This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Towards Realistic Long-Tailed Semi-Supervised Learning: Consistency Is All You Need

Tong Wei, Kai Gan
School of Computer Science and Engineering, Southeast University, Nanjing 210096, China
{weit, gank}@seu.edu.cn

Abstract

While long-tailed semi-supervised learning (LTSSL) has received tremendous attention in many real-world classification problems, existing LTSSL algorithms typically assume that the class distributions of labeled and unlabeled data are almost identical. Those LTSSL algorithms built upon the assumption can severely suffer when the class distributions of labeled and unlabeled data are mismatched since they utilize biased pseudo-labels from the model. To alleviate this issue, we propose a new simple method that can effectively utilize unlabeled data of unknown class distributions by introducing the adaptive consistency regularizer (ACR). ACR realizes the dynamic refinery of pseudo-labels for various distributions in a unified formula by estimating the true class distribution of unlabeled data. Despite its simplicity, we show that ACR achieves state-of-the-art performance on a variety of standard LTSSL benchmarks, e.g., an averaged 10% absolute increase of test accuracy against existing algorithms when the class distributions of labeled and unlabeled data are mismatched. Even when the class distributions are identical, ACR consistently outperforms many sophisticated LTSSL algorithms. We carry out extensive ablation studies to tease apart the factors that are most important to ACR's success. Source code is available at https://github.com/Gank0078/ACR.

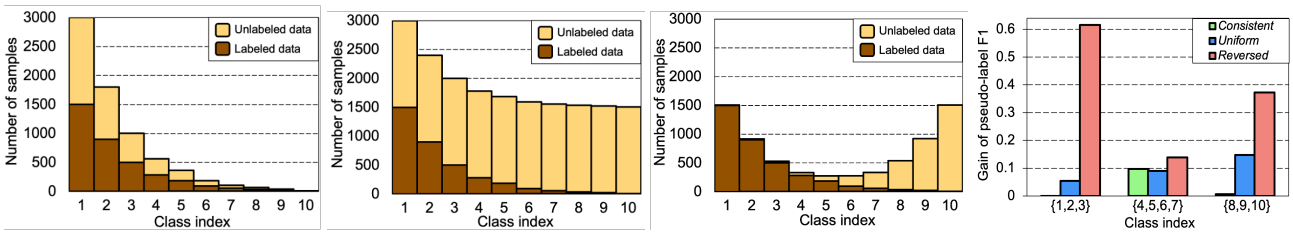
1. Introduction

Semi-supervised learning (SSL) is an effective way of using unlabeled data to improve the generalization of deep neural networks (DNNs) [1, 10, 16] when only a limited amount of labeled data is accessible [3, 23, 29, 31]. The core idea of most SSL algorithms is to generate pseudo-labels for unlabeled data and select confident ones to train models. Recent progress on SSL has revealed promising performance in various tasks, such as image recognition [29] and text categorization [35, 39]. However, most existing SSL algorithms assume the datasets are class-balanced, i.e., each class is associated with an equivalent number of samples in both labeled and unlabeled datasets. In contrast, class distributions in many real-world tasks are long-tailed [6, 19, 33, 38, 43]. It is well known that classifiers trained on long-tailed datasets tend to be biased towards majority classes, leading to low test accuracy on minority classes [20, 37, 44].

To improve the performance, many long-tailed semi-supervised learning (LTSSL) algorithms have been proposed to generate unbiased pseudo-labels. They pursue class-balanced classifiers using techniques including re-sampling [18], re-weighting [17], label smoothing [36], and pseudo-label alignment [14, 34]. These algorithms have shown strong generalization for the minority class by assuming the class distributions of labeled and unlabeled data are almost identical. However, this assumption is frequently violated in real-world applications, for instance, if the labeled and unlabeled data are collected from different tasks. The unlabeled data may have a large class distribution gap from labeled data, and using the erroneous assumption can severely deteriorate the performance [17, 25].

Contribution. This paper studies the under-explored yet practical LTSSL problem, i.e., learning from unlabeled data of unknown class distributions. Notably, we start with three representative types of class distributions of unlabeled data, i.e., *consistent*, *uniform*, and *reversed*, as illustrated in Figures 1a to 1c. We then propose a new simple algorithm to effectively use unlabeled data through the adaptive consistency regularizer (ACR), which is built upon one of the most popular SSL algorithms FixMatch [29]. Concretely, ACR is developed based on two findings: i) to learn a class-balanced classifier, it is helpful to generate pseudo-labels biased appropriately toward the minority class, whereas ii) to learn a better feature extractor, the accuracy of pseudo-labels is critical. Those two findings seem to contradict. We thus present a two-branch network, including a balanced branch and a standard branch, to resolve this con-

Tong Wei is the corresponding author. This research was supported by the National Science Foundation of China (62206049).



(a) *Consistent* class distribution (b) *Uniform* class distribution (c) *Reversed* class distribution (d) F_1 gain of pseudo-labels

Figure 1. (1a to 1c): Three typical types of class distribution of unlabeled data. (1d): F_1 gain due to our method ACR compares to a recent state of the art DASO [25] under three types of class distributions of unlabeled data. We can see that ACR significantly improves the quality of pseudo-labels, showing its great capability of taking advantage of unlabeled data to alleviate the class imbalance problem.

flict. Specifically, ACR learns a class-balanced classifier via imposing consistency between its predictions and the *adjusted* outputs of the standard classifier. The adjusted outputs are designed to be appropriately biased toward the minority class. However, for the second finding, it is observed that the accuracy of pseudo-labels produced by the standard classifier varies as the class distribution of unlabeled data changes. We resolve this difficulty by refining the original pseudo-labels to match the true class distribution of unlabeled data and enhance their accuracy. Importantly, ACR realizes the adaptive refinery of pseudo-labels for various distributions in a unified formula by estimating the true class distribution.

We demonstrate the effectiveness of the proposed approach under various realistic LTSSL scenarios by varying the class distributions of unlabeled data. Despite its simplicity, the proposed algorithm improves recent LTSSL algorithms in all test cases, e.g., our method improves DARP [14], CReST [34], DASO [25] with up to **10.8%**, **11.2%**, and **7.2%** absolute increase on the test accuracy, respectively. Nevertheless, more importantly, in addition to three types of representative class distributions, i.e., *consistent*, *uniform*, and *reversed*, we also test our method under many other class distributions. As expected, our method significantly improves the performance when the class distributions are mismatched between labeled and unlabeled data.

