

# Sampling Control for Imbalanced Calibration in Semi-Supervised Learning

Senmao Tian, Xiang Wei, Shunli Zhang\*

School of Software Engineering, Beijing Jiaotong University  
smtian1204@gmail.com, slzhang@bjtu.edu.cn

## Abstract

Class imbalance remains a critical challenge in semi-supervised learning (SSL), especially when distributional mismatches between labeled and unlabeled data lead to biased classification. Although existing methods address this issue by adjusting logits based on the estimated class distribution of unlabeled data, they often handle model imbalance in a coarse-grained manner, conflating data imbalance with bias arising from varying class-specific learning difficulties. To address this issue, we propose a unified framework, SC-SSL, which suppresses model bias through decoupled sampling control. During training, we identify the key variables for sampling control under ideal conditions. By introducing a classifier with explicit expansion capability and adaptively adjusting sampling probabilities across different data distributions, SC-SSL mitigates feature-level imbalance for minority classes. In the inference phase, we further analyze the weight imbalance of the linear classifier and apply post-hoc sampling control with an optimization bias vector to directly calibrate the logits. Extensive experiments across various benchmark datasets and distribution settings validate the consistency and state-of-the-art performance of SC-SSL.

**Code** — <https://github.com/Sheldon04/SC-SSL>

## Introduction

Semi-supervised learning (SSL) (Berthelot et al. 2019b; Miyato et al. 2017; Tarvainen and Valpola 2017) is a powerful strategy aimed at enhancing the generalization capabilities of deep neural networks (DNNs) by leveraging limited labeled data, particularly in scenarios where labeled samples are scarce. The core of most SSL methods (Sohn et al. 2020; Li, Xiong, and Hoi 2020; Berthelot et al. 2019a) lies in generating pseudo-labels for unlabeled data and selecting reliable labels for model training. However, real-world data often exhibits a long-tailed distribution (Li et al. 2022; Wang et al. 2020; Xiang and Ding 2020), leading models to primarily focus on common categories during training, which results in an imbalance of pseudo-labels. This phenomenon has given rise to class-imbalanced semi-supervised learning (CISSL). Traditional CISSL approaches (Fan, Dai, and

Schiele 2021; Lee, Shin, and Kim 2021; Wei et al. 2021) typically assume that the class distributions of labeled and unlabeled data are consistent, a premise that is often overly idealistic. In practical applications, class distributions may be inconsistent or even unknown (Lai et al. 2022; Kim et al. 2020; Oh, Kim, and Kweon 2021), especially when continuously collecting new data or processing data from different tasks. Variations in the distribution of unlabeled data significantly affect the performance of CISSL methods. To address these challenges, recent approaches (Wang et al. 2022a; Zheng et al. 2024; Li et al. 2024; Lee and Kim 2024) have focused on adapting to unknown and mismatched class distributions. Techniques such as ACR (Wei and Gan 2023), CPE (Ma et al. 2024), and SimPro (Du, Han, and Huang 2024) have been proposed to estimate the distance between distributions and train classifiers that adapt to specific or arbitrary distributions. However, due to the inherent limitations in the accuracy of classifiers, the adjusted pseudo-labels may not accurately reflect the true distribution of unlabeled data. Current methods (Wei and Gan 2023; Ma et al. 2024; Du, Han, and Huang 2024) improve model generalization by retaining a small portion of high-quality pseudo-labels to avoid confirmation bias (Wang et al. 2022b), resulting in a significant amount of unlabeled data not effectively contributing to training. Although some methods (Guo and Li 2022; Yu, Li, and Lee 2023) use adaptive thresholds to adjust the class probabilities for pseudo-labeling, these methods rely on the model’s prediction confidence. These methods overlook the importance of addressing imbalance from a finer-grained perspective, which results in feature-level model bias and renders logit adjustment ineffective in targeting the root cause. As a result, these methods still yield suboptimal performance.

To address this issue, we propose a novel method called SC-SSL (Sampling Control for Imbalanced Semi-supervised Learning), which suppresses model bias through decoupled sampling control. Starting from known information, we divide the labeled data into head and non-head classes based on their sample sizes. During training, instead of solely focusing on improving classification accuracy, SC-SSL emphasizes whether the learned features of non-head classes are balanced.

Without altering the training objective (i.e., the classification loss), we introduce an expansive classifier that focuses

\*Corresponding author.

Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

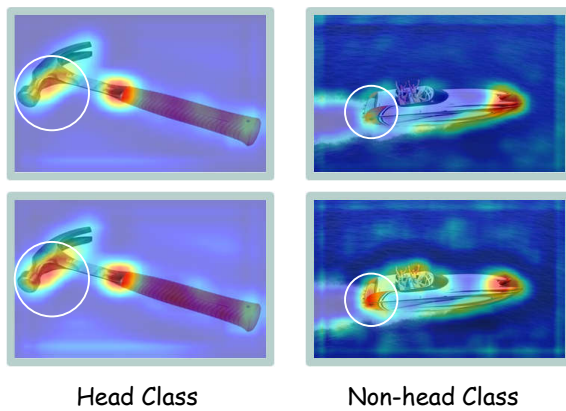


Figure 1: We select one image from a head class and one from a non-head class, and visualize their features using Grad-CAM (Selvaraju et al. 2016). The top row shows the results from training with uniform sampling, while the bottom row shows the results from SC-SSL. It can be observed that the attention regions for the non-head class are significantly improved, capturing more relevant features. To enhance visual clarity, we applied upsampling.

on feature learning for non-head classes. We incorporate the expansion–separation assumption and leverage the inherent denoising property of consistency regularization, allowing even noisy pseudo-labels to be used effectively. This balances the gradient contributions between head and non-head classes during training. To improve the sampling probability of non-head classes, we theoretically identify key control factors and associate them with the expansion factor in the assumption. This enables effective sampling regulation across various data distributions, allowing SC-SSL to mitigate the effects of data imbalance and suppress feature-level bias as fig. 1 shows.

