

# Twice Class Bias Correction for Imbalanced Semi-supervised Learning

Lan Li, Bowen Tao, Lu Han, De-chuan Zhan, Han-jia Ye\*

National Key Laboratory for Novel Software Technology, Nanjing University  
School of Artificial Intelligence, Nanjing University  
Nanjing, 210023, China  
{lil, taobw, hanlu, yehj}@lamda.nju.edu.cn, zhandc@nju.edu.cn

## Abstract

Differing from traditional semi-supervised learning, class-imbalanced semi-supervised learning presents two distinct challenges: (1) The imbalanced distribution of training samples leads to model bias towards certain classes, and (2) the distribution of unlabeled samples is unknown and potentially distinct from that of labeled samples, which further contributes to class bias in the pseudo-labels during training. To address these dual challenges, we introduce a novel approach called **Twice Class Bias Correction (TCBC)**. We begin by utilizing an estimate of the class distribution from the participating training samples to correct the model, enabling it to learn the posterior probabilities of samples under a class-balanced prior. This correction serves to alleviate the inherent class bias of the model. Building upon this foundation, we further estimate the class bias of the current model parameters during the training process. We apply a secondary correction to the model’s pseudo-labels for unlabeled samples, aiming to make the assignment of pseudo-labels across different classes of unlabeled samples as equitable as possible. Through extensive experimentation on CIFAR10/100-LT, STL10-LT, and the sizable long-tailed dataset SUN397, we provide conclusive evidence that our proposed TCBC method reliably enhances the performance of class-imbalanced semi-supervised learning.

## Introduction

Semi-supervised learning (SSL) (Van Engelen and Hoos 2020; Li and Liang 2019; Yang et al. 2019a,b) has shown promise in using unlabeled data to reduce the cost of creating labeled data and improve model performance on a large scale. In SSL, many algorithms generate pseudo-labels (Lee et al. 2013) for unlabeled data based on model predictions, which are then utilized to regularize model training. However, most of these methods assume that the data is balanced across classes. In reality, many real-world datasets exhibit imbalanced distributions (Buda, Maki, and Mazurowski 2018; Pouyanfar et al. 2018; Byrd and Lipton 2019; Li, Zhan, and Li 2022), with some classes being much more prevalent than others. This imbalance affects both the labeled and unlabeled samples, resulting in biased pseudo-labels that further worsen the class imbalance during train-

ing and ultimately hinder model performance. Recent research (Wei et al. 2021; Lee, Shin, and Kim 2021; Guo and Li 2022; Wei et al. 2022; Tao et al. 2023) has highlighted the significant impact of class imbalance on the effectiveness of pseudo-labeling methods. Therefore, it is crucial to develop SSL algorithms that can effectively handle class imbalance in both labeled and unlabeled data, leading to improved performance in real-world scenarios.

Unlike traditional SSL techniques that assume identical distributions of labeled and unlabeled data, this paper addresses a more generalized scenario of imbalanced SSL. Specifically, we consider situations where the distribution of unlabeled samples is unknown and may diverge from the distribution of labeled samples (Oh, Kim, and Kweon 2022; Wang et al. 2022; Wei and Gan 2023).

In this context, two challenges need to be addressed: (1) How to mitigate the model’s class bias induced by training on imbalanced data, and (2) How to leverage the model’s predictions on unlabeled samples during the training process to obtain improved pseudo-labels. To elucidate these challenges, we devised an experiment, as depicted in Figure 1a, where labeled samples follow a long-tailed distribution, and unlabeled samples follow a uniform distribution. Figure 1b illustrates the recall performance of the FixMatch model trained under this scenario on the test set. Despite the presence of many minority class samples in the unlabeled data, due to the class imbalance in the labeled samples, the eventual model still exhibits significant class bias. Figure 1c represents the pseudo-label class distribution on unlabeled data obtained by FixMatch, highlighting a notable class bias present in FixMatch’s pseudo-labels, which can adversely affect model training. To address these two challenges within imbalanced SSL, we introduce a novel approach termed “**Twice Class Bias Correction**” (TCBC).

The first challenge primarily stems from the potential inconsistency between the class distributions of labeled and unlabeled samples. During training, the class distribution of training samples may undergo substantial fluctuations, rendering it infeasible to rely on the assumption of consistent class distribution to reduce model bias. To address this challenge, we dynamically estimate the class distribution of participating training samples. Leveraging the assumption of consistent class-conditional probabilities, we guide the model to learn a reduced class bias objective on both labeled

\*Corresponding author.

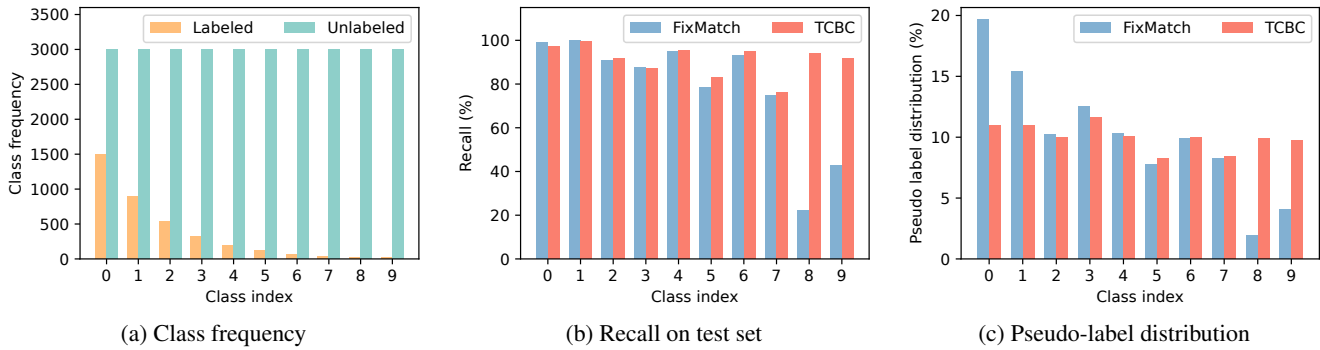


Figure 1: Experiments on CIFAR-10. (a) Class frequency of the labeled and unlabeled data. (b) Recall for each class on test set. (c) The pseudo-label distribution generated from the unlabeled set.

and pseudo-labeled samples. Specifically, this objective is the posterior probabilities of samples under the assumption of a class-balanced prior. As depicted in Figure 1b, our approach significantly diminishes the model’s bias across different classes.