2. Related Work

Semi-supervised learning. A popular class of Semi-supervised learning (SSL) algorithms use unlabeled data to improve the performance via learning to predict the pseudo-labels produced by the model, which can be viewed as a self-training process [2, 3, 23, 31]. Recent SSL algorithms [2,29] combine pseudo labeling and consistency regularization, which encourages similar predictions between two different views of an image, to improve the robustness of DNNs. As a representative approach, FixMatch [29] achieves significantly more superb performance than many other SSL algorithms in the image recognition task. Hence, the performance of SSL algorithm is quite sensitive to the quality of pseudo-labels. However, most existing SSL algorithms assume balanced class distributions of labeled and unlabeled data, resulting in poor generalization of the minority class due to biased pseudo-labels. Recently, Fixmatch has been frequently used as the base model for performance improvement under long-tailed class distribution.

Long-tailed semi-supervised learning. Long-tailed semi-supervised learning (LTSSL) has received significant attention for its practicality in many real-world tasks. For instance, DARP [14] and CReST [34] propose eliminating biased pseudo-labels generated by the model by distribution alignment to refine pseudo-labels according to the class distribution of labeled data. ABC [18] uses an auxiliary balanced classifier trained by down-sampling majority classes to improve the generalization. CoSSL [9] designs a novel feature enhancement module for the minority class using mixup [41] to train balanced classifiers. Although these algorithms can significantly enhance performance, they assume identical class distributions of labeled and unlabeled data. A recent work, DASO [25], proposes to handle this issue by employing a dynamic combination of linear and semantic pseudo-labels based on the current estimated class distribution of unlabeled data. It is noted that the accuracy of semantic pseudo-labels in DASO relies on the discrimination of learned representations. However, long-tailed class distribution negatively impacts representation learning, reducing the reliability of semantic pseudo-labels. We also demonstrate the gain of pseudo-labels F_1 of our method ACR compared to DASO in Figure 1d.

3. The Proposed Method

In this section, we first introduce the problem setup. Next, we present the ACR algorithm for handling unknown class distribution of unlabeled data.

3.1. Preliminaries

Problem setup. In LTSSL, we have a labeled dataset $\mathcal{D}^l = \{(x_i^{(l)}, y_i^{(l)})\}_{i=1}^N$ of size N and an unlabeled dataset

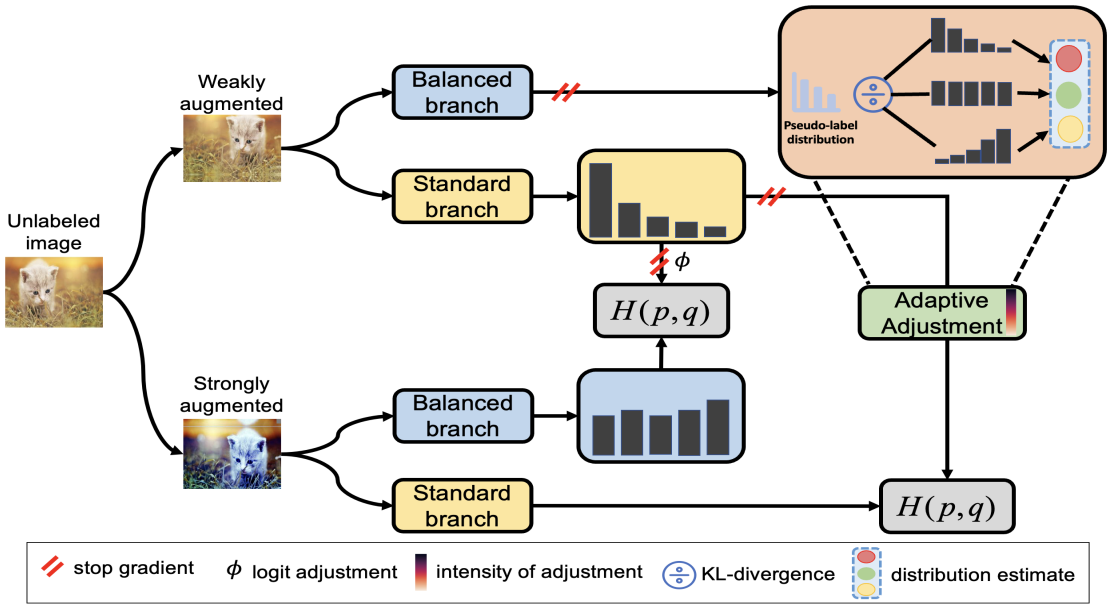


Figure 2. Illustration of the proposed framework. ACR uses a dual-branch network which utilizes unlabeled data of various class distributions by adaptively generating pseudo-labels that are good for representation and classifier learning. $H(p, q)$ denotes the cross-entropy.

$\mathcal{D}^u = \{x_j^{(u)}\}_{j=1}^M$ of size M, where $x_i^{(l)}, x_j^{(u)} \in \mathbb{R}^d$ are d-dimensional feature for labeled and unlabeled training samples, respectively. For the i-th labeled sample, it is associate with a ground-truth label $y_i^{(l)} \in \{0,1\}^C$ where C is the size of output label space. Let N_c denote the number of samples for class c in the labeled dataset, we have $N_1 \geq N_2 \geq ... \geq N_C$, and the imbalance ratio of the labeled dataset is denoted by $\gamma_l = \frac{N_1}{N_C}$. Similarly, let M_c denote the numbers of unlabeled samples for class c, and its imbalance ratio is $\gamma_u = \frac{\max_c M_c}{\min_c M_c}$ because we do not require any assumptions on the class distribution of the unlabeled dataset. Instead, we consider three representative distributions in this work, i.e., *consistent*, *uniform*, and *reversed*. Specifically, i) for *consistent* setting, we have $M_1 \geq M_2 \geq \cdots \geq M_C$ and $\gamma_u = \gamma_l$; ii) for *uniform* setting, we have $M_1 = M_2 = \cdots = M_C$, i.e., $\gamma_u = 1$; iii) for *reversed* setting, we have $M_1 \leq M_2 \leq \cdots \leq M_C$ and $\gamma_u = 1/\gamma_l$. Our goal is to train a classifier $f : \mathbb{R}^d \to [0,1]^C$ parameterized by θ using $\mathcal{D}^l$ and $\mathcal{D}^u$.

