 46.4$\pm$0.41	77.8$\pm$0.33 / 76.5$\pm$0.40	69.9$\pm$1.77 / 65.4$\pm$3.07
FixMatch + SaR	**62.5**$\pm$0.15 / **59.4**$\pm$0.37	**55.7**$\pm$0.13 / **47.1**$\pm$0.05	**79.4**$\pm$0.25 / **78.4**$\pm$0.21	**74.6**$\pm$0.26 / **72.7**$\pm$0.36

Table 4. Comparison of classification performance on CIFAR10-LT under $\gamma = \gamma_l = \gamma_u$ (hold-out test set is of reversed distribution). The evaluation criterion is bACC/GM. The best results are in bold.

Algorithm	CIFAR10-LT		
	$\gamma = 50$	$\gamma = 100$	$\gamma = 150$
ReMixMatch [4]	71.0$\pm$0.55 / 83.5$\pm$0.29	54.7$\pm$0.51 / 74.4$\pm$0.47	41.5$\pm$1.69 / 66.4$\pm$1.22
ReMixMatch + DARP [22]	66.9$\pm$0.75 / 80.5$\pm$0.46	49.7$\pm$1.55 / 70.5$\pm$0.90	35.8$\pm$1.81 / 60.9$\pm$2.42
ReMixMatch + SaR	**85.7**$\pm$0.34 / **85.5**$\pm$0.56	**78.5**$\pm$0.78 / **77.2**$\pm$0.37	**70.4**$\pm$0.97 / **67.8**$\pm$0.71
FixMatch [37]	70.5$\pm$0.26 / 82.2$\pm$0.31	51.0$\pm$1.65 / 71.5$\pm$1.24	38.5$\pm$1.15 / 63.4$\pm$0.31
FixMatch + DARP [22]	72.2$\pm$0.62 / 82.8$\pm$0.17	57.6$\pm$0.36 / 74.8$\pm$0.48	46.5$\pm$1.26 / 68.1$\pm$0.10
FixMatch + SaR	**80.3**$\pm$0.49 / **85.6**$\pm$0.61	**65.7**$\pm$2.12 / **78.2**$\pm$1.09	**55.6**$\pm$1.57 / **70.4**$\pm$1.35

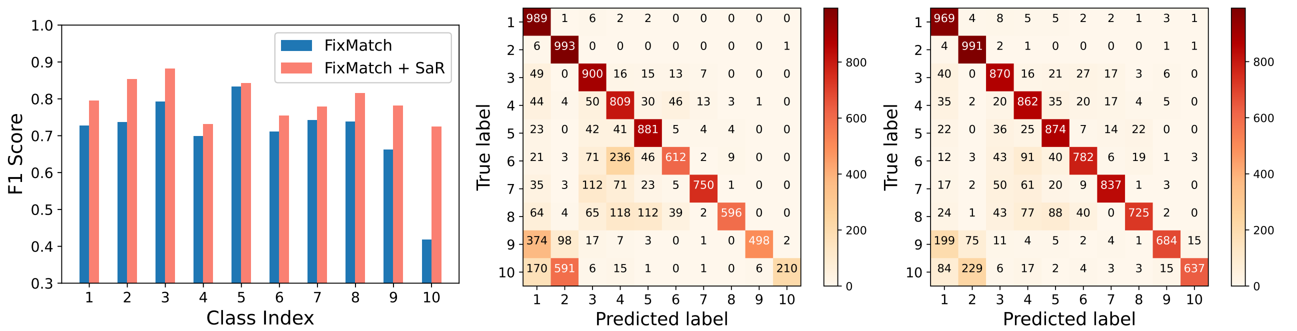
GM with FixMatch [37]. Even compared to DARP [22], SaR can still achieve 7.9% absolute gain on bACC and 10.6% on GM with ReMixMatch [4]; 4.7% absolute gain on bACC and 7.3% on GM with FixMatch [37] when the imbalanced ratio is 20.