The complexity of the second challenge lies in the presence of class bias in the model during training and the unknown distribution of unlabeled samples. This results in uncontrollable pseudo-labels acquired by the model on unlabeled data. A balanced compromise solution is to ensure that the model acquires pseudo-labels as equitably as possible across different classes. Hence, we introduce a method based on the model’s output on samples to estimate the model’s class bias under current parameters. We leverage this bias to refine predictions on unlabeled samples, thereby reducing class bias in pseudo-labels. As illustrated in Figure 1c, our approach achieves a less biased pseudo-label distribution on class-balanced unlabeled samples.

Our primary contributions are as follows:

- We present a novel technique that harnesses the class distribution of training samples to rectify the biases introduced by class imbalance in the model’s learning objectives.
- We introduce a method to evaluate the model’s class bias under the current model parameter conditions during the training process and utilize it to refine pseudo-labels.
- Our approach is straightforward yet effective, as demonstrated by extensive experiments in various imbalanced SSL settings, highlighting the superiority of our method. Code and appendix is publicly available at <https://github.com/Lain810/TCBC>.

## Related Work

**Class-imbalanced learning** attempts to learn models that generalize well on each class from imbalanced data. Resampling and reweighting are two commonly used methods. Resampling methods balance the number of training samples for each class in the training set by undersampling (Drummond, Holte et al. 2003; Buda, Maki, and Mazurowski 2018) the majority classes or oversampling (Buda, Maki, and Mazurowski 2018; Pouyanfar et al. 2018; Byrd and Lipton 2019) the minority classes. Reweighting (Huang et al.

2016; Wang, Ramanan, and Hebert 2017; Cui et al. 2019) methods assign different losses to training samples of each class or each example. In addition, some works have used logits compensation (Cao et al. 2019; Menon et al. 2020; Ren et al. 2020; Ye et al. 2020) based on class distribution or transfer learning (Yin et al. 2019; Chu et al. 2020; Ye, Zhan, and Chao 2021) to address this problem.

**Semi-supervised learning** try to improve the model’s performance by leveraging unlabeled data (Berthelot et al. 2019). A common approach in SSL is to utilize model predictions to generate pseudo-labels for each unlabeled data and use these pseudo-labels for supervised training. Recent SSL algorithms, exemplified by FixMatch (Sohn et al. 2020), achieve enhanced performance by encouraging consistent predictions between two different views of an image and employing consistency regularization. While these methods have seen success, most of them are based on the assumption that labeled and unlabeled data follow a uniform label distribution. When applied to class-imbalanced scenarios, the performance of these methods can significantly deteriorate due to both model bias and pseudo-label bias.

**Class imbalanced semi-supervised learning** has garnered widespread attention due to its alignment with real-world tasks. The DARP (Kim et al. 2020) refines initial pseudo-labels through convex optimization, aiming to alleviate distribution bias resulting from imbalanced and unlabeled training data. CREST (Wei et al. 2021) employs a combination of re-balancing and distribution alignment techniques to mitigate training bias. ABC (Lee, Shin, and Kim 2021) introduces an auxiliary balanced classifier trained through down-sampling of majority classes. However, many existing methods assume a similarity between marginal distributions of labeled and unlabeled data classes, an assumption that often doesn’t hold or remains unknown before training. To address this limitation, DASO (Oh, Kim, and Kweon 2022) combines pseudo-labels from both linear and similarity-based classifiers, leveraging their complementary properties to combat bias. L2AC (Wang et al. 2022) introduces a bias adaptive classifier to tackle the issue of training bias in imbalanced semi-supervised learning tasks. Compared to these methods, we account for both class bias in the final model and pseudo-labeling during training.

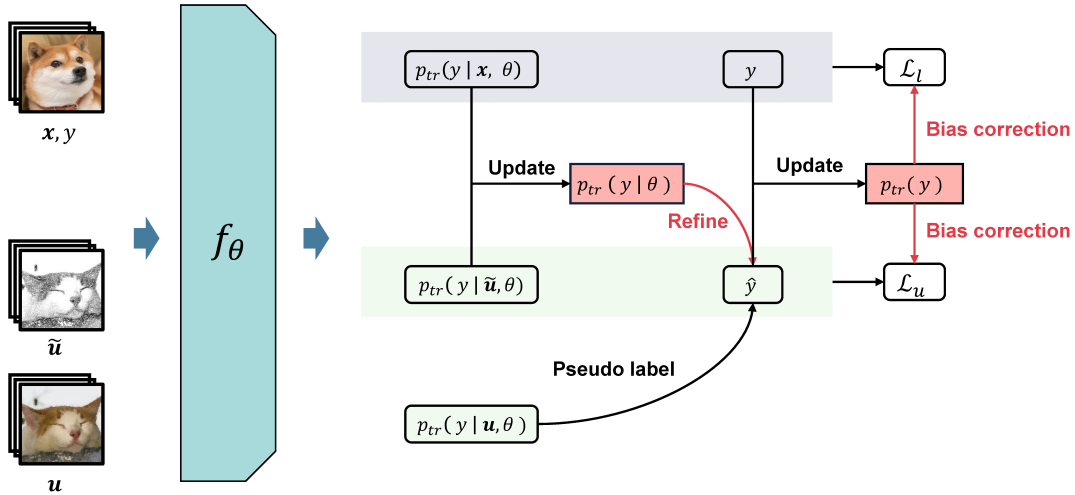


Figure 2: Illustration of the proposed method. Our approach corrects the class bias in the model’s learning objectives by estimating the class distribution based on the participating training data. Simultaneously, it refines pseudo-labels by estimating the model’s class bias under the current parameters, reducing bias in the pseudo-labels.

**Preliminaries**

Assuming the existence of a labeled set denoted by  $\mathcal{D}_l = \{(\mathbf{x}_n, y_n)\}_{n=1}^N$  and an unlabeled set denoted by  $\mathcal{D}_u = \{\mathbf{u}_m\}_{m=1}^M$ , where  $\mathbf{x}_n, \mathbf{u}_m \in \mathcal{X}$  represents training samples in the input space, and  $y_n \in \mathcal{Y}$  represents the labels assigned to labeled samples, with  $\mathcal{Y}$  denoting the label space consisting of  $K$  classes. The class distribution of labeled data and unlabeled data is denoted by  $p_l(y)$  and  $p_u(y)$ , respectively. Moreover, we denote  $N_k$  and  $M_k$  as the number of labeled and unlabeled samples in class  $k$ , respectively. Without loss of generality, we assume that the classes are arranged in descending order based on the number of training samples, such that  $N_1 \geq N_2 \geq \dots \geq N_K$ . The goal of imbalanced SSL is to learn the model  $f$  that generalizes well on each class from imbalanced data, parameterized by  $\theta$ .