Semi-supervised learning. Many existing SSL algorithms seek to minimize a supervised classification loss on labeled data and an unsupervised regularizer on unlabeled data. Formally, the objective function is given as follows:

$$\min_{\theta \in \Theta} \underbrace{\sum_{i=1}^{N} \ell(f(x_i^{(l)};\theta), y_i^{(l)})}_{\text{supervised } (\mathcal{L}_{\text{labeled}})} + \underbrace{\sum_{j=1}^{M} \Omega(x_j^{(u)};\theta)}_{\text{unsupervised}}, \tag{1}$$

where ℓ denotes the cross-entropy loss, $\Omega(x_j^{(u)};\theta)$ is a per-sample reguarlizer. Particularly, in FixMatch [29], Ω is realized by the per-sample consistency regularization:

$$\mathcal{L}_{\text{con}} = \sum_{j=1}^{M} \underbrace{M(x_j^{(u)})}_{\text{sample mask}} \underbrace{\ell\left(f(\mathcal{A}(x_j^{(u)})), q_j\right)}_{\text{consistency}}, \tag{2}$$

where q_j is the pseudo-label of $x_j^{(u)}$ predicted by f, the sample masks $M(x_j^{(u)}) := \mathbb{I}\left(\max\left(\delta(f(x_j^{(u)}))\right) \geq \rho\right)$ selects unlabeled samples whose predicted confidence is higher than a predefined threshold ρ ($\rho = 0.95$ for FixMatch). Here, $\delta(\cdot)$ is the softmax function and $\mathbb{I}(\cdot)$ is the indicator function. To generate another view for each sample, $\mathcal{A}(x_j^{(u)})$ represents the specific augmentation scheme for $x_j^{(u)}$, such as Cutout [8] and RandomAugment [5]. Incorporating the consistency regularizer improves the model's robustness to spurious feature patterns. Additionally, it is worth noting that FixMatch includes all labeled data as part of unlabeled data without using their ground-truth labels, and we follow this practice in this work.

3.2. Adaptive consistency regularizer

Latterly, two-stage algorithms have been prevalently inspired by the empirical finding that long-tailed datasets suffer a more significant negative impact on classifier learning than representation learning [13, 42]. Therefore, two-stage algorithms discover class-balanced classifiers using various techniques such as re-sampling [13] and label smoothing [11], based on the feature extractor trained in the first stage. However, two-stage algorithms are computationally expen-

sive for SSL [18, 36]. We thus employ a double-branch network with a standard branch and a balanced branch to emerge the training of the standard classifier and a class-balanced classifier. For the standard branch, we employ FixMatch, which optimizes the standard cross-entropy on labeled data to learn good feature representations (denoted by f). For the balanced branch (denoted by $\tilde{f}$), we optimize the balanced softmax [22,28] which is an improved version of standard cross-entropy:

$$\mathcal{L}_{\text{b-labeled}} = -\sum_{i=1}^{N} \log \frac{e^{\tilde{f}_{y_i^{(l)}}(x_i^{(l)}) + \tau \cdot \log \pi_{y_i^{(l)}}}}{\sum_{c=1}^{C} e^{\tilde{f}_c(x_i^{(l)}) + \tau \cdot \log \pi_c}}, \quad (3)$$

where π_c is an estimate of class prior $\mathbb{P}(y=c)$ which is approximated by the empirical frequency on the training samples, and τ is a scaling parameter that affects the intensity of logit adjustment. Through minimizing Equation (3), $\tilde{f}$ can predict more balanced predictive probabilities. For both the standard and balanced branches, the consistency regularizer is employed, e.g., the balanced branch jointly optimizes Equation (3) and Equation (2). Ultimately, those two branches are jointly learned with a shared feature extractor. The diagram of ACR is summarized in Figure 2.

Can the double branch network handle unknown class distributions of unlabeled data? We respond to this question with two findings: i) pseudo-labels biased towards minority classes can benefit the classifier learning, whereas ii) pseudo-label distribution that approximates the true distribution helps learn better feature extractor. Inspired by the first finding, we propose to apply a simple logit adjustment (a.k.a post-hoc logit adjustment [22]) to the output of the standard classifier f whose predictions are initially biased towards majority classes. The refined logits are used to generate pseudo-labels which will be treated as targets for the balanced branch. Specifically, the pseudo-label of the j-th unlabeled data $x_j^{(u)}$ used in the consistency regularizer in the balanced branch is generated by:

$$\tilde{q}(x_j^{(u)}) = \arg\max f\left(x_j^{(u)}\right) - \tau \cdot \log \pi. \quad (4)$$

So that the consistency regularizer for balanced branch is:

$$\mathcal{L}_{\text{b-con}} = \sum_{j=1}^{M} \tilde{M}(x_j^{(u)}) \ell\left(\tilde{f}(\mathcal{A}(x_j^{(u)})), \tilde{q}_j\right). \quad (5)$$

Refining pseudo-labels in a unified formula. Inspired by the second finding, we strive for accurate pseudo-labels of unlabeled data for the standard branch. We empirically find that the standard branch can produce high quality of pseudo-labels and their overall distribution matches the true distribution in the *consistent* setting but are biased towards majority classes in the other two scenarios, with different degrees. Fortunately, we overcome this difficulty by a simple *dynamic logit adjustment* in a unified formula. Specifically, ACR automatically changes the intensity of logit adjustment, which is controlled by the scaling parameter τ, according to the unlabeled data class distribution estimate.

To estimate the true class distribution, we first craft three anchor distributions, including the class distribution of labeled data (π_{con}), a uniform distribution (π_{uni}), and a reversed class distribution of the labeled data (π_{rev}). We take the pseudo-labels produced by the balanced branch to estimate the distribution and calculate its distance to each anchor distribution. Specifically, let π_{est} be the estimate distribution from the model, we calculate the bidirectional KL-divergence, which is a symmetric distance measure as:

$$\begin{aligned} dist_{\text{con}} &= \frac{1}{2}\left(D_{KL}\left(\pi_{\text{con}} \| \pi_{\text{est}}\right) + D_{KL}\left(\pi_{\text{est}} \| \pi_{\text{con}}\right)\right) \\ dist_{\text{uni}} &= \frac{1}{2}\left(D_{KL}\left(\pi_{\text{uni}} \| \pi_{\text{est}}\right) + D_{KL}\left(\pi_{\text{est}} \| \pi_{\text{uni}}\right)\right) \\ dist_{\text{rev}} &= \frac{1}{2}\left(D_{KL}\left(\pi_{\text{rev}} \| \pi_{\text{est}}\right) + D_{KL}\left(\pi_{\text{est}} \| \pi_{\text{rev}}\right)\right), \end{aligned} \quad (6)$$