4.3. Empirical analysis on SaR

Stress-test. Both DARP [22] and CReST [44] only consider the situation where the hold-out test sets are balanced. To comprehensively study the effectiveness of these algorithms in real-world applications, we must consider the cases where the hold-out test sets can also be imbalanced, or may even follow a different distribution compared to the training set. We first evaluate all aforementioned methods on a hold-out test set that shares similar distribution as the training set. We find all of them perform very well as expected. To stress-test these algorithms, we construct another hold-out test that has a reversed distribution compared to the training set. In other words, the distribution of testing set is totally different from that of our training set.

Table 4 summarizes the results on CIFAR10-LT when labeled and unlabeled training data share similar distribution. We find that DARP [22] would even hurt the performance of ReMixMatch [4] severely: with up to 5.7% absolute decrease on bACC and 5.9% on GM. Another surprising finding is that SaR achieves impressive improvement under this setting. SaR improves original ReMixMatch [4] with up to 24.7% of absolute gain on bACC. Meanwhile, it improves original FixMatch [37] with up to 17.1% absolute gain on bACC and 7% on GM, especially when the imbalanced degree is more severe (e.g. $\gamma = 150$). Compared to DARP [22], SaR has shown significant improvement: up to 34.8% absolute gain on bACC with ReMixMatch [4] and 9.1% with FixMatch [37]. This denotes that SaR has the potential to perform more robust under different conditions of various imbalanced settings in the real-world scenarios.

Per-class performance. Due to the surprising improvement in our stress-test, we compare F1-score for each class to investigate the source of the absolute gain. From Figure 4, we find that SaR improves all classes on the F1-score. The improvements on the minority classes (index as 8, 9, 10) can be significant. As SaR is to refine soft labels before generating pseudo labels, the pseudo labels are less biased. Thus the performance for each class can be improved. Figure 4 shows the comparison of the confu-



(a) (b) (c)

Figure 4. Per-class performance on the balanced test set of CIFAR-10 under the imbalanced ratio $\gamma = 100$. (a) F1-score for each class. (b) Confusion matrix of original FixMatch [37]. (c) Confusion matrix of FixMatch [37] with SaR.

sion matrix from original FixMatch [37] and SaR. We find that SaR improves the precision for the majority classes and the recall for the minority classes without much penalty on other metrics. This explains again why SaR achieves the surprising improvement in our stress-test (Table 4) where the hold-out test set is of a reversed distribution compared to the training set.

Computation complexity. Besides the promising performance of SaR, another advantage of SaR is that it introduces trivial additional computation to the original SSL algorithms during the training process. The additional running time incurred by DARP due to the pseudo-label alignment can be up to 20% of that of the original SSL algorithms [22]. For CReST [44], as it iteratively samples the training data and re-initializes the classifier's network, the additional running time can be an order of magnitude longer than the original SSL algorithms. However, in SaR, as we neither have any resampling actions nor alignment optimization, the additional training time constitutes only 5% of the original SSL training as we simply refine the soft labels. After the training process, the inference complexity remains the same as the classifier is fixed. Hence, we conclude that SaR is a simple yet effective approach to deal with the imbalanced issue under SSL settings as it improves the performance of SSL without any significant overhead.

Table 5. Comparison of bACC on CIFAR10-LT under $\gamma_l = \gamma_u = 100$ under different ratios (η) of labeled data.

	CIFAR10-LT		
Labeled Ratio	$\eta = 1\%$	$\eta = 10\%$	$\eta = 20\%$
FixMatch [37]	40.0$_{\pm 0.59}$	66.1$_{\pm 0.59}$	70.2$_{\pm 0.33}$
FixMatch + DARP [22]	43.7$_{\pm 1.95}$	68.5$_{\pm 0.98}$	72.5$_{\pm 0.89}$
FixMatch + CReST [44]	46.9$_{\pm 1.98}$	68.1$_{\pm 1.20}$	72.4$_{\pm 1.46}$
FixMatch + SaR	**52.5**$_{\pm 1.23}$	**70.1**$_{\pm 0.35}$	**74.7**$_{\pm 0.47}$

Different percentages of labeled data. We also test SaR on the situations where the labeled data are more rare (e.g., 1%). The results are summarized in Table 5: η denotes the percentage of labeled data in the training set. We vary η from 1%, 10%, to 20%. We set $\gamma_l = \gamma_u = 100$ so that both DARP [22] and CReST [44] can be applied here for our comparison study even though SaR does not require the distributions to be the same. We find SaR can consistently achieve better performance across all settings with different amounts of labeled data.