In SSL, one effective method involves utilizing pseudo-labeling techniques to enhance the training dataset with pseudo-labels for unlabeled data. In pseudo-labeling SSL, each unlabeled sample is provided with a pseudo-label based on the model’s prediction. An optimization problem with an objective function  $\mathcal{L} = \mathcal{L}_s + \lambda \mathcal{L}_u$  is utilized to train the model on both labeled and pseudo-labeled samples. The loss function consists of two terms: the supervised loss  $\mathcal{L}_s$  computed on labeled data and the unsupervised loss  $\mathcal{L}_u$  computed on unlabeled data. The parameter  $\lambda$  is used to balance the loss from labeled data and pseudo-labeled data. The computation formula for the supervised loss on labeled samples is given by

$$\mathcal{L}_s = \frac{1}{|\mathcal{B}_l|} \sum_{\mathbf{x}_i \in \mathcal{B}_l} \mathcal{H}(y_i, p(y|\mathbf{x}_i)), \quad (1)$$

where  $\mathcal{B}_l$  denotes a iteration of labeled data sampled from  $\mathcal{D}_l$ .  $p(y|\mathbf{x}_i) = \text{softmax}(f(\mathbf{x}_i))$  represents the output probability, and  $\mathcal{H}$  is the cross-entropy loss. Similarly, the loss

function on unlabeled samples can also be formulated as

$$\mathcal{L}_u = \frac{1}{|\mathcal{B}_u|} \sum_{\mathbf{u}_j \in \mathcal{B}_u} \mathcal{M} \cdot \mathcal{H}(\hat{y}_j, p(y|\tilde{\mathbf{u}}_j)), \quad (2)$$

where  $\mathcal{M} = \mathbf{I}[\max(p(y|\mathbf{u}_j)) \geq \tau_c]$  and  $\mathbf{I}$  is the indicator function,  $\tau_c$  is the threshold.  $\hat{y}_j = \arg \max p(y|\mathbf{u}_j)$  denotes the pseudo-label assigned by the model to the weakly augmented sample  $\mathbf{u}_j$ , and  $\tilde{\mathbf{u}}_j$  represents the unlabeled samples that undergo strong augmentation.

**Model Bias Correction**

In imbalanced SSL, the first challenge to address is *how to learn a model that is devoid of class bias*. Generally, achieving an evaluation model without class bias involves approximating a Bayesian optimal model under a class-balanced distribution. This entails minimizing the loss on data where the marginal distribution follows a uniform distribution,  $p_{ev}(y) = 1/K$ . The corresponding posterior probability of the samples is denoted as  $p_{ev}(y|\mathbf{x})$ .

When directly optimizing surrogate loss, such as the softmax cross-entropy loss, on training data with an imbalanced class distribution  $p_{tr}(y)$ , the learned posterior probability of the samples becomes  $p_{tr}(y|\mathbf{x})$ , which differs from  $p_{ev}(y|\mathbf{x})$  and tends to favor the classes with more samples. Therefore, resolving this challenge involves addressing the mismatch between  $p_{tr}(y|\mathbf{x})$  and  $p_{ev}(y|\mathbf{x})$ .

When considering imbalanced SSL, the situation becomes more complex. The training data consist of labeled data (with a marginal distribution  $p_l(y)$ ) and unlabeled data (with a marginal distribution  $p_u(y)$ ). Here,  $p_l(y)$  represents a known long-tail distribution, while  $p_u(y)$  represents an unknown distribution that may differ from  $p_l(y)$ .

Assuming that the training samples, including labeled, unlabeled samples and test samples, are drawn from the same probability distribution of the corresponding class,

$$p_l(\mathbf{x}|y) = p_u(\mathbf{x}|y) = p_{ev}(\mathbf{x}|y) = p(\mathbf{x}|y). \quad (3)$$

By applying the Bayes' theorem, we establish that  $p(y|\mathbf{x}) \propto p(\mathbf{x}|y)p(y)$ . If we consider labeled and unlabeled losses separately, due to the fact that  $p_l(y) \neq p_u(y)$ , the posterior probabilities learned from these two parts are different. Specifically, let's assume a sample  $\mathbf{x}_i$  exists in both the labeled and unlabeled data. The posterior probabilities  $p_{tr}(y|\mathbf{x}_i)$  learned by the model represent a weighted average of  $p_l(y|\mathbf{x}_i)$  and  $p_u(y|\mathbf{x}_i)$ . We denote the marginal probability distribution corresponding to  $p_{tr}(y|\mathbf{x})$  as  $p_{tr}(y)$ . According to Eq. (3), and considering  $p_{ev}(y) = 1/K$ , we have:

$$\begin{aligned} p_{tr}(y|\mathbf{x}) &\propto p(\mathbf{x}|y) \cdot p_{tr}(y) \\ &\propto p_{ev}(y|\mathbf{x})/p_{ev}(y) \cdot p_{tr}(y) \\ &\propto p_{ev}(y|\mathbf{x}) \cdot p_{tr}(y). \end{aligned} \quad (4)$$

Because we aim to learn  $p_{ev}(y|\mathbf{x})$ , which can be represented as  $p_{ev}(y|\mathbf{x}) = \text{softmax}(f(\mathbf{x}))$ , but what we learn through the cross-entropy loss is  $p_{tr}(y|\mathbf{x})$ . According to Eq. (4), we have:

$$\begin{aligned} p_{tr}(y|\mathbf{x}) &\propto p_{ev}(y|\mathbf{x}) \cdot p_{tr}(y) \\ &\propto \text{softmax}(f(\mathbf{x}) + \ln p_{tr}(y)). \end{aligned} \quad (5)$$

Therefore, the  $p(y|\mathbf{x}_i)$  in  $\mathcal{L}_s$  and  $p(y|\tilde{\mathbf{u}}_j)$  in  $\mathcal{L}_u$  should be:

$$\begin{aligned} p_{tr}(y|\mathbf{x}_i) &= \text{softmax}(f(\mathbf{x}_i) + \ln p_{tr}(y)) \\ p_{tr}(y|\tilde{\mathbf{u}}_j) &= \text{softmax}(f(\tilde{\mathbf{u}}_j) + \ln p_{tr}(y)). \end{aligned}$$