where $D_{KL}(p\|q) = \sum_{c=1}^{C} p_c \log\left(\frac{p_c}{q_c}\right)$. In our implementation, the model is trained for several iterations of warm-up before estimating the class distribution to prevent inconsistent results. We update π_{est} using an exponential moving average with the predicted class distribution of unlabeled data in each mini-batch. Finally, the anchor distribution corresponding to the smallest of the three distances calculated above is closest to the true class distribution of unlabeled data. Based on the distances to anchor distributions, ACR can adaptively handle various class distributions of unlabeled data in a unified formula that adjusts the scaling parameter τ as follows:

$$\tau(t) = \frac{2e^{dist_{\text{con}}^{(t-1)}}}{e^{dist_{\text{con}}^{(t-1)}} + e^{dist_{\text{uni}}^{(t-1)}} + e^{dist_{\text{rev}}^{(t-1)}}}, \quad (7)$$

where t is the training iteration. $dist_{\text{con}}^{(t-1)}$ represents the average distance between the predicted distribution at iteration $t-1$ and the *consistent* anchor distribution. The other two distances are defined likewise. From Equation (7), it is easy to derive that i) when the class distributions of labeled and unlabeled data are *consistent*, $e^{dist_{\text{rev}}}$ should be much larger than $e^{dist_{\text{con}}}$ so that τ will almost approach 0. The pseudo-labels for the standard branch are largely depend on its output. ii) In *uniform* setting, it yields a moderate value of τ which is below 1. iii) On the contrary, for the *reversed* cases, $e^{dist_{\text{con}}}$ will be quite larger than $e^{dist_{\text{rev}}}$, and as a result, τ will be larger than other two cases but less than 2, which means the pseudo-labels will be more biased towards minority classes. As expected, empirical results as depicted in Figure 4d coincides with our analysis. Ultimately, we maximize the consistency between the standard branch's outputs and refined pseudo-labels by the adaptive τ in Equation (7).

3.3. Sample mask generation

To utilize pseudo-labels that are likely to be correct for calculating consistency regularizer, the most popular way is to select unlabeled samples with high predictive confidence. We follow this principle but introduce another complementary term to further enhance the quality of pseudo-labels. We take the balanced branch for an example, and its consistency regularization loss is modified to:

$$\begin{aligned}\mathcal{L}_{\text{b-con}} &= \sum_{j=1}^{M} \tilde{M}(x_j^{(u)})\ell\left(\tilde{f}(\mathcal{A}(x_j^{(u)})), \tilde{q}_j\right), \\ \tilde{M}(x_j^{(u)}) &= \mathbb{I}\left(\max\left(\delta(\tilde{f}(x_j^{(u)}))\right) \geq \rho\right) \quad \vee \\ &\quad \mathbb{I}\left(\max\left(\delta(f(x_j^{(u)}) - \tau \cdot \log \pi)\right) \geq \rho\right),\end{aligned} \tag{8}$$

where $\delta(\cdot)$ denotes the softmax function. In this way, we can select more samples for the minority classes by considering the balanced branch's output. The consistency regularizer $\mathcal{L}_{\text{con}}$ for the standard branch can be written likewise, which can pick more examples that fit the true distribution. We can obtain more confident samples through the newly constructed sample mask, which is beneficial for consistency loss to work. Considering that the output logits from each branch have been well aligned with the target after a training period, both the original and the adjusted logits are trustworthy. We demonstrate the effect of the new sample mask selection principle in Section 4.5.

Overall, each branch of the network has two losses to minimize, i.e., the classification loss and the consistency regularizer. Put it together, our total objective function is:

$$\mathcal{L}_{\text{total}} = \underbrace{\mathcal{L}_{\text{labeled}} + \mathcal{L}_{\text{con}}}_{\text{standard branch}} + \underbrace{\mathcal{L}_{\text{b-labeled}} + \mathcal{L}_{\text{b-con}}}_{\text{balanced branch}}. \tag{9}$$

4. Experiments

We conduct extensive experiments to demonstrate the effectiveness of the proposed method under various class distributions of unlabeled data.

4.1. Experimental setup

The experiments we conducted are based on widely used datasets, including CIFAR10-LT [15], CIFAR100-LT [15], STL10-LT [4], and ImageNet-127 [9]. Recall that parameter γ_l is used to control the imbalance ratio of the labeled dataset, and we can decide the number of labeled samples for class c as $N_c = N_1 \cdot \gamma_l^{-\frac{c-1}{C-1}}$ once N_1 is given. Likewise, given the imbalance ratio of unlabeled dataset γ_u and M_1 (or M_C in the *reversed* setting), we set M_c as we did for the labeled dataset.

- **CIFAR-10-LT**: Following DASO [25], we test our method under $N_1 = 500, M_1 = 4000$ and $N_1 = 1500, M_1 = 3000$ settings. We report results with imbalance ratios $\gamma_l = \gamma_u = 100$ and $\gamma_l = \gamma_u = 150$. For *uniform* and *reversed* settings, we fix $\gamma_l = 100$ and adjust $\gamma_u \in \{1, 1/100\}$ to simulate various class distribution of unlabeled data.
- **CIFAR-100-LT**: We test our method under $N_1 = 50, M_1 = 400$ and $N_1 = 150, M_1 = 300$ settings. The imbalance ratio is set to $\gamma_l = \gamma_u = 10$ and $\gamma_l = \gamma_u = 20$. With a fixed $\gamma_l = 10$, we also test our method under $\gamma_u \in \{1, 1/10\}$ for the *uniform* and *reversed* unlabeled data class distributions.
- **STL10-LT**: Since ground-truth labels of unlabeled data in STL-10 are unknown, we conduct experiments by controlling the imbalance ratio of labeled data. We follow the settings by DASO and set $\gamma_l \in \{10, 20\}$.
- **ImageNet-127**: ImageNet127 is a naturally long-tailed dataset, so we do not need to construct the datasets manually. Following CoSSL [9], we downsample the image size to 32×32 and 64×64 due to limited resources.

Following previous work [9, 26], we implement our method using Wide ResNet-28-2 [40] on CIFAR10-LT, CIFAR100-LT, and STL10-LT; and ResNet-50 on ImageNet-127. Following FixMatch, we train the network for 500 epochs with 500 mini-batches in each epoch, with a batch size of 64, using standard SGD with momentum [24, 27, 30]. We use a cosine learning rate decay [21] which sets the learning rate to $\eta cos(\frac{7\pi t}{16T})$, where η is the initial learning rate, t is the current training step, and T is the total number of training steps. Considering the imbalanced test set in ImageNet-127, we set $\tau = 0.5$ for the balanced softmax defined in Equation (3), while for other datasets, we fix $\tau = 2$. To demonstrate the superiority of our approach, we compare with many existing LTSSL algorithms, including DARP [14], CReST [34], DASO [25], ABC [18], TRAS [36]. We measure the performance of all methods using top-1 accuracy on the test set. We report each method's mean and standard deviation of three independent runs in our experiments.