5. Discussion & Limitation

In this work, we propose Self-adaptive Refinement (SaR) to refine the soft labels before generating pseudo labels for mutliclass imbalanced semi-supervised learning. We consider different imbalanced situations where unlabeled data can have both similar or different distribution as the labeled data. We also conduct stress-tests where the hold-out test sets have different distribution compared to the training set. SaR consistently improves both FixMatch [37] and ReMixMatch [4]. It also outperforms recent state-of-the-art SSL imbalanced algorithms on both performance and computation cost. We conclude that SaR has the potential to achieve stable performance on various real-world conditions. The main advantages of SaR can be summarized as: 1) it is compatible with consistency-loss based SSL algorithms; 2) it does not rely on the pre-defined distribution information of the unlabeled data; 3) it only introduces trivial additional running time compared to the original SSL.

However, the limitation of this work is that we still assume the unlabeled data have the same classes as the labeled data. There are also more severely class-imbalanced scenarios that new classes appear in the unlabeled data. The performance of SaR on such situations is still under study. On the other hand, we only studied one Mitigating Vector inspired by [14]. We will study more mitigating schemes systematically in the future work.

References

[1] Eric Arazo, Diego Ortego, Paul Albert, Noel E O'Connor, and Kevin McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2020. 4

[2] Yuri Sousa Aurelio, Gustavo Matheus de Almeida, Cristiano Leite de Castro, and Antonio Padua Braga. Learning from imbalanced data sets with weighted cross-entropy function. *Neural Processing Letters*, 50(2):1937–1949, 2019. 2, 4

[3] Philip Bachman, Ouais Alsharif, and Doina Precup. Learning with pseudo-ensembles. *Advances in neural information processing systems*, 27:3365–3373, 2014. 2

[4] David Berthelot, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. In *International Conference on Learning Representations (ICLR)*, 2020. 1, 2, 5, 6, 7, 8

[5] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 1, 2

[6] Paula Branco, Luís Torgo, and Rita P Ribeiro. A survey of predictive modeling on imbalanced domains. *ACM Computing Surveys (CSUR)*, 49(2):31, 2016. 2, 5

[7] 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. 2, 4

[8] Saul Calderon-Ramirez, Armaghan Moemeni, David Elizondo, Simon Colreavy-Donnelly, Luis Fernando Chavarria-Estrada, Miguel A Molina-Cabello, et al. Correcting data imbalance for semi-supervised covid-19 detection using x-ray chest images. *arXiv preprint arXiv:2008.08496*, 2020. 1, 2

[9] 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*, 32:1567–1578, 2019. 1, 2, 4, 5, 6

[10] 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 (NeurIPS)*, volume 32, 2019. 1, 2, 4

[11] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. SMOTE: synthetic minority oversampling technique. *Journal of Artificial Intelligence Research*, 2002. 1, 2, 4

[12] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics (AISTATS)*, 2011. 5, 6

[13] Guillem Collell, Drazen Prelec, and Kaustubh R Patil. A simple plug-in bagging ensemble based on threshold-moving for classifying binary and multiclass imbalanced data. *Neurocomputing*, 275:330–340, 2018. 2

[14] 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/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2, 4, 5, 8

[15] 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 (CVPR)*, 2016. 5

[16] Tomoharu Iwata, Akinori Fujino, and Naonori Ueda. Semi-supervised learning for maximizing the partial AUC. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 4239–4246, 2020. 2

[17] 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 (CVPR)*, pages 7610–7619, 2020. 1

[18] Nathalie Japkowicz. The class imbalance problem: Significance and strategies. In *Proceedings of the International Conference on Artificial Intelligence (ICAI)*, 2000. 5, 6

[19] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *International Conference on Learning Representations*, 2019. 1, 2, 4

[20] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *International Conference on Learning Representations (ICLR)*, 2020. 1, 2, 4, 5, 6

[21] 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. 2

[22] 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 *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 14567–14579, 2020. 1, 2, 3, 4, 5, 6, 7, 8