However, the supervised losses of the three classifiers are influenced by their respective sampling factors, leading to additional imbalance. This imbalance arises from both data distribution and optimization dynamics. As a result, an imbalance remains in the linear layers due to optimization. At inference time, since feature imbalance has been alleviated, we can directly correct the optimization-induced imbalance in the classifier. We observe that the bias term of the linear layer reflects a combination of optimization imbalance and data imbalance. Under random sampling, head classes tend to have higher bias values; under the expansive classifier’s sampling, tail classes exhibit higher bias values; whereas the output classifier—trained with balanced sampling for inference—has bias terms that exclude data imbalance, thus isolating the optimization bias. Unlike the classifier weights, which require interaction with feature vectors and are harder to isolate, the bias term serves as a clean proxy for optimization imbalance. Therefore, we treat it as an optimization bias vector to calibrate the final logits, producing an overall unbiased prediction.

Additionally, SC-SSL also leverages this bias estimation prior to training to approximate the distribution of the unlabeled data. The resulting estimation is then used to initialize

pseudo-label sampling strategies for both the balanced and expansive classifiers.

We summarize our contributions as follows:

- We propose SC-SSL, a unified framework that mitigates both feature-level bias and logits-level bias in imbalanced settings through decoupled sampling control with an expansion classifier guided by the expansion–separation assumption.
- We theoretically analyze the key factors for controlling pseudo-label sampling under class imbalance, and dynamically adjust the sampling probabilities during training.
- Our proposed method achieves state-of-the-art performance on datasets such as CIFAR10-LT, CIFAR100-LT, STL10-LT, and ImageNet-127, across various distributions of unlabeled data.

## Related Work

**Semi-supervised Learning** In the realm of SSL, a subset of algorithms has risen to prominence by leveraging unlabeled data to enhance model performance. These algorithms operate on the principle of generating pseudo-labels, thereby establishing a self-training mechanism that refines the model iteratively. Advanced SSL strategies have taken this a step further by merging pseudo labeling with consistency regularization. This fusion promotes consistent predictions for the same image across different presentations, enhancing the resilience of DNNs. Among these, FixMatch (Sohn et al. 2020) and ReMixMatch (Berthelot et al. 2019a) stand out as particularly effective methods, achieving remarkable success in image recognition tasks and surpassing other SSL techniques in performance.

**Class Imbalanced Semi-supervised Learning** CISSL has garnered considerable interest due to its effectiveness in various real-world applications. CReST (Wei et al. 2021) emphasizes the importance of leveraging unlabeled instances from less common classes during its iterative self-training phase. Meanwhile, ABC (Lee, Shin, and Kim 2021) and CoSSL (Fan, Dai, and Schiele 2021) introduce an additional classifier to prevent bias in the training process towards any specific class. Although these methods have shown significant performance enhancements, they generally operate under the assumption that the class distributions for both labeled and unlabeled data are the same. In real-world scenarios, the distribution of unlabeled data may be unknown and mismatched. To tackle the challenges of realistic CISSL, DASO (Oh, Kim, and Kweon 2021) innovatively adjusts the ratio of linear and semantic pseudo-labels based on the unknown class distribution of unlabeled data. Conversely, ACR (Wei and Gan 2023) seeks to improve consistency regularization by establishing predefined distribution anchors, yielding encouraging outcomes. Following ACR, SimPro (Du, Han, and Huang 2024) takes a step further to model arbitrary unlabeled data distributions using probability theory. CPE (Ma et al. 2024) takes a different approach by training multiple experts, with each expert focusing on modeling a specific distribution. When generating

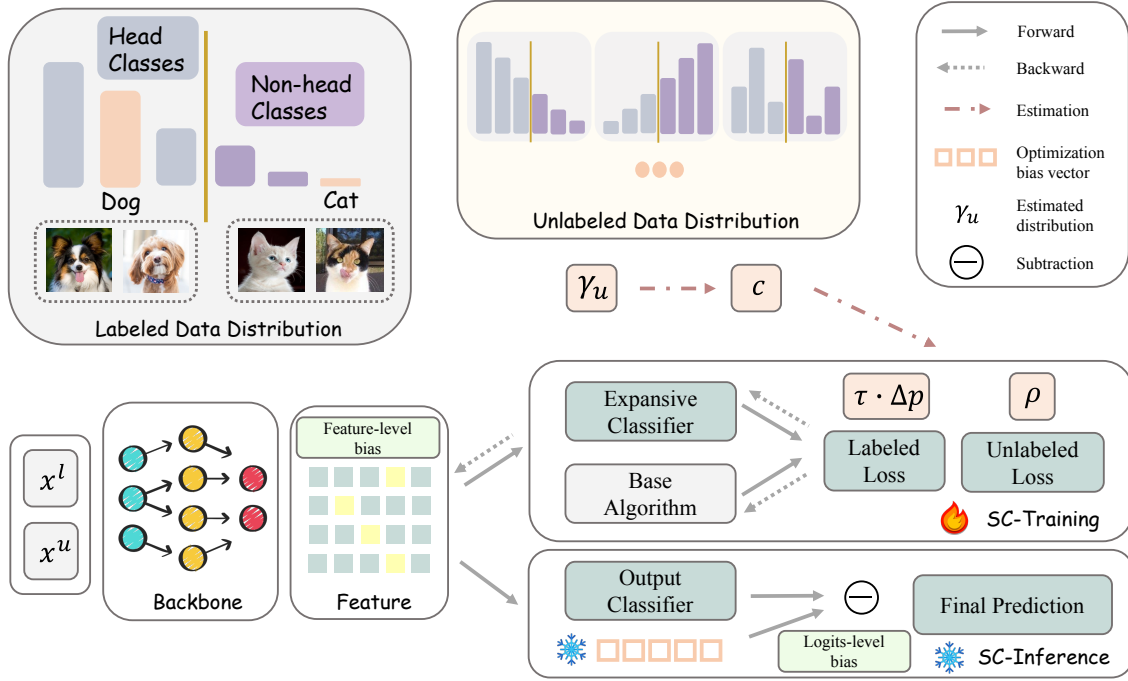


Figure 2: Illustration of the proposed framework. Details of the three key factors  $\gamma_u$ ,  $\Delta p$ , and  $\rho$  can be found in eq. (4). The utilization of these factors can be found in the Method section.