4.2. Results on CIFAR10/100-LT and STL10-LT

We first evaluate the performance when the class distributions are *consistent* (i.e., $\gamma_l = \gamma_u$) in Table 1. Subsequently, in Table 2 and Table 3, we report results when the unlabeled data class distribution is *uniform* or *reversed* (e.g., $\gamma_u = 1$ or $\gamma_u = 1/100$).

In case of $\gamma_l = \gamma_u$. We compare our approach ACR with several state-of-the-art LTSSL methods: DARP [14], CReST+ [34], and DASO [25]. Results are reported in Table 1. Without exception, ACR consistently outperforms existing methods by a large margin, even though most of

Algorithm	CIFAR10-LT				CIFAR100-LT			
	$\gamma = \gamma_l = \gamma_u = 100$		$\gamma = \gamma_l = \gamma_u = 150$		$\gamma = \gamma_l = \gamma_u = 10$		$\gamma = \gamma_l = \gamma_u = 20$	
	$N_1 = 500$ $M_1 = 4000$	$N_1 = 1500$ $M_1 = 3000$	$N_1 = 500$ $M_1 = 4000$	$N_1 = 1500$ $M_1 = 3000$	$N_1 = 50$ $M_1 = 400$	$N_1 = 150$ $M_1 = 300$	$N_1 = 50$ $M_1 = 400$	$N_1 = 150$ $M_1 = 300$
Supervised	47.3 ±0.95	61.9 ±0.41	44.2 ±0.33	58.2 ±0.29	29.6 ±0.57	46.9 ±0.22	25.1 ±1.14	41.2 ±0.15
w/ LA [22]	53.3 ±0.44	70.6 ±0.21	49.5 ±0.40	67.1 ±0.78	30.2 ±0.44	48.7 ±0.89	26.5 ±1.31	44.1 ±0.42
FixMatch [29]	67.8 ±1.13	77.5 ±1.32	62.9 ±0.36	72.4 ±1.03	45.2 ±0.55	56.5 ±0.06	40.0 ±0.96	50.7 ±0.25
w/ DARP [14]	74.5 ±0.78	77.8 ±0.63	67.2 ±0.32	73.6 ±0.73	49.4 ±0.20	58.1 ±0.44	43.4 ±0.87	52.2 ±0.66
w/ CReST+ [34]	76.3 ±0.86	78.1 ±0.42	67.5 ±0.45	73.7 ±0.34	44.5 ±0.94	57.4 ±0.18	40.1 ±1.28	52.1 ±0.21
w/ DASO [25]	76.0 ±0.37	79.1 ±0.75	70.1 ±1.81	75.1 ±0.77	49.8 ±0.24	59.2 ±0.35	43.6 ±0.09	52.9 ±0.42
FixMatch+LA [22]	75.3 ±2.45	82.0 ±0.36	67.0 ±2.49	78.0 ±0.91	47.3 ±0.42	58.6 ±0.36	41.4 ±0.93	53.4 ±0.32
w/ DARP [14]	76.6 ±0.92	80.8 ±0.62	68.2 ±0.94	76.7 ±1.13	50.5 ±0.78	59.9 ±0.32	44.4 ±0.65	53.8 ±0.43
w/ CReST+ [34]	76.7 ±1.13	81.1 ±0.57	70.9 ±1.18	77.9 ±0.71	44.0 ±0.21	57.1 ±0.55	40.6 ±0.55	52.3 ±0.20
w/ DASO [25]	77.9 ±0.88	82.5 ±0.08	70.1 ±1.68	79.0 ±2.23	50.7 ±0.51	60.6 ±0.71	44.1 ±0.61	55.1 ±0.72
FixMatch+ABC [18]	78.9 ±0.82	83.8 ±0.36	66.5 ±0.78	80.1 ±0.45	47.5 ±0.18	59.1 ±0.21	41.6 ±0.83	53.7 ±0.55
w/ DASO [25]	80.1 ±1.16	83.4 ±0.31	70.6 ±0.80	80.4 ±0.56	50.2 ±0.62	60.0 ±0.32	44.5 ±0.25	55.3 ±0.53
FixMatch w/ ACR (ours)	**81.6** ±0.19	**84.1** ±0.39	**77.0** ±1.19	**80.9** ±0.22	**55.7** ±0.12	**65.6** ±0.16	**48.0** ±0.75	**58.9** ±0.36

Table 1. Test accuracy of previous LTSSL algorithms and our proposed ACR under consistent class distributions, i.e., $\gamma_l = \gamma_u$, on CIFAR10-LT and CIFAR100-LT datasets. The best results are in **bold**.

Algorithm	CIFAR10-LT ($\gamma_l \neq \gamma_u$)				STL10-LT (γ_u = N/A)			
	$\gamma_u = 1$ (uniform)		$\gamma_u = 1/100$ (reversed)		$\gamma_l = 10$		$\gamma_l = 20$	
	$N_1 = 500$ $M_1 = 4000$	$N_1 = 1500$ $M_1 = 3000$	$N_1 = 500$ $M_C = 4000$	$N_1 = 1500$ $M_C = 3000$	$N_1 = 150$ $M = 100k$	$N_1 = 450$ $M = 100k$	$N_1 = 150$ $M = 100k$	$N_1 = 450$ $M = 100k$
FixMatch [29]	73.0 ±3.81	81.5 ±1.15	62.5 ±0.94	71.8 ±1.70	56.1 ±2.32	72.4 ±0.71	47.6 ±4.87	64.0 ±2.27
w/ DARP [14]	82.5 ±0.75	84.6 ±0.34	70.1 ±0.22	80.0 ±0.93	66.9 ±1.66	75.6 ±0.45	59.9 ±2.17	72.3 ±0.60
w/ CReST [34]	83.2 ±1.67	87.1 ±0.28	70.7 ±2.02	80.8 ±0.39	61.7 ±2.51	71.6 ±1.17	57.1 ±3.67	68.6 ±0.88
w/ CReST+ [34]	82.2 ±1.53	86.4 ±0.42	62.9 ±1.39	72.9 ±2.00	61.2 ±1.27	71.5 ±0.96	56.0 ±3.19	68.5 ±1.88
w/ DASO [25]	86.6 ±0.84	88.8 ±0.59	71.0 ±0.95	80.3 ±0.65	70.0 ±1.19	78.4 ±0.80	65.7 ±1.78	75.3 ±0.44
w/ ACR (ours)	**92.1** ±0.18	**93.5** ±0.11	**85.0** ±0.09	**89.5** ±0.17	**77.1** ±0.24	**83.0** ±0.32	**75.1** ±0.70	**81.5** ±0.25

Table 2. Test accuracy of previous LTSSL algorithms and our proposed ACR under inconsistent class distributions, i.e., $\gamma_l \neq \gamma_u$, on CIFAR10-LT and STL10-LT datasets. The γ_l is fixed to 100 for CIFAR10-LT, while it is set to 10 and 20 for STL10-LT dataset. The best results are in **bold**.