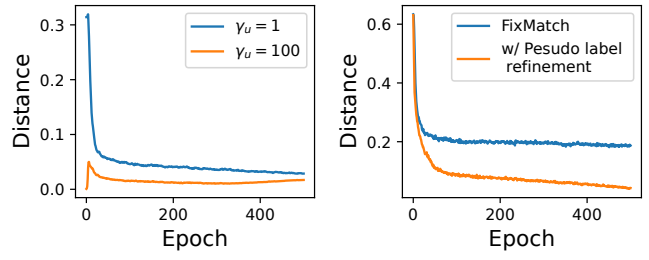
This loss can be considered as an extension of the logit adjustment loss (Menon et al. 2020) or balanced softmax (Ren et al. 2020) in imbalanced SSL. As the class distribution of unlabeled data is unknown and varies during training, we need to estimate  $p_{tr}(y)$ . We use the class distribution of the participating training samples in the most recent  $T$  iterations as an estimate of  $p_{tr}(y)$ . Additionally, since there are weights associated with the loss in  $\mathcal{L}_u$ , we treat it as a form of undersampling. Consequently, the number of samples for class  $y$  in a batch is given by:

$$\text{count}(y) = \sum_{\mathbf{x}_i \in \mathcal{B}_l} \mathbf{I}(y_i = y) + \sum_{\mathbf{u}_j \in \mathcal{B}_u} \lambda \mathcal{M} \cdot \mathbf{I}(\hat{y}_j = y).$$

To investigate the correlation between  $p_{tr}(y)$  and the true distribution (representing the class distribution of  $\mathcal{D}_l \cup \mathcal{D}_u$ ), we monitored the L2 distance between  $p_{tr}(y)$  and the true distribution during the training process. Figure 3a shows the results for both cases: when the distributions of labeled and unlabeled samples are consistent and when they are inconsistent. It can be observed that when the distributions are consistent,  $p_{tr}(y)$  maintains a small distance from the true distribution. When the distributions are inconsistent,  $p_{tr}(y)$  continuously approaches the true distribution.

### Pseudo-Label Refinement

In the previous section, we utilized an estimate of the marginal distribution  $p_{tr}(y)$  of the training data to learn a class-balanced model. However, *is it optimal to use this model to generate pseudo-labeled data for training during the training process?*



(a) Model bias correction

(b) Pseudo-label refinement

Figure 3: (a) L2 Distance between  $p_{tr}(y)$  and true distribution in consistent ( $\gamma_u = 100$ ) and inconsistent ( $\gamma_u = 1$ ) setting when applying model bias correction. (b) L2 distance between the pseudo-label distribution and the unlabeled distribution with pseudo-label refinement.

We conducted a parametric decomposition of Eq. (4) and (5), taking into further consideration the parameters  $\theta_t$  of the model at the  $t$ -th iteration:

$$\begin{aligned} p_{bal}(y|\mathbf{x}; \theta_t) &\propto p(\mathbf{x}|y; \theta_t) \cdot p_{tr}(y|\theta_t)/p_{tr}(y|\theta_t) \\ &\propto p_{tr}(y|\mathbf{x}; \theta_t)/p_{tr}(y|\theta_t) \\ &\propto \text{softmax}(f(\mathbf{x}) + \ln p_{tr}(y) - \ln p_{tr}(y|\theta_t)), \end{aligned}$$

where  $p_{bal}(y|\mathbf{x}; \theta_t)$  represents the posterior probability under the uniform distribution prior for a given model parameters  $\theta_t$ . It takes into account the model parameters more explicitly than  $p_{ev}(y|\mathbf{x})$ . To compute  $p_{bal}(y|\mathbf{x}; \theta_t)$ , a crucial step is to estimate the class prior  $p_{tr}(y|\theta_t)$  under the current parameters. Referring to Hong et al. (2023), we can estimate  $p_{tr}(y|\theta_t)$  through the model's outputs on the training samples:

$$\begin{aligned} p_{tr}(y|\theta_t) &= \mathbf{E}_{\mathbf{x}_i \in \mathcal{D}_{tr}} p_{tr}(y|\mathbf{x}_i; \theta_t) \\ &= \mathbf{E}_{\mathbf{x}_i \in \mathcal{D}_{tr}} (p_{ev}(y|\mathbf{x}_i; \theta_t) \cdot p_{tr}(y)), \end{aligned}$$

where  $\mathcal{D}_{tr}$  represents the dataset composed of participating training samples. Let  $d_y(\theta_t) = \ln p_{tr}(y) - \ln p_{tr}(y|\theta_t)$ , we have:

$$\begin{aligned} d_y(\theta_t) &= \ln p_{tr}(y) - \ln p_{tr}(y|\theta_t) \\ &= \ln \left( \frac{p_{tr}(y)}{\mathbf{E}_{\mathbf{x}_i \in \mathcal{D}_{tr}} (\text{softmax}(f(\mathbf{x}_i) + \ln p_{tr}(y)))} \right) \\ &= -\ln \mathbf{E}_{\mathbf{x}_i \in \mathcal{D}_{tr}} \left( \frac{e^{f^y(\mathbf{x}_i)}}{\sum_{k=1}^K e^{f^k(\mathbf{x}_i) + \ln p_{tr}(k)}} \right). \end{aligned}$$

Utilizing  $d_y(\theta_t)$  to modify  $f(\mathbf{x})$  allows us to acquire pseudo-labels under the current parameters that mitigate class bias. However, an accurate estimation of  $d_y(\theta_t)$  necessitates considering the entire dataset. To ensure stable and efficient estimation of  $d_y(\theta_t)$ , we devised a momentum mechanism that leverages expectations computed over each iteration for momentum updates:

$$d_y(\theta_{t+1}) = m \cdot d_y(\theta_t) + (1 - m) \cdot d'_y(\theta_t),$$

where  $m \in [0, 1]$  is a momentum coefficient,  $d'_y(\theta_t)$  is computed utilizing data from the  $t$ -th iteration. Consequently, the refined pseudo-label of  $\mathbf{u}_j$  can be expressed as follows:

$$\hat{y}_j^{r_e} = \arg \max (\text{softmax}(f(\mathbf{u}_j) + d_y(\theta_{t+1}))).$$