pseudo-labels, logit adjustments are made by ACR and Sim-Pro based on the estimated distribution, which makes the quality of the pseudo-labels heavily dependent on the accuracy of the estimates.

## Method

### Preliminaries

**Problem Setup** In the domain of CISSL, which is also referred to as long-tailed semi-supervised learning, we deal with a scenario that includes a labeled training dataset  $\mathcal{D}_l = \{(x_i^l, y_i^l)\}_{i=1}^N$  and an unlabeled training dataset  $\mathcal{D}_u = \{x_j^u\}_{j=1}^M$ . Both datasets share the same set of classes. Each data point  $x$  is associated with a class label  $y_i \in \{0, 1\}^K$ . For the labeled dataset  $\mathcal{D}_l$ , the number of samples for the  $k$ -th class is denoted by  $N_k$ , and for the unlabeled dataset  $\mathcal{D}_u$ , it is denoted by  $M_k$ . The number of samples in the most frequent class in the labeled and unlabeled datasets are  $N$  and  $M$ , respectively. Given that the distribution of classes in  $\mathcal{D}_l$  is typically imbalanced, classes are arranged in descending order, resulting in  $N_1 > N_2 > \dots > N_K$ . Here, the head classes are defined as those in the first half, while the non-head classes are those in the latter half. The imbalance ratio for the labeled dataset is defined as  $\gamma_l = \frac{N_1}{N_K}$ . The class distribution for the unlabeled dataset  $\mathcal{D}_u$  remains unknown due to the absence of labels. Although we do not need to make assumptions about the distribution of unlabeled data, we still utilize predefined anchor distributions for a better comparison with previous methods. We consider two general types of distributions: long-tail and Gaussian. For long-tail settings, we adopt consistent, uniform, and inverse distributions as the pre-defined anchors with an imbalance ratio

$\gamma_u$ . In the inverse setting, the sample count for each class is sorted in ascending order, meaning it is the opposite of the labeled data. For Gaussian scenes, we have normal and inverse settings. The mean and variance of the Gaussian distribution are set to  $(K - 1)/2$  and  $K/6$ , respectively.

**Notations**  $\mathcal{D}_u$  is the set of samples drawn from distribution  $P$ , reflecting the characteristics of the data in the input space  $\mathcal{X}$ . We let  $B : \mathcal{X} \rightarrow \mathbb{R}^Q$  denote a learned backbone mapping function (e.g., the continuous features output by a neural network),  $F : \mathbb{R}^Q \rightarrow [K]$  the scoring function from features to logits,  $G : \mathcal{X} \rightarrow [K]$  the discrete labels from  $G(x) \triangleq \operatorname{argmax}_i(\sigma(F(B(x))_i))$  and  $\sigma(\cdot)$  denotes the Softmax function. For different classification heads,  $F_b$  denotes the output classifier and  $F_e$  denotes the expansive classifier, respectively. They share the backbone  $B$ .

### Motivation

When the class distributions of labeled and unlabeled datasets are misaligned, many existing approaches (Wei and Gan 2023; Du, Han, and Huang 2024; Ma et al. 2024) adopt conservative strategies, avoiding the use of pseudo-labels for tail classes due to concerns about noise and label errors. However, recent theoretical perspectives on self-training suggest that consistency regularization can still extract meaningful supervision from noisy pseudo-labels, as long as certain structural assumptions about the data distribution hold.

Specifically, the expansion assumption implies that even if only a small fraction of samples from a class are confidently predicted, their surrounding regions in the feature space are likely to contain samples from the same class.

This implies that well-initialized class-conditional feature clusters can propagate label information to nearby unlabeled samples, thereby enhancing the balance of feature representations across classes. Formally, the class-conditional distribution  $P_i$  is said to satisfy an  $(a, c)$ -expansion if for any subset  $V \subseteq \mathcal{X}$  with  $P_i(V) \leq a$ , the following holds:

$$P_i(\mathcal{N}(V)) \geq \min\{cP_i(V), 1\}, \quad (1)$$

where  $\mathcal{N}(V)$  denotes a neighborhood around  $V$  in feature space.

Complementary to this, the separation assumption ensures local consistency under input perturbation. It requires that augmented versions of the same sample are predicted consistently:

$$R_B(G) \triangleq \mathbb{E}_P [\mathbb{I}(\exists x' \in \mathcal{A}(x) \text{ s.t. } G(x') \neq G(x))] \leq \mu, \quad (2)$$

where  $\mathcal{A}(x)$  defines the augmentation neighborhood of  $x$ , and  $\mu$  reflects the classifier's robustness to such perturbations.

Under these assumptions, the classifier's generalization error is upper-bounded by the quality of the pseudo-label generator and its robustness to augmentations. Specifically, the pseudo-label denoising theorem states that the error of the classifier  $\hat{G}$  satisfies:

$$\text{Err}(\hat{G}) \leq \frac{2c}{c-3}\mu, \quad (3)$$

where  $c > 3$  denotes the expansion factor. The detailed derivation can be found in the Appendix of (Wei et al. 2020). We adopt the FixMatch (Sohn et al. 2020) setting, using  $\hat{G}$  directly as the pseudo-labeler. This implies that even if pseudo-labels are noisy, as long as the model maintains low perturbation sensitivity (small  $\mu$ ) and high feature-local consistency (large  $c$ ), it can still facilitate class-balanced feature learning, especially for tail classes.