Algorithm	CIFAR100-LT ($\gamma_l \neq \gamma_u$)			
	$\gamma_u = 1$ (uniform)		$\gamma_u = 1/10$ (reversed)	
	$N_1 = 50$ $M_1 = 400$	$N_1 = 150$ $M_1 = 300$	$N_1 = 50$ $M_C = 400$	$N_1 = 150$ $M_C = 300$
FixMatch [29]	45.5 ±0.71	58.1 ±0.72	44.2 ±0.43	57.3 ±0.19
w/ DARP [14]	43.5 ±0.95	55.9 ±0.32	36.9 ±0.48	51.8 ±0.92
w/ CReST [34]	43.5 ±0.30	59.2 ±0.25	39.0 ±1.11	56.4 ±0.62
w/ CReST+ [34]	43.6 ±1.60	58.7 ±0.16	39.1 ±0.77	56.4 ±0.78
w/ DASO [25]	53.9 ±0.66	61.8 ±0.98	51.0 ±0.19	60.0 ±0.31
w/ ACR (ours)	**66.0** ±0.25	**73.4** ±0.22	**57.0** ±0.46	**67.6** ±0.12

Table 3. Test accuracy on CIFAR100-LT dataset under *uniform* and *reversed* class distributions. The best results are in **bold**.

Algorithm	32×32	64×64
FixMatch [29]	29.7	42.3
w/ DARP [14]	30.5	42.5
w/ DARP+cRT [14]	39.7	51.0
w/ CReST+ [34]	32.5	44.7
w/ CReST++LA [22]	40.9	55.9
w/ CoSSL [9]	43.7	53.9
w/ TRAS [36]	46.2	54.1
w/ ACR (ours)	**57.2**	**63.6**

Table 4. Test accuracy on ImageNet-127 dataset. The best results are in **bold**.

these methods are particularly developed based on the assumption that labeled and unlabeled data share the same class distribution. This observation verifies the superior performance of our method. By further combing with logit adjustment (LA) and auxiliary balanced classifier (ABC), DASO achieves a noticeable improvement. However, its performance is still 3.7% below ACR on average.

In case of $\gamma_l \neq \gamma_u$. In real-world datasets, the class distribution of unlabeled data is likely to be significantly inconsistent with labeled data. Therefore, we consider *uniform* and *reversed* class distributions, e.g., $\gamma_u = 1$ or $\gamma_u = 1/100$ for CIFAR10-LT. On the STL10-LT dataset, due to the unknown ground-truth labels of the unlabeled data, we can only control the imbalance ratio of labeled data. The results are summarized in the Table 2 and Table 3.

It can be seen that our method achieves the best results when the class distributions of unlabeled data are inconsistent. For example, ACR obtains 15.6% and 20.1% absolute performance gains over FixMatch under $\gamma_u = 1$ and $\gamma_u = 1/100$ on CIFAR10-LT, respectively. Similarly, the CIFAR100-LT results show that our method outperforms DASO by an average of 9.3% accuracy increase. For STL10-LT, ACR achieves the best results with averaged 6.8% accuracy gain compared with DASO, even if the unlabeled data distribution is unknown. Generally speaking, empirical results under unknown class distributions of unlabeled data on three datasets justify that ACR can effectively utilize unlabeled data to alleviate the negative impact of class imbalance.

4.3. Results on ImageNet-127

ImageNet127 is initially introduced in previous work [12] and is applied to LTSSL by CReST [34], which groups the 1000 classes of ImageNet [7] into 127 classes based on the WordNet hierarchy. Compared with other datasets, we do not need to construct the dataset artificially because it naturally follows a long-tailed class distribution with an imbalance ratio $\gamma \approx 286$. Following CoSSL [9], we downsample the original images to smaller sizes of 32×32 or 64×64 pixels using the box method from the Pillow library and randomly select 10% training samples to form the labeled set. It is worth noting that the test set of ImageNet-127 is also long-tailed, so we set the scaling parameter $\tau = 0.5$ in balanced softmax to reduce the bias of the classifier towards the minority class. The results are summarized in Table 4. We can see that ACR achieves superior results for both image sizes 32×32 and 64×64 with 11% and 9.5% absolute improvement on test accuracy compared with TRAS [36], respectively. The results show that our method can successfully apply to tasks with long-tailed test datasets.

4.4. Results under more class distributions

To examine the performance of our method in more imbalanced scenarios, we conduct additional experiments on CIFAR-10-LT by fixing $\gamma_l = 100$ while varying the imbalance ratio of unlabeled data γ_u from *consistent* to *reversed* by a step size 20. We set $N_1 = 500$ and $M_1 = 4000$ ($M_C = 4000$ in *reversed* setting) and compare the performance with DASO w/ LA [25]. The results are reported in Figure 3. It can be easily observed that our method ACR consistently outperforms DASO in all test cases. Overall, the performance gain becomes increasingly significant as the class distribution of unlabeled data changes from *consistent* to *reversed*. This demonstrates the capability of our method to handle various realistic LTSSL problems adaptively.

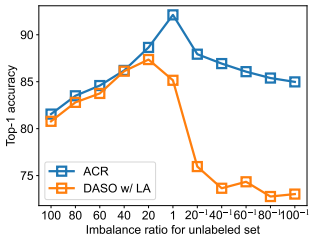


Figure 3. Generalize to more realistic LTSSL settings.