Algorithm	CIFAR10-LT				CIFAR10-LT			
	$N_1 = 1500, M_1 = 3000$		$N_1 = 500, M_1 = 4500$		$N_1 = 1500, M_1 = 3000$		$N_1 = 500, M_1 = 4500$	
	$\gamma_l = 100$ $\gamma_u = 100$	$\gamma_l = 150$ $\gamma_u = 150$	$\gamma_l = 100$ $\gamma_u = 100$	$\gamma_l = 150$ $\gamma_u = 150$	$\gamma_l = 100$ $\gamma_u = 1$	$\gamma_l = 100$ $\gamma_u = 1/100$	$\gamma_l = 100$ $\gamma_u = 1$	$\gamma_l = 100$ $\gamma_u = 1/100$
Supervised	61.9 ± 0.41	58.2 ± 0.29	47.3 ± 0.95	44.2 ± 0.33	61.9 ± 0.41	61.9 ± 0.41	47.3 ± 0.95	47.3 ± 0.95
FixMatch	77.5 ± 1.32	72.4 ± 1.03	67.8 ± 1.13	62.9 ± 0.36	81.5 ± 1.15	71.8 ± 1.70	73.0 ± 3.81	62.5 ± 0.49
w / DARP	77.8 ± 0.63	73.6 ± 0.73	74.5 ± 0.78	67.2 ± 0.32	84.6 ± 0.34	80.0 ± 0.93	82.5 ± 0.75	70.1 ± 0.22
w / CReST+	78.1 ± 0.42	73.7 ± 0.34	76.3 ± 0.86	67.5 ± 0.45	86.4 ± 0.42	72.9 ± 2.00	82.2 ± 1.53	62.9 ± 1.39
w / ABC	83.8 ± 0.36	80.1 ± 0.45	78.9 ± 0.82	66.5 ± 0.78	82.8 ± 0.61	77.9 ± 0.86	73.1 ± 0.33	66.6 ± 0.39
w / DASO	79.1 ± 0.75	75.1 ± 0.77	76.0 ± 0.37	70.1 ± 1.81	88.8 ± 0.59	80.3 ± 0.65	86.6 ± 0.84	71.0 ± 0.95
w / L2AC	82.1 ± 0.57	77.6 ± 0.53	76.1 ± 0.45	70.2 ± 0.63	89.5 ± 0.18	82.2 ± 1.23	87.1 ± 0.23	73.1 ± 0.28
w / TCBC (ours)	<b>84.0</b> ± 0.55	<b>80.4</b> ± 0.58	<b>80.3</b> ± 0.45	<b>75.2</b> ± 0.32	<b>92.8</b> ± 0.42	<b>85.9</b> ± 0.17	<b>92.4</b> ± 0.29	<b>79.9</b> ± 0.41

Table 1: Comparison of accuracy (%) on CIFAR10-LT and under  $\gamma_l = \gamma_u$  and  $\gamma_l \neq \gamma_u$  setting. We set  $\gamma_l$  to 100 and 150 for CIFAR10-LT. We use bold to mark the best results.

Algorithm	CIFAR100-LT				STL10			
	$N_1 = 150, M_1 = 300$		$N_1 = 150, M_1 = 300$		$N_1 = 150, M_1 = 100k$		$N_1 = 450, M_1 = 100k$	
	$\gamma_l = 10$ $\gamma_u = 10$	$\gamma_l = 20$ $\gamma_u = 20$	$\gamma_l = 10$ $\gamma_u = 1$	$\gamma_l = 10$ $\gamma_u = 1/10$	$\gamma_l = 10$ $\gamma_u = N/A$	$\gamma_l = 20$ $\gamma_u = N/A$	$\gamma_l = 10$ $\gamma_u = N/A$	$\gamma_l = 20$ $\gamma_u = N/A$
Supervised	46.9 ± 0.22	41.2 ± 0.15	46.9 ± 0.22	46.9 ± 0.22	40.9 ± 4.11	36.4 ± 3.12	60.1 ± 5.22	50.6 ± 4.15
FixMatch	46.5 ± 0.06	50.7 ± 0.25	58.6 ± 0.73	57.6 ± 0.53	51.6 ± 2.32	47.6 ± 4.87	72.4 ± 0.71	64.0 ± 2.27
w / DARP	58.1 ± 0.44	52.2 ± 0.66	55.8 ± 0.30	51.6 ± 0.83	66.9 ± 1.66	59.9 ± 2.17	75.6 ± 0.45	72.3 ± 0.60
w / CReST+	57.4 ± 0.18	52.1 ± 0.21	59.3 ± 0.88	56.6 ± 0.23	61.2 ± 1.27	57.1 ± 3.67	71.5 ± 0.96	68.5 ± 1.88
w / ABC	59.1 ± 0.21	53.7 ± 0.55	61.3 ± 0.53	59.8 ± 0.63	69.6 ± 0.78	64.1 ± 0.67	78.6 ± 0.86	66.9 ± 0.98
w / DASO	59.2 ± 0.35	52.9 ± 0.42	59.8 ± 0.32	59.7 ± 0.43	70.1 ± 1.19	65.7 ± 1.78	78.4 ± 0.80	75.3 ± 0.44
w / L2AC	57.8 ± 0.19	52.6 ± 0.13	61.3 ± 0.44	59.2 ± 0.36	75.1 ± 0.44	72.3 ± 0.22	79.9 ± 0.52	77.0 ± 0.65
w / TCBC (ours)	<b>59.4</b> ± 0.28	<b>53.9</b> ± 0.72	<b>63.2</b> ± 0.65	<b>59.9</b> ± 0.42	<b>77.6</b> ± 0.93	<b>74.9</b> ± 1.42	<b>84.5</b> ± 0.52	<b>82.6</b> ± 1.23

Table 2: Comparison of accuracy (%) on CIFAR100-LT and STL10 under  $\gamma_l = \gamma_u$  and  $\gamma_l \neq \gamma_u$  setting. We set  $\gamma_l$  to 10 and 20 for them. We use bold to mark the best results.

Fixing  $p_{tr}(y)$  at  $1/K$  leads the algorithm to degenerate into a process that exclusively incorporates pseudo-label refinement into the FixMatch. We conducted a comparative analysis between the approach that exclusively employs pseudo-label refinement and the original FixMatch, aiming to explore the characteristics of pseudo-label refinement. Experimental trials were carried out utilizing unlabeled samples distributed uniformly, aligning with the conditions depicted in Figure 1a. Figure 3b visualizes the L2 distance between the distribution of pseudo-labels and the uniform distribution. Clearly, with the advancement of training, the refined distribution of pseudo-labels gradually converges toward the uniform distribution. Our proposed approach successfully mitigates the class bias present in pseudo-labels.

In summary, Figure 2 presents the overall training procedure for TCBC. The labeled loss in Eq. (1) and unlabeled loss in Eq. (2) are given by:

$$\mathcal{L}_s = \frac{1}{|\mathcal{B}_l|} \sum_{\mathbf{x}_i \in \mathcal{B}_l} \mathcal{H}(y_i, p_{tr}(y|\mathbf{x}_i)),$$

$$\mathcal{L}_u = \frac{1}{|\mathcal{B}_u|} \sum_{\mathbf{u}_j \in \mathcal{B}_u} \mathcal{M} \cdot \mathcal{H}(\hat{y}_j^{re}, p_{tr}(y|\tilde{\mathbf{u}}_j)).$$

## Experiments

This section presents a comprehensive evaluation of our algorithm’s performance within the context of imbalanced SSL in classification problems.