## The Design of SC-SSL

**Sampling Control for Training** In cases of long-tailed distributions of labeled data, the classifier's predictions tend to be biased towards the head classes, which leads to an imbalance in the predictions of pseudo-labels. This results in a misalignment between directly using thresholds to adjust sampling probabilities and prediction biases, leading to the occurrence of confirmation bias. Fortunately, prior work has proposed using logit adjustment-based balanced loss (Menon et al. 2020; Wei and Gan 2023; Du, Han, and Huang 2024; Ma et al. 2024) to reduce the bias of classifiers. Building on this insight and some explorations from previous studies (Wang et al. 2022b; Yang and Xu 2020), we utilize a simplified binary classification problem to further investigate the sampling probabilities of pseudo-labels. Given the imbalanced unlabeled data, we assume that the probability of the label  $Y$  being positive (+1) is  $\gamma \in (\frac{1}{2}, 1)$ , while the probability of it being negative (-1) is  $1 - \gamma$ . The input  $X$  has the following conditional distributions:

$$X | Y = -1 \sim \mathcal{N}(\mu_1, \sigma_1^2), \quad X | Y = +1 \sim \mathcal{N}(\mu_2, \sigma_2^2).$$

We assume  $\mu_2 > \mu_1$  without loss of generality. Since we have already balanced the labeled loss and the adjusted input can be viewed as  $(\xi + \Delta p)$  ( $\Delta p$  is set to the natural log of the label frequency of the positive class), we can roughly estimate the prediction of the pseudo-labels as  $(\xi - \Delta p)$ . Suppose the classifier outputs a confidence score defined as:  $s(\xi) = 1/(1 + \exp(-\beta((\xi - \Delta p) - \frac{\mu_1 + \mu_2}{2})))$ , where  $\beta$  is a positive parameter reflecting the model's learning status and is expected to increase during training as the model becomes more confident gradually, and  $\frac{\mu_1 + \mu_2}{2}$  represents the Bayes' optimal linear decision boundary. We consider a scenario where a fixed threshold  $\rho \in (\frac{1}{2}, 1)$  is used to generate pseudo labels. A sample  $\xi$  is assigned a pseudo label of +1 if  $s(\xi) > \rho$  and -1 if  $s(\xi) < 1 - \rho$ . The pseudo label is set to 0 (masked) if  $1 - \rho \leq s(\xi) \leq \rho$ .

**Theorem 0.1.** *The pseudo label  $Y_{psl}$  has the following probability distribution:*

$$\begin{aligned} P(Y_{psl} = 1) &= \gamma \Phi\left(\frac{\frac{\mu_2 - \mu_1}{2} - \frac{1}{\beta} \log\left(\frac{\rho}{1-\rho}\right) - \Delta p}{\sigma_2}\right) \\ &\quad + (1 - \gamma) \Phi\left(\frac{\frac{\mu_1 - \mu_2}{2} - \frac{1}{\beta} \log\left(\frac{\rho}{1-\rho}\right) - \Delta p}{\sigma_1}\right), \\ P(Y_{psl} = -1) &= (1 - \gamma) \Phi\left(\frac{\frac{\mu_2 - \mu_1}{2} - \frac{1}{\beta} \log\left(\frac{\rho}{1-\rho}\right) + \Delta p}{\sigma_1}\right) \\ &\quad + \gamma \Phi\left(\frac{\frac{\mu_1 - \mu_2}{2} - \frac{1}{\beta} \log\left(\frac{\rho}{1-\rho}\right) + \Delta p}{\sigma_2}\right), \\ P(Y_{psl} = 0) &= 1 - P(Y_{psl} = 1) - P(Y_{psl} = -1), \end{aligned} \quad (4)$$

where  $\Phi$  is the cumulative distribution function of a standard normal distribution.

The proof can be found in the Appendix. Then we can derive the following implications:

- The sampling probability of pseudo-labels is primarily influenced by  $\gamma$ . Given that  $\mu_2 > \mu_1$ , the sampling probability for the head classes in the labeled data is naturally greater than that for the non-head classes, assuming other factors are not considered.
- By controlling the logit adjustment amount  $\Delta p$ , it is possible to mitigate or even reverse the extent to which the sampling probabilities are affected by the inherent imbalance in the data.
- The choice of confidence also impacts the sampling probabilities; however, it is essential to consider that lower sampling probabilities can lead to confirmation bias as  $\beta$  gradually increases.

In dual-classifier settings (Lee, Shin, and Kim 2021; Wei and Gan 2023), the original classifier is typically kept unchanged, while the second classifier is usually introduced to produce balanced output logits. However, this setup prevents us from adjusting the pseudo-label sampling probabilities of either classifier under the separation assumption. On one hand, for the balanced classifier, balanced logits imply that the pseudo-labels sampled are likely to be balanced as well—meaning they approximately match the true distribution of pseudo-labels. In this true distribution, the proportion

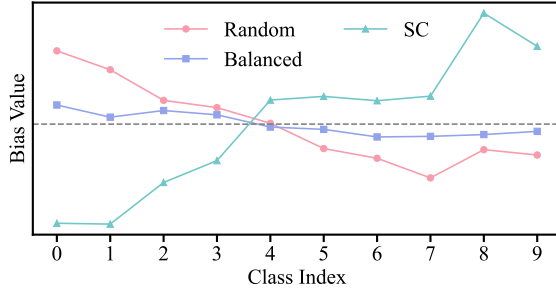


Figure 3: We choose the distribution setting of Consist to illustrate the pattern of the linear layer’s bias term. Under random sampling, since the data follows a long-tailed distribution, the bias term tends to favor head classes. In contrast, under SC-SSL’s controlled expansive sampling, the pseudo-label sampling probability is deliberately skewed toward minority classes, resulting in the opposite trend. Therefore, under balanced sampling, the bias term reflects the effect of optimization imbalance.