Ablations	CIFAR10-LT			CIFAR100-LT		
	Con	Uni	Rev	Con	Uni	Rev
ACR(ours)	81.6	92.1	85.0	55.7	66.0	57.0
w/o sample mask principle	81.7	91.1	84.6	55.0	63.7	55.0
w/o adaptive LA	76.8	92.4	85.1	53.5	62.8	56.1
w/o LA for balanced branch	74.3	90.6	83.5	54.5	66.2	56.7
w/o balanced softmax	76.7	93.0	84.8	55.3	65.6	57.3
w/o gradients from balanced branch	73.7	92.3	85.2	54.3	65.2	56.7
w/o labeled data in unlabeled set	81.0	92.7	79.9	56.1	66.4	56.8

Table 5. Ablation studies of our proposed ACR algorithm. *Con*, *Uni*, and *Rev* represent *consistent*, *uniform*, and *reversed* for short.

4.5. Systematic analysis of the proposed method

To better understand our method, we conduct extensive ablation studies. Due to limited space, we defer more detailed analysis to supplementary material.

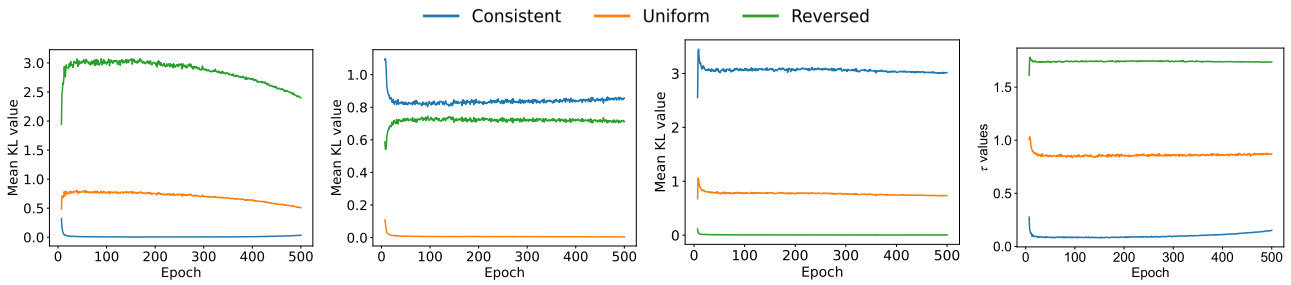
Distribution estimation. Figure 4 illustrates the bidirectional KL-divergence for different unlabeled data distributions and the τ values during the training. From Figures 4a to 4c, we can observe that distance between the estimated and the true distributions is small under three different settings so that our method can accurately determine the underlying distribution of unlabeled data. It implies that the balanced classifier has a strong generalization ability.

Scaling parameter τ. Our model's accurate distribution discrimination ability benefits the proposed adaptive consistency regularizer, which adjusts the value of scaling parameter τ in Equation (7) to handle various unlabeled data class distributions. Figure 4d displays the dynamic values of τ under three settings. We can see that τ increases from a minimal value for *consistent* to a relatively large value for *reversed* setting, showing good adaptability of our method.

Visualization. Moreover, we visualize the learned representations of ACR using the t-distributed stochastic neighbor embedding (t-SNE) [32] and compare them with the previous method DASO. Figure 5 illustrates the comparison results on test set under *uniform* and *reversed* settings. It can be seen from the figure that the representations obtained by ACR allow for clearer classification boundaries.

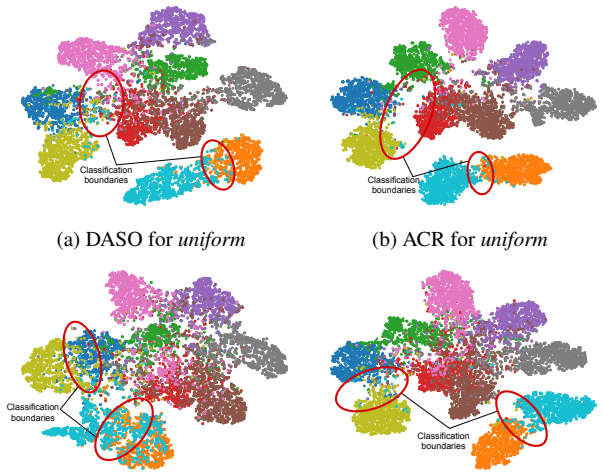
Additionally, we conduct many ablation studies on the critical components of ACR on CIFAR10-LT and CIFAR100-LT. Detailed results are reported in Table 5.

Impact of sample mask principle. To verify the effect of the mask generation principle, we perform ablation experiments to compare our principle with a standard mask generation strategy based on the confidence of the pseudo-labels used as the targets in the consistency loss. With an averaged 1.7% improvement on CIFAR100-LT, it clearly shows the superiority of our sample mask generation principle.



(a) Distance for *consistent* setting (b) Distance for *uniform* setting (c) Distance for *reversed* setting (d) τ values for LA

Figure 4. (4a to 4c): The average bi-directional KL distance of three settings for each epoch during the training of CIFAR10-LT and the imbalance ratio for *consistent* distribution and *reversed* distribution are 100 and $1/100$, respectively. (4d): The dynamic τ values of LA used in standard branch consistency regularizer.



(a) DASO for *uniform* (b) ACR for *uniform*

(c) DASO for *reversed* (d) ACR for *reversed*

Figure 5. The t-SNE visualization of the test set for DASO and ACR on CIFAR-10-LT dataset in *uniform* and *reversed* settings.

Impact of adaptive LA on the standard branch. ACR adjusts the intensity of LA applied to pseudo-labels in consistency loss according to the current distribution of unlabeled data. When the intensity of LA applied to the standard branch is fixed (we set $\tau = 2$ in Equation (7) in this experiment), the performance decreases in all three settings on CIFAR100-LT dataset, especially in *consistent* and *uniform* settings. Also, the performance in *consistent* setting decreases 4.8% on CIFAR10-LT, showing the importance of developing an adaptive LA strategy for the standard branch.

Impact of LA on the balanced branch. To study the influence of LA on the balanced branch, we remove the LA in the consistency loss. The results reflect an averaged 0.4% slightly drop in accuracy on CIFAR100-LT. However, the performance decreases significantly (3.4%) on CIFAR-10-LT, indicating the necessity of LA for the balanced branch.

Impact of balanced softmax. We replace the balanced softmax with standard cross-entropy to study its impact on performance. The results indicate a 0.2% drop indicating that balanced softmax is not quite sensitive for CIFAR100-LT. However, for CIFAR10-LT, the performance penalty in the *consistent* setting exceeds 7.3%.