[23] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, Department of Computer Science, University of Toronto, 2009. 5, 6

[24] Zhengfeng Lai, Chao Wang, Luca Cerny Oliveira, Brittany N Dugger, Sen-Ching Cheung, and Chen-Nee Chuah. Joint semi-supervised and active learning for segmentation of gigapixel pathology images with cost-effective labeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 591–600, 2021. 1

[25] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning (ICML)*, 2013. 2

[26] 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. 1, 3

[27] Jialun Liu, Yifan Sun, Chuchu Han, Zhaopeng Dou, and Wenhui Li. Deep representation learning on long-tailed data: A learnable embedding augmentation perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2970–2979, 2020. 1

[28] 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/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2537–2546, 2019. 1

[29] Yassine Marrakchi, Osama Makansi, and Thomas Brox. Fighting class imbalance with contrastive learning. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 466–476. Springer, 2021. 3

[30] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018. 4

[31] Varun Nair, Javier Fuentes Alonso, and Tony Beltramelli. Realmix: Towards realistic semi-supervised deep learning algorithms. *arXiv preprint arXiv:1912.08766*, 2019. 1

[32] Avital Oliver, Augustus Odena, Colin A Raffel, Ekin Dogus Cubuk, and Ian Goodfellow. Realistic evaluation of deep semi-supervised learning algorithms. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 3235–3246, 2018. 5, 6

[33] Samira Pouyanfar, Yudong Tao, Anup Mohan, Haiman Tian, Ahmed S Kaseb, Kent Gauen, Ryan Dailey, Sarah Aghajanzadeh, Yung-Hsiang Lu, Shu-Ching Chen, et al. Dynamic sampling in convolutional neural networks for imbalanced data classification. In *2018 IEEE conference on multimedia information processing and retrieval (MIPR)*, pages 112–117. IEEE, 2018. 2, 4

[34] J Vince Pulido, Shan Guleria, Lubaina Ehsan, Matthew Fasullo, Robert Lippman, Pritesh Mutha, Tilak Shah, Sana Syed, and Donald E Brown. Semi-supervised classification of noisy, gigapixel histology images. In *2020 IEEE 20th International Conference on Bioinformatics and Bioengineering (BIBE)*, pages 563–568. IEEE, 2020. 1, 2

[35] Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016. 2

[36] Tomoya Sakai, Gang Niu, and Masashi Sugiyama. Semi-supervised AUC optimization based on positive-unlabeled learning. *Machine Learning*, 107(4):767–794, 2018. 2

[37] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in Neural Information Processing Systems (NeurIPS)*, 33, 2020. 1, 2, 4, 5, 6, 7, 8

[38] Kai Sheng Tai, Peter Bailis, and Gregory Valiant. Sinkhorn label allocation: Semi-supervised classification via annealed self-training. In *International Conference on Machine Learning (ICML)*, 2021. 1

[39] Jingru Tan, Changbao Wang, Buyu Li, Quanquan Li, Wanli Ouyang, Changqing Yin, and Junjie Yan. Equalization loss for long-tailed object recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11662–11671, 2020. 1, 2, 4

[40] Ziqi Tang, Kangway V Chuang, Charles DeCarli, Lee-Way Jin, Laurel Beckett, Michael J Keiser, and Brittany N Dugger. Interpretable classification of alzheimer's disease pathologies with a convolutional neural network pipeline. *Nature communications*, 10(1):1–14, 2019. 3

[41] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 1, 4

[42] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1

[43] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Learning to model the tail. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 5

[44] Chen Wei, Kihyuk Sohn, Clayton Mellina, Alan Yuille, and Fan Yang. Crest: A class-rebalancing self-training framework for imbalanced semi-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10857–10866, 2021. 1, 2, 3, 5, 6, 7, 8

[45] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10687–10698, 2020. 2

[46] Zheng Xie and Ming Li. Semi-supervised AUC optimization without guessing labels of unlabeled data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. 2

[47] Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 1, 2, 4

[48] Boyan Zhou, Quan Cui, Xiu-Shen Wei, and Zhao-Min Chen. Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9719–9728, 2020. 1, 2, 4

[49] 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. 2, 4