of non-head classes may be high or low. When the proportion is low, the sampling probability is essentially governed by the data imbalance factor  $\gamma$ , meaning head classes are still sampled with high probability. Since this classifier is responsible for producing the final outputs and optimizing classification accuracy, it becomes difficult to make significant adjustments to  $\Delta p$ , the change in sampling probability. As a result, increasing the sampling rate of non-head classes becomes challenging. On the other hand, for the original classifier, since  $\Delta p$  cannot be modified directly, the only option is to increase the sampling probability of non-head classes by adjusting the confidence threshold  $\rho$ . However, due to the classifier’s inherent bias toward head classes, strong data augmentations tend to obscure class-discriminative features, causing the prediction  $G(x')$  to favor head classes. This violates the separation assumption defined in Eq. (2).

In summary, neither the output classifier nor the original classifier is suitable for significantly adjusting pseudo-label sampling. Therefore, we introduce an additional *expansive classifier* that shares the same backbone but allows direct control over the sampling probability of non-head classes.

**Sampling Control for Inference** At inference time, despite the alleviation of feature imbalance through our expansive classifier during training, an optimization-induced imbalance still persists within the linear classifier parameters. Formally, a classifier can be expressed as:

$$F(B(x)) = \mathbf{W}B(x) + \mathbf{b},$$

where  $\mathbf{W}$  represents the weight matrix,  $B(x)$  is the feature vector produced by the backbone network, and  $\mathbf{b}$  is the bias term. Since supervised losses of all linear classifiers are optimized using labeled data and traditional cross-entropy, the inherent class imbalance within labeled data inevitably impacts the bias term  $\mathbf{b}$ . We observe that these bias terms encapsulate two distinct sources of imbalance: data distribution imbalance and optimization-induced imbalance, as fig. 3 shows. Under a random sampling regime, head classes

typically exhibit higher bias values, reflecting data imbalance. Conversely, under expansive sampling during training, tail classes display elevated bias values, suggesting an over-correction that inversely mirrors the original data imbalance.

However, the output classifier, trained using a balanced sampling strategy intended for inference, contains bias terms largely independent of data imbalance, thus isolating pure optimization-induced bias. Unlike classifier weights  $\mathbf{W}$ , which involve complex interactions with feature vectors and are challenging to decouple clearly, the bias term  $\mathbf{b}$  provides a straightforward proxy for optimization imbalance.

Therefore, we define this bias term explicitly as an *optimization bias vector*, denoted as  $\mathbf{b}_{opt}$ . At inference, we leverage this vector to calibrate final logits as follows:

$$\tilde{F}(B(x)) = F_b(B(x)) - \mathbf{b}_{opt} = W_b(B(x)), \quad (5)$$

where  $\tilde{F}(B(x))$  represents the bias-corrected logits. This simple yet effective adjustment ensures that the predictions are unbiased with respect to the optimization-induced imbalance, enhancing overall classification performance.

Additionally, we incorporate the  $\mathbf{b}_{opt}$  as a prior to approximate the underlying distribution of unlabeled data. After several epochs of estimation training and performing inference on the unlabeled dataset, we can roughly determine the quantity of each class  $N^e = \{N_1^e, \dots, N_k^e\}$  for  $k = 1, \dots, K$  according to the adjusted balanced output. Let there be  $O$  predefined distributions  $P^{(1)}, P^{(2)}, \dots, P^{(o)}$ , where each distribution  $P^{(o)} = \{p_k^{(o)}\}_{k=1}^K$  satisfies  $\sum_{k=1}^K p_k^{(o)} = 1$  for  $o = 1, \dots, O$ . These distributions are then re-scaled based on the counts of  $N_e$ :

$$Q_k^{(o)} = \frac{p_k^{(o)} \cdot \text{sum}(N^e)}{\sum_{i=1}^K p_i^{(o)}}.$$

The closest distribution can be identified as ( $D_{KL}$  measures KL divergence):

$$o^* = \arg \min_o D_{KL}(N^e, Q^{(o)}).$$

## The Loss Design

Building on the FixMatch algorithm (Sohn et al. 2020), the supervised loss with balanced softmax for each classifier can be formulated as:

$$\mathcal{L}_{sup}(\tau, F) = \frac{1}{B_l} \sum_{i=1}^{B_l} \ell_{CE}(F(B(x_i^l)) + \tau \cdot \Delta p_{y_i}, y_i), \quad (6)$$

where  $\Delta p_{y_i}$  is interpreted as the label frequency of class  $y_i$ ,  $\tau$  is a hyperparameter and  $\ell_{CE}$  denotes the cross entropy loss. For unlabeled data, the consistency loss is:

$$\begin{aligned} \mathcal{L}_{con}(\rho, F) &= \frac{1}{B_u} \sum_{j=1}^{B_u} \ell_{CE}(F(B(\mathcal{A}(x_j^u))), \tilde{y}_j) \cdot \mathcal{M}(x_j^u, \rho), \\ \mathcal{M}(x_j^u, \rho) &= \mathbb{I}(\max(F(B(\mathcal{A}(x_j^u)))) \geq \rho), \end{aligned} \quad (7)$$