### Experimental Setup

**Datasets.** We conduct experiments on three benchmarks including CIFAR10, CIFAR100 (Krizhevsky, Hinton et al. 2009) and STL10 (Coates, Ng, and Lee 2011), which are commonly used in imbalance learning and SSL task. Results on real-world dataset, SUN-397, are also given in appendix. To validate the effectiveness of TCBC, we evaluate TCBC under various ratio of class imbalance. For imbalance types, we adopt long-tailed (LT) imbalance by exponentially decreasing the number of samples from the largest to the smallest class. Following (Lee, Shin, and Kim 2021), we denote the amount of samples of head class in labeled data and unlabeled data as  $N_1$  and  $M_1$  respectively. The imbalance ratio for the labeled data and unlabeled data is defined as  $\gamma_l$  and  $\gamma_u$ , which can vary independently. We have  $N_k = N_1 \cdot \gamma_l^{\epsilon_k}$  and  $M_k = M_1 \cdot \gamma_u^{\epsilon_k}$ , where  $\epsilon_k = \frac{k-1}{K-1}$ .

**Baseline Methods.** For supervised learning, we train network using cross-entropy loss with only labeled data. For

semi-supervised learning, we compare the performance of TCBC with FixMatch (Sohn et al. 2020), which do not consider class imbalance. To have a comprehensive comparison, we combine several re-balancing algorithms with FixMatch, including DARP (Kim et al. 2020), CReST (Wei et al. 2021), ABC (Lee, Shin, and Kim 2021), DASO (Oh, Kim, and Kweon 2022) and L2AC (Wang et al. 2022).

**Training and Evaluation.** We train Wide ResNet-28-2 (WRN28-2) on CIFAR10-LT, CIFAR100-LT and STL10-LT as a backbone. We evaluate the performance of TCBC using an EMA network, where parameters are updating via exponential moving average every steps, following Oh, Kim, and Kweon (2022). We measure the top-1 accuracy on test data and finally report the median of accuracy values of the last 20 epochs following Berthelot et al. (2019). Each set of experiments was conducted three times. Additional experimental details are provided in the appendix.

## Results

**In the Case of  $\gamma_l = \gamma_u$ .** We initiate our investigation by conducting experiments in the scenario where  $\lambda_l = \lambda_u$ . Our evaluation of the proposed TCBC approach is exhaustive, encompassing a comprehensive comparative analysis against various recent state-of-the-art methods. These methods include DARP (Kim et al. 2020), CReST+ (Wei et al. 2021), ABC (Lee, Shin, and Kim 2021), DASO (Oh, Kim, and Kweon 2022), and L2AC (Wang et al. 2022). Further details about these methods are provided in appendix.

The main results on the CIFAR-10 dataset are shown in Table 1. It is evident that across various dataset sizes and imbalance ratios, our approach (TCBC) substantially enhances the performance of FixMatch. Moreover, our TCBC consistently surpasses all the compared approaches in these settings, even when they are designed with the assumption of shared class distributions between labeled and unlabeled data. For instance, considering the highly imbalanced scenario with  $\gamma_l = \gamma_u = 150$ , our TCBC achieves improvements of 2.8% and 5.0% in situations where  $N_1 = 1500$ ,  $M_1 = 3000$  and  $N_1 = 500$ ,  $M_1 = 4000$ , respectively, compared to the L2AC.

To facilitate a more comprehensive comparison, we also conducted an evaluation of TCBC using the CIFAR-100 dataset. As illustrated in Table 2, our TCBC exhibits a more competitive performance in comparison to the state-of-the-art methods ABC, DASO and L2AC.

**In the Case of  $\gamma_l \neq \gamma_u$ .** In practical datasets, the distribution of unlabeled data might significantly differs from that of labeled data. Therefore, we explore uniform and reversed class distributions, such as setting  $\gamma_u$  to 1 or 1/100 for CIFAR10-LT. In the case of the STL10-LT dataset, as the ground-truth labels of the unlabeled data are unknown, we can only control the imbalance ratio of the labeled data. We present the summarized results in Table 2.

Our method demonstrates superior performance when confronted with inconsistent class distributions in unlabeled data. For instance, when  $\gamma_u$  is set to 1 and 1/100 on CIFAR10-LT, TCBC achieves absolute performance gains of 19.4% and 17.4% respectively compared to FixMatch.

	$\gamma_u = 100$	$\gamma_u = 1$	$\gamma_u = 1/100$
FixMatch	67.8	73.0	62.5
w / model bias correction	79.2	92.3	79
w / pseudo-label refinement	77.8	92.1	76.8
TCBC	80.3	92.4	79.9

Table 3: Ablation studies of our proposed TCBC algorithm.

$\gamma_u$	DASO	TCBC	$\gamma_u$	DASO	TCBC
100	79.1	84.0(+4.9)	1/100	80.3	85.9(+5.6)
75	80.7	84.3(+3.6)	1/75	82.1	86.8(+4.7)
50	82.9	85.2(+2.3)	1/50	81.6	87.6(+6.0)
25	85.4	88.9(+3.5)	1/25	84.0	86.0(+2.0)
1	88.8	92.8(+4.0)	Avg	82.8	86.8(+4.0)

Table 4: Results on more realistic imbalanced SSL settings.

Similarly, on CIFAR100-LT, our method consistently outperforms compared methods. Even in the case of STL10-LT where the distribution of unlabeled data is unknown, TCBC attains the best results with an average accuracy gain of 3.2% compared to L2AC. These empirical results across the three datasets with unknown class distributions of unlabeled data validate the effectiveness of TCBC in leveraging unlabeled data to mitigate the negative impact of class imbalance.

## Ablation Study

To explore the contributions of each key component in TCBC, we conducted a series of ablation studies. We set  $N_1$  to 500 and  $M_1$  to 4000, and performed experiments on CIFAR-10 with various settings of  $\gamma_l = 100$ . As shown in Table 3, it is evident that using either model bias correction or pseudo-label refinement alone can significantly enhance the performance of FixMatch. This underscores the effectiveness of the two components in our approach.