Impact of balanced branch on feature learning. The two branches in ACR will update the feature extractor together, as mentioned above. So we explore the effect of the balanced branch on representation learning by blocking the gradient of the balanced branch to update the feature extractor. The results in Table 5 indicate that the balanced branch does not negatively affect representation learning. Conversely, it has some promoting effects, especially for *consistent* setting when the feature extractor is updated by both branches simultaneously.

Impact of labeled data in the unlabeled set. Following FixMatch [29], we include all labeled data as part of unlabeled data without their labels when constructing the unlabeled set. So when we exclude labeled data from the unlabeled set during the training, the results decrease dramatically, particularly for *reversed* setting on CIFAR10-LT.

5. Conclusion

This paper presents a simple and effective method by minimizing the adaptive consistency regularizer (ACR) for long-tailed semi-supervised learning with unknown class distributions of the unlabeled data. Our main idea is to i) benefit classifier learning by generating pseudo-labels that are properly biased towards minority classes while ii) benefit representation learning by generating pseudo-labels whose distribution approximates the true class distribution. We implement our idea in a double-branch network and realize ACR through on-the-fly distribution estimation and a novel dynamic logit adjustment. We empirically show that our method significantly outperforms all competing methods under various scenarios, offering a solid baseline for future studies in this task.

References

[1] Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al. Deep speech 2: End-to-end speech recognition in english and mandarin. In *International conference on machine learning*, pages 173–182. PMLR, 2016. 1

[2] 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 *International Conference on Learning Representations*, 2019. 2

[3] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. *Advances in Neural Information Processing Systems*, 32:5050–5060, 2019. 1, 2

[4] 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*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011. 5

[5] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703, 2020. 3

[6] Jiequan Cui, Zhisheng Zhong, Shu Liu, Bei Yu, and Jiaya Jia. Parametric contrastive learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 715–724, 2021. 1

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 7

[8] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. 3

[9] Yue Fan, Dengxin Dai, Anna Kukleva, and Bernt Schiele. Cossl: Co-learning of representation and classifier for imbalanced semi-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14574–14584, 2022. 2, 5, 6, 7

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1

[11] Yin-Yin He, Jianxin Wu, and Xiu-Shen Wei. Distilling virtual examples for long-tailed recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 235–244, 2021. 3

[12] Minyoung Huh, Pulkit Agrawal, and Alexei A Efros. What makes imagenet good for transfer learning? *arXiv preprint arXiv:1608.08614*, 2016. 7

[13] 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*, 2020. 3

[14] 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. *Advances in Neural Information Processing Systems*, 33:14567–14579, 2020. 1, 2, 5, 6

[15] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 5

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017. 1

[17] Zhengfeng Lai, Chao Wang, Henrry Gunawan, Sen-Ching S. Cheung, and Chen-Nee Chuah. Smoothed adaptive weighting for imbalanced semi-supervised learning: Improve reliability against unknown distribution data. In *International Conference on Machine Learning*, pages 11828–11843, 2022. 1

[18] Hyuck Lee, Seungjae Shin, and Heeyoung Kim. Abc: Auxiliary balanced classifier for class-imbalanced semi-supervised learning. *Advances in Neural Information Processing Systems*, 34:7082–7094, 2021. 1, 2, 4, 5, 6

[19] Jun Li, Zichang Tan, Jun Wan, Zhen Lei, and Guodong Guo. Nested collaborative learning for long-tailed visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6949–6958, 2022. 1

[20] 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*, pages 2537–2546, 2019. 1

[21] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 5

[22] Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. In *International Conference on Learning Representations*, 2020. 4, 6

[23] 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*, 41(8):1979–1993, 2018. 1, 2

[24] Yurii Nesterov. A method of solving a convex programming problem with convergence rate $o(1/k^2)$. In *Sov. Math. Dokl*, volume 27. 5

[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*, pages 9786–9796, 2022. 1, 2, 5, 6, 7

[26] Avital Oliver, Augustus Odena, Colin A Raffel, Ekin Dogus Cubuk, and Ian Goodfellow. Realistic evaluation of deep semi-supervised learning algorithms. *Advances in neural information processing systems*, 31, 2018. 5

[27] Boris T Polyak. Some methods of speeding up the convergence of iteration methods. *Ussr computational mathematics and mathematical physics*, 4(5):1–17, 1964. 5

[28] Jiawei Ren, Cunjun Yu, Xiao Ma, Haiyu Zhao, Shuai Yi, et al. Balanced meta-softmax for long-tailed visual recognition. *Advances in Neural Information Processing Systems*, 33:4175–4186, 2020. 4

[29] 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*, 33:596–608, 2020. 1, 2, 3, 6, 8

[30] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147. PMLR, 2013. 5

[31] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in Neural Information Processing Systems*, 30:1195–1204, 2017. 1, 2

[32] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(11), 2008. 7

[33] Xudong Wang, Long Lian, Zhongqi Miao, Ziwei Liu, and Stella X Yu. Long-tailed recognition by routing diverse distribution-aware experts. *arXiv preprint arXiv:2010.01809*, 2020. 1

[34] 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, 5, 6, 7

[35] Tong Wei and Yu-Feng Li. Does tail label help for large-scale multi-label learning? *IEEE Transactions on Neural Networks and Learning Systems*, 31(7):2315–2324, 2019. 1

[36] Tong Wei, Qian-Yu Liu, Jiang-Xin Shi, Wei-Wei Tu, and Lan-Zhe Guo. Transfer and share: Semi-supervised learning from long-tailed data. *Machine Learning*, 2022. 1, 4, 5, 6, 7

[37] Tong Wei, Jiang-Xin Shi, Wei-Wei Tu, and Yu-Feng Li. Robust long-tailed learning under label noise. *CoRR*, abs/2108.11569, 2021. 1

[38] Liuyu Xiang, Guiguang Ding, and Jungong Han. Learning from multiple experts: Self-paced knowledge distillation for long-tailed classification. In *European Conference on Computer Vision*, pages 247–263. Springer, 2020. 1

[39] Qizhe Xie, Zihang Dai, Eduard H. Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. In *Advances in Neural Information Processing Systems*, 2020. 1

[40] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016. 5

[41] Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018. 2

[42] Zhisheng Zhong, Jiequan Cui, Shu Liu, and Jiaya Jia. Improving calibration for long-tailed recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 16489–16498, 2021. 3

[43] 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

[44] Beier Zhu, Yulei Niu, Xian-Sheng Hua, and Hanwang Zhang. Cross-domain empirical risk minimization for unbiased long-tailed classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022. 1