| Dataset                             | CIFAR10-LT                  |                             |                             |                             | CIFAR100-LT                 |                             |                             |                             |
|-------------------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|
|                                     | Consist                     |                             | Inverse                     |                             | Consist                     | Consist                     | Inverse                     | Inverse                     |
| Distribution<br>$\gamma_l-\gamma_u$ | 100-100                     |                             |                             |                             | 10-10                       | 15-15                       | 10-10                       | 15-15                       |
| $N-M$                               | 1500-3000                   | 500-4000                    | 1500-3000                   | 500-4000                    | 150-300                     |                             |                             |                             |
| Supervised                          | 64.25 ( $\pm 0.43$ )        | 49.55 ( $\pm 0.92$ )        | 64.25 ( $\pm 0.43$ )        | 49.55 ( $\pm 0.92$ )        | 47.92 ( $\pm 0.55$ )        | 45.65 ( $\pm 0.63$ )        | 48.25 ( $\pm 0.39$ )        | 46.23 ( $\pm 0.50$ )        |
| FixMatch+LA                         | 81.49 ( $\pm 0.77$ )        | 75.26 ( $\pm 1.58$ )        | 80.68 ( $\pm 0.54$ )        | 73.23 ( $\pm 1.01$ )        | 58.56 ( $\pm 0.19$ )        | 55.26 ( $\pm 0.05$ )        | 58.21 ( $\pm 0.24$ )        | 57.93 ( $\pm 0.69$ )        |
| w/ DARP                             | 80.41 ( $\pm 0.63$ )        | 76.93 ( $\pm 0.52$ )        | 81.19 ( $\pm 0.37$ )        | 74.05 ( $\pm 1.28$ )        | 59.94 ( $\pm 0.33$ )        | 55.71 ( $\pm 0.42$ )        | 59.88 ( $\pm 0.39$ )        | 56.04 ( $\pm 0.25$ )        |
| w/ CReST                            | 79.45 ( $\pm 0.15$ )        | 77.24 ( $\pm 0.35$ )        | 85.92 ( $\pm 0.76$ )        | 78.52 ( $\pm 0.90$ )        | 58.78 ( $\pm 0.64$ )        | 56.20 ( $\pm 0.13$ )        | 60.44 ( $\pm 0.09$ )        | 56.40 ( $\pm 0.17$ )        |
| FixMatch+CDMAD                      | 83.33 ( $\pm 0.52$ )        | 79.10 ( $\pm 1.32$ )        | 77.76 ( $\pm 0.56$ )        | 74.65 ( $\pm 1.88$ )        | 59.31 ( $\pm 0.42$ )        | 56.05 ( $\pm 0.23$ )        | 58.72 ( $\pm 0.45$ )        | 55.46 ( $\pm 0.30$ )        |
| w/ ABC                              | 84.56 ( $\pm 0.12$ )        | 80.09 ( $\pm 0.31$ )        | 83.55 ( $\pm 0.08$ )        | 79.46 ( $\pm 0.29$ )        | 58.95 ( $\pm 0.33$ )        | 56.09 ( $\pm 0.26$ )        | 60.23 ( $\pm 0.18$ )        | 57.49 ( $\pm 0.04$ )        |
| w/ DASO                             | 79.56 ( $\pm 0.98$ )        | 75.24 ( $\pm 0.66$ )        | 75.01 ( $\pm 0.12$ )        | 68.88 ( $\pm 0.35$ )        | 58.72 ( $\pm 0.14$ )        | 56.39 ( $\pm 0.28$ )        | 61.06 ( $\pm 0.29$ )        | 58.12 ( $\pm 0.17$ )        |
| FixMatch                            | 76.49 ( $\pm 0.72$ )        | 73.14 ( $\pm 0.13$ )        | 68.92 ( $\pm 0.79$ )        | 62.52 ( $\pm 0.93$ )        | 57.61 ( $\pm 0.44$ )        | 54.02 ( $\pm 0.14$ )        | 57.10 ( $\pm 0.31$ )        | 53.88 ( $\pm 0.25$ )        |
| w/ ABC                              | 82.69 ( $\pm 0.64$ )        | 79.96 ( $\pm 0.09$ )        | 83.22 ( $\pm 0.47$ )        | 79.26 ( $\pm 0.35$ )        | 58.30 ( $\pm 0.24$ )        | 55.67 ( $\pm 0.09$ )        | 59.24 ( $\pm 0.40$ )        | 56.65 ( $\pm 0.33$ )        |
| w/ DASO                             | 78.68 ( $\pm 0.59$ )        | 73.62 ( $\pm 0.40$ )        | 74.52 ( $\pm 0.61$ )        | 67.59 ( $\pm 1.50$ )        | 58.16 ( $\pm 0.45$ )        | 54.92 ( $\pm 0.10$ )        | 59.25 ( $\pm 0.19$ )        | 55.38 ( $\pm 0.14$ )        |
| w/ ACR                              | 84.10 ( $\pm 0.39$ )        | 81.52 ( $\pm 0.24$ )        | 89.46 ( $\pm 0.42$ )        | 84.88 ( $\pm 0.16$ )        | 60.34 ( $\pm 0.66$ )        | 57.46 ( $\pm 0.32$ )        | 61.79 ( $\pm 0.43$ )        | 58.53 ( $\pm 0.51$ )        |
| w/ CPE                              | 84.46 ( $\pm 0.20$ )        | 80.89 ( $\pm 0.09$ )        | 87.10 ( $\pm 0.21$ )        | 83.76 ( $\pm 0.32$ )        | 59.83 ( $\pm 0.29$ )        | 57.00 ( $\pm 0.51$ )        | 60.83 ( $\pm 0.30$ )        | 58.54 ( $\pm 0.08$ )        |
| w/ SC-SSL (Ours)                    | <b>86.53</b> ( $\pm 0.16$ ) | <b>83.89</b> ( $\pm 0.35$ ) | <b>89.97</b> ( $\pm 0.20$ ) | <b>86.02</b> ( $\pm 0.09$ ) | <b>60.65</b> ( $\pm 0.14$ ) | <b>57.88</b> ( $\pm 0.26$ ) | <b>62.99</b> ( $\pm 0.42$ ) | <b>60.27</b> ( $\pm 0.14$ ) |

Table 1: Test accuracy of previous CISSL algorithms and our proposed SC-SSL under consistent and inverse distributions with different numbers of training samples on CIFAR10-LT and CIFAR100-LT benchmarks. The network architecture is WRN-28-2 trained from scratch. We highlight the best number in **bold** and the second best is underlined.

| Dataset                             | CIFAR10-LT                  |                             |                             |                             | STL10-LT                    |                             |                             |                             |
|-------------------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|
|                                     | Uniform                     |                             | Gaussian                    |                             | Unknown                     | Unknown                     |                             |                             |
| Distribution<br>$\gamma_l-\gamma_u$ | 100-1                       |                             | 100-                        | Gaussian-I                  | 10-                         | 20-                         |                             |                             |
| $N-M$                               | 1500-3000                   | 1500-300                    | 500-4000                    | 500-400                     | 1500-3000                   | 150-                        |                             |                             |
| FixMatch                            | 81.51 ( $\pm 1.15$ )        | 73.27 ( $\pm 0.99$ )        | 73.01 ( $\pm 3.81$ )        | 66.47 ( $\pm 0.84$ )        | 76.25 ( $\pm 1.29$ )        | 74.74 ( $\pm 2.03$ )        | 66.67 ( $\pm 1.89$ )        | 55.99 ( $\pm 3.84$ )        |
| w/ ABC                              | 87.89 ( $\pm 1.24$ )        | 81.89 ( $\pm 0.32$ )        | 86.92 ( $\pm 0.68$ )        | 78.26 ( $\pm 0.60$ )        | 86.15 ( $\pm 0.45$ )        | 85.78 ( $\pm 0.39$ )        | 71.12 ( $\pm 1.36$ )        | 66.23 ( $\pm 2.14$ )        |
| w/ CDMAD                            | 90.79 ( $\pm 0.43$ )        | 82.12 ( $\pm 0.74$ )        | 87.11 ( $\pm 0.73$ )        | 80.65 ( $\pm 0.78$ )        | 87.64 ( $\pm 0.36$ )        | 86.25 ( $\pm 0.67$ )        | 71.66 ( $\pm 0.22$ )        | 66.84 ( $\pm 0.37$ )        |
| w/ ACR                              | 93.52 ( $\pm 0.11$ )        | 84.61 ( $\pm 0.50$ )        | 92.13 ( $\pm 0.15$ )        | 80.10 ( $\pm 1.21$ )        | 89.65 ( $\pm 0.68$ )        | 90.00 ( $\pm 0.76$ )        | 76.94 ( $\pm 0.35$ )        | 74.53 ( $\pm 0.83$ )        |
| w/ CPE                              | <b>93.81</b> ( $\pm 0.14$ ) | <b>85.86</b> ( $\pm 0.40$ ) | <b>92.29</b> ( $\pm 0.35$ ) | <b>82.32</b> ( $\pm 0.43$ ) | <b>89.26</b> ( $\pm 0.12$ ) | <b>88.01</b> ( $\pm 0.08$ ) | 73.07 ( $\pm 0.47$ )        | 69.60 ( $\pm 0.20$ )        |
| w/ SC-SSL (Ours)                    | <u>93.79</u> ( $\pm 0.22$ ) | <b>86.45</b> ( $\pm 0.02$ ) | <b>93.33</b> ( $\pm 0.19$ ) | <b>83.11</b> ( $\pm 0.17$ ) | <b>90.84</b> ( $\pm 0.29$ ) | <b>91.25</b> ( $\pm 0.34$ ) | <b>79.26</b> ( $\pm 0.31$ ) | <b>77.11</b> ( $\pm 0.35$ ) |

Table 2: Test accuracy of recent CISSL algorithms and our proposed SC-SSL under uniform, Gaussian, and unknown distributions on CIFAR10-LT and STL10-LT benchmarks.

where  $\tilde{y}_j$  denotes the predicted pseudo-labels and  $\rho$  is initialized from an estimated distribution, which is dynamically adjusted at each iteration  $t$  by:

$$\rho^t(k) = \rho^{t-1}(k) - \alpha \cdot \mathbb{I}(\mathbf{b}_{opt}(k) > \nu), \quad (8)$$

where  $\alpha$  and  $\nu$  are hyperparameters.  $\rho^0$  is initialized based on the expansion factor  $c$ .

The assumed expansion factor  $c$  is determined by  $o^*$ . Generally, the greater the relative quantity of non-head classes in the unlabeled data, the larger the expansion factor  $c$ . Keeping  $\rho_b$ (head) and  $\rho_e$ (head) constant, we only need to initialize  $\rho_b$ (non-head) and  $\rho_e$ (non-head) based on  $c$ :

$$\rho_b^0(\text{non-head}) = \rho_{\max} - \frac{c-4}{10} \cdot \min\{\gamma_u/50, 1\}, \quad (9)$$

$$\rho_e^0(\text{non-head}) = \rho_{\max} - \frac{c-3}{5} \cdot \min\{\gamma_u/20, 1\}. \quad (10)$$

where  $\rho_{\max}$  is a constant representing the maximum confidence level and  $\gamma_u$  is the imbalance ratio according to  $o^*$ .

The overall training objective function is formulated as:

$$\begin{aligned} \mathcal{L} = & \mathcal{L}_{basic} \\ & + \mathcal{L}_{sup}^b(\tau_b, F_b) + \mathcal{L}_{con}^b(\rho_b, F_b) \\ & + \mathcal{L}_{sup}^e(\tau_e, F_e) + \mathcal{L}_{con}^e(\rho_e, F_e), \end{aligned} \quad (11)$$

where  $\mathcal{L}_{basic}$  is the loss of base SSL algorithms,  $\tau_b$  and  $\tau_e$  are hyperparameters for output classifier and expansive classifier respectively.

## Experiments

### Analysis

#### Experimental Settings

**Datasets and Baselines** We conduct the experiments on widely used datasets, including CIFAR10-LT (Krizhevsky 2009), CIFAR100-LT (Krizhevsky 2009), STL10-LT (Coates, Ng, and Lee 2011), and ImageNet-127 (Fan, Dai, and Schiele 2021). Following previous works, we split the training data with a certain number of samples while

| Algorithm / Resolution | 32 × 32     | 64 × 64     |
|------------------------|-------------|-------------|
| FixMatch               | 35.7        | 44.0        |
| w/ CDMAD               | 55.6        | 61.7        |
| w/ ACR                 | 57.2        | 63.6        |
| w/ CPE                 | 57.8        | 64.1        |
| w/ SimPro              | 59.4        | 67.2        |
| w/ SC-SSL              | <b>62.3</b> | <b>69.4</b> |

Table 3: Test accuracy of recent CISSL algorithms and our proposed SC-SSL on ImageNet-127 benchmarks.

| - | C     | I     |
|---|-------|-------|
| 2 | 83.89 | 86.02 |
| 3 | 83.54 | 85.98 |
| 4 | 83.50 | 86.15 |

Table 4: Comparison of different partitioning strategies. Classes are divided into 2/3/4 intervals. ‘C’ and ‘I’ denote consist and inverse, respectively.

| -    | A1          | A2          | B1          | B2          |
|------|-------------|-------------|-------------|-------------|
| ACR  | 82.9        | 82.7        | 83.4        | 83.1        |
| CPE  | 82.0        | 81.4        | 82.7        | 81.9        |
| Ours | <b>84.2</b> | <b>83.1</b> | <b>85.1</b> | <b>85.2</b> |

Table 5: Test accuracy on 4 chaotic distributions ( $N : M = 500 : 4000$ ). A and B indicate the dominance of head/non-head classes in the unlabeled data. We generate 2 types of random distribution for each of them.

controlling the imbalance ratio of data.  $\tau_{ub}$  and  $\tau_e$  are set to 2 and 4 for all datasets. For simplicity, we assume that  $c$  is dataset-independent and varies only with changes in the distribution of unlabeled data. Given 5 anchor distributions (consist, uniform, inverse, Gaussian, and inverse-Gaussian) above, the values of  $c$  are set to 4, 5, 6, 4, and 6, respectively.  $\rho_{\max}$  is set to 0.95.  $\rho_b(\text{head})$  and  $\rho_e(\text{head})$  are set to  $\rho_{\max}$ . Step size  $\alpha$  and threshold  $\nu$  in eq. (8) are set to 0.005 and 1.0 for all datasets.

We conduct comparisons with several CISSL algorithms that have been published in leading conferences and journals over the last few years. The baseline algorithms include LA (Menon et al. 2020), DARP (Kim et al. 2020), CReST (Wei et al. 2021), ABC (Lee, Shin, and Kim 2021), DASO (Oh, Kim, and Kweon 2021), ACR (Wei and Gan 2023), CDMAD (Lee and Kim 2024), CPE (Ma et al. 2024), and SimPro (Du, Han, and Huang 2024). All the reproduced results are consistent with the original paper or the codebase. We test our SC-SSL algorithm on the widely used USB codebase. More comparisons can be found in the Appendix.

**Training Details** We adhere to the default hyperparameters specified in USB, setting the batch sizes for labeled and unlabeled data to 64 and 128, respectively. For CIFAR10-LT, CIFAR100-LT, and STL10-LT, the input images are resized to  $32 \times 32$  pixels and the backbone network is WRN-28-2 (Zagoruyko and Komodakis 2016) without any pre-training. For ImageNet-127, we downsample the images to  $64 \times 64$  and  $32 \times 32$  following CoSSL (Fan, Dai, and Schiele 2021) and the backbone is ResNet-50 (He et al. 2015). For the balanced and expansive classifier, the loss ratio of the unlabeled part is set to 2. Additionally, since the distributions of unlabeled data for comparison follow predefined distributions (as in some previous works (Ma et al. 2024; Wei and Gan

| $\rho$  | $\tau$ | $c$ | Consist      | Inverse      |
|---------|--------|-----|--------------|--------------|
| c-init  | (0, 0) | 4   | 82.33        | 82.16        |
| c-init  | (2, 2) | 6   | 81.55        | 86.79        |
| c-init  | (1, 2) | 4   | 85.28        | 86.12        |
| c-init  | (2, 4) | 4   | <b>86.53</b> | 88.54        |
| c-init  | (2, 4) | 6   | 84.11        | <b>89.97</b> |
| fix-max | (2, 4) | -   | 84.70        | 86.90        |
| fix-max | (2, -) | -   | 83.80        | 86.49        |
| fix-min | (2, 1) | -   | 80.22        | 85.65        |
| fix-min | (2, 4) | -   | 79.04        | 86.05        |

Table 6: Ablation study of the key factors in our framework. ‘fix’ indicates the  $\rho$  is set to be  $\rho_{\max}$  or  $\rho_{\min}$  without dynamic sampling.

2023; Lee and Kim 2024)), SC-SSL reweights (Lai et al. 2022) the loss function based on the prior of unlabeled sample counts. We employ the SGD optimizer with a fixed learning rate of 0.03, a momentum of 0.9, and a weight decay of 0.0005.

## Main Results

In our experiments, we assessed the test accuracy of various CISSL algorithms, including our proposed SC-SSL, on CIFAR10-LT and CIFAR100-LT datasets. Using a WRN-