However, pseudo-label refinement performs poorly when  $\gamma_u = 100$  and  $\gamma_u = 1/100$ , mainly due to the imbalanced distribution of  $\mathcal{D}_l \cup \mathcal{D}_u$ , which introduces class bias into the trained model. Model bias correction effectively addresses this issue. Similarly, pseudo-label refinement also enhances the effectiveness of model bias correction.

## Discussion

**Can our method adapt to unlabeled samples with different class distributions?** To assess the effectiveness of our approach across different distributions of unlabeled samples, we conducted experiments on CIFAR-10-LT. The labeled sample distribution  $\gamma_l$  was maintained at a constant value of 100, while systematically modifying the imbalance ratio ( $\gamma_u$ ) of unlabeled data. In this study, we configured  $N_1$  as 1500 and  $M_1$  as 3000, and we conducted a comparative analysis of our method against the performance of DASO. The outcomes of the experiments are illustrated in Table 4. Notably, our TCBC method consistently demonstrated superior performance compared to DASO across all test scenarios, achieving an average performance increase of 4%. These

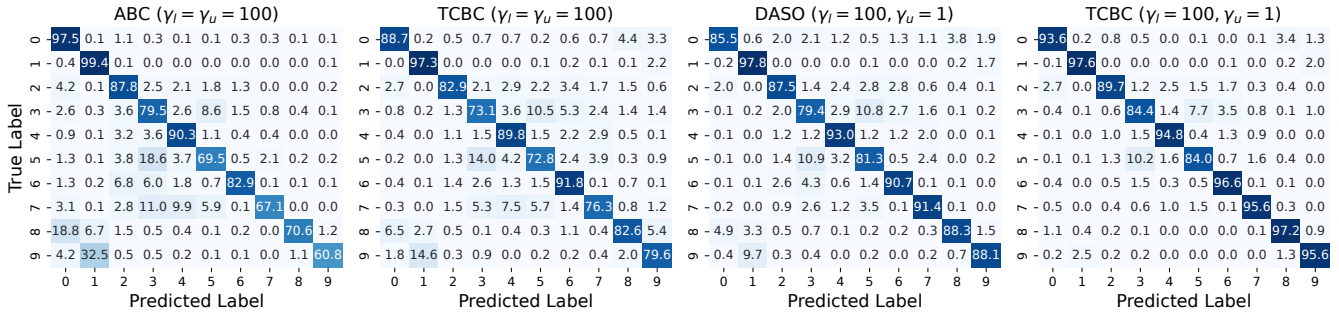


Figure 4: Confusion matrices of ABC, DASO and ours on CIFAR-10 under the imbalance ratio  $\gamma_u = 100$  and  $\gamma_u = 1$ .

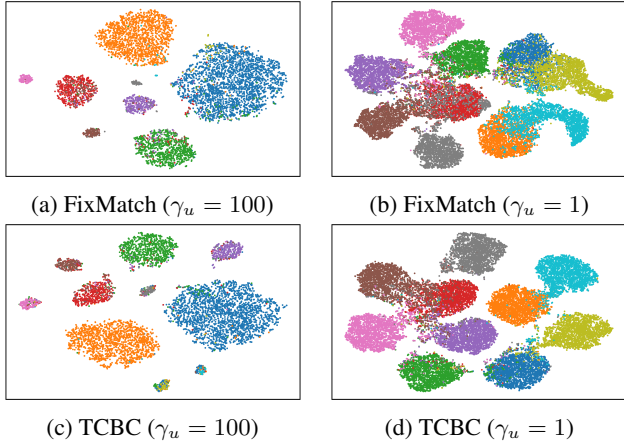


Figure 5: The t-SNE visualization of the unlabeled set on CIFAR-10-LT dataset.

results clearly demonstrate that our method can effectively adapt to imbalanced SSL environments in the real world.

**How does our method perform on samples of different frequencies?** A comparative analysis was performed on our TCBC in comparison to various configurations of ABC and DASO. Figure 4 shows the confusion matrices of the models obtained by different algorithms on the test set when  $N_1$  is set to 1500 and  $M_1$  is set to 3000. The two confusion matrices on the left depict the test results of ABC and TCBC when the labeled and unlabeled class distributions are consistent. From the recall of the minority class samples, it is evident that ABC exhibits bias across different classes. However, our approach effectively mitigates such biases. Under the settings of  $\gamma_l = 100$  and  $\gamma_u = 1$ , both our method and the model learned by DASO reduced the model’s class bias. However, our method outperforms DASO on both majority and minority classes. This observation indicates that our TCBC has successfully learned a model without class bias.

**Have our methods learned better features?** We further visualized the features of unlabeled samples using t-SNE (Van der Maaten and Hinton 2008) in the same setting as depicted in Figure 4. As shown in Figure 5, compared to FixMatch, TCBC has learned more discriminative features. For instance, in the case of  $\gamma_u = 100$ , FixMatch exhibits

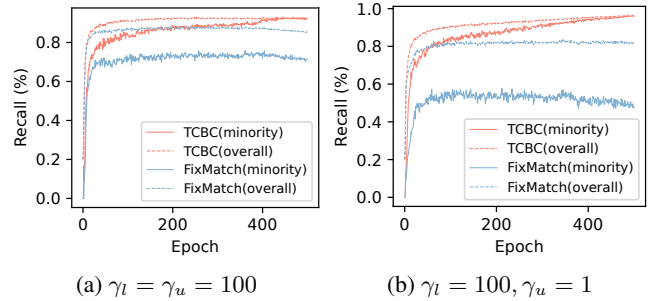


Figure 6: Train curves for the recall of pseudo-labels.

only 8 clusters, while our method demonstrates all 10 clusters. This high-quality feature representation reflects the effectiveness of our method in learning an unbiased model and refining pseudo-labels.

**How does the TCBC enhance model performance?** We examined the changes in recall of minority and all classes of unlabeled samples under the settings of Figure 4 for both FixMatch and TCBC methods. As shown in Figure 6, it is evident that in FixMatch, the recall of the minority class quickly plateaus and exhibits a significant disparity from the recall of all classes. In contrast, in TCBC, the recall of the minority class continues to increase and approaches the average recall. This indicates that TCBC ensures equitable treatment of pseudo-labeling for minority classes throughout the process, aligning with our motivation.

## Conclusion

In this work, we address the model bias and pseudo-label bias in imbalanced SSL through the introduction of a novel twice correction approach. For tackling model bias, we propose the utilization of an estimate of the training sample class distribution to rectify the model’s learning objectives onto an unbiased posterior probability. To address pseudo-label bias, we refine a better set of pseudo-labels by estimating the class bias under the current parameters during the training process. Extensive experimental results demonstrate that our method outperforms existing approaches.

## Acknowledgments

This work is supported by the National Science Foundation of China (61921006). We would like to thank Xin-chun Li, and the anonymous reviewers for their helpful discussions and support.

## References

- Berthelot, D.; Carlini, N.; Goodfellow, I.; Papernot, N.; Oliver, A.; and Raffel, C. A. 2019. Mixmatch: A holistic approach to semi-supervised learning. *Advances in neural information processing systems*, 32.
- Buda, M.; Maki, A.; and Mazurowski, M. A. 2018. A systematic study of the class imbalance problem in convolutional neural networks. *Neural networks*, 106: 249–259.
- Byrd, J.; and Lipton, Z. 2019. What is the effect of importance weighting in deep learning? In *International conference on machine learning*, 872–881. PMLR.
- Cao, K.; Wei, C.; Gaidon, A.; Arechiga, N.; and Ma, T. 2019. Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in neural information processing systems*, 32.
- Chu, P.; Bian, X.; Liu, S.; and Ling, H. 2020. Feature space augmentation for long-tailed data. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIX 16*, 694–710. Springer.
- Coates, A.; Ng, A.; and Lee, H. 2011. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 215–223. JMLR Workshop and Conference Proceedings.
- Cui, Y.; Jia, M.; Lin, T.-Y.; Song, Y.; and Belongie, S. 2019. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9268–9277.
- Drummond, C.; Holte, R. C.; et al. 2003. C4. 5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling. In *Workshop on learning from imbalanced datasets II*, volume 11, 1–8.
- Guo, L.-Z.; and Li, Y.-F. 2022. Class-imbalanced semi-supervised learning with adaptive thresholding. In *International Conference on Machine Learning*, 8082–8094. PMLR.
- Hong, F.; Yao, J.; Zhou, Z.; Zhang, Y.; and Wang, Y. 2023. Long-tailed partial label learning via dynamic rebalancing. *arXiv preprint arXiv:2302.05080*.
- Huang, C.; Li, Y.; Loy, C. C.; and Tang, X. 2016. Learning deep representation for imbalanced classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5375–5384.
- Kim, J.; Hur, Y.; Park, S.; Yang, E.; Hwang, S. J.; and Shin, J. 2020. Distribution aligning refinery of pseudo-label for imbalanced semi-supervised learning. *Advances in neural information processing systems*, 33: 14567–14579.
- Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images.
- Lee, D.-H.; et al. 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, 896.
- Lee, H.; Shin, S.; and Kim, H. 2021. Abc: Auxiliary balanced classifier for class-imbalanced semi-supervised learning. *Advances in Neural Information Processing Systems*, 34: 7082–7094.
- Li, L.; Zhan, D.-c.; and Li, X.-c. 2022. Aligning model outputs for class imbalanced non-IID federated learning. *Machine Learning*, 1–24.
- Li, Y.-F.; and Liang, D.-M. 2019. Safe semi-supervised learning: a brief introduction. *Frontiers of Computer Science*, 13: 669–676.
- Menon, A. K.; Jayasumana, S.; Rawat, A. S.; Jain, H.; Veit, A.; and Kumar, S. 2020. Long-tail learning via logit adjustment. *arXiv preprint arXiv:2007.07314*.
- Oh, Y.; Kim, D.-J.; and Kweon, I. S. 2022. 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*, 9786–9796.
- Pouyanfar, S.; Tao, Y.; Mohan, A.; Tian, H.; Kaseb, A. S.; Gauen, K.; Dailey, R.; Aghajanzadeh, S.; Lu, Y.-H.; Chen, S.-C.; et al. 2018. Dynamic sampling in convolutional neural networks for imbalanced data classification. In *2018 IEEE conference on multimedia information processing and retrieval (MIPR)*, 112–117. IEEE.
- Ren, J.; Yu, C.; Ma, X.; Zhao, H.; Yi, S.; et al. 2020. Balanced meta-softmax for long-tailed visual recognition. *Advances in neural information processing systems*, 33: 4175–4186.
- Sohn, K.; Berthelot, D.; Carlini, N.; Zhang, Z.; Zhang, H.; Raffel, C. A.; Cubuk, E. D.; Kurakin, A.; and Li, C.-L. 2020. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33: 596–608.
- Tao, B.; Li, L.; Li, X.-C.; and Zhan, D.-C. 2023. CLAF: Contrastive Learning with Augmented Features for Imbalanced Semi-Supervised Learning. *arXiv preprint arXiv:2312.09598*.
- Van der Maaten, L.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11).
- Van Engelen, J. E.; and Hoos, H. H. 2020. A survey on semi-supervised learning. *Machine learning*, 109(2): 373–440.
- Wang, R.; Jia, X.; Wang, Q.; Wu, Y.; and Meng, D. 2022. Imbalanced Semi-supervised Learning with Bias Adaptive Classifier. In *The Eleventh International Conference on Learning Representations*.
- Wang, Y.-X.; Ramanan, D.; and Hebert, M. 2017. Learning to model the tail. *Advances in neural information processing systems*, 30.
- Wei, C.; Sohn, K.; Mellina, C.; Yuille, A.; and Yang, F. 2021. 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*, 10857–10866.

- Wei, T.; and Gan, K. 2023. Towards Realistic Long-Tailed Semi-Supervised Learning: Consistency Is All You Need. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3469–3478.
- Wei, T.; Liu, Q.-Y.; Shi, J.-X.; Tu, W.-W.; and Guo, L.-Z. 2022. Transfer and share: semi-supervised learning from long-tailed data. *Machine Learning*, 1–18.
- Yang, Y.; Fu, Z.-Y.; Zhan, D.-C.; Liu, Z.-B.; and Jiang, Y. 2019a. Semi-supervised multi-modal multi-instance multi-label deep network with optimal transport. *IEEE Transactions on Knowledge and Data Engineering*, 33(2): 696–709.
- Yang, Y.; Wang, K.-T.; Zhan, D.-C.; Xiong, H.; and Jiang, Y. 2019b. Comprehensive semi-supervised multi-modal learning. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 4092–4098.
- Ye, H.-J.; Chen, H.-Y.; Zhan, D.-C.; and Chao, W.-L. 2020. Identifying and compensating for feature deviation in imbalanced deep learning. *arXiv preprint arXiv:2001.01385*.
- Ye, H.-J.; Zhan, D.-C.; and Chao, W.-L. 2021. Procrustean training for imbalanced deep learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, 92–102.
- Yin, X.; Yu, X.; Sohn, K.; Liu, X.; and Chandraker, M. 2019. Feature transfer learning for face recognition with under-represented data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 5704–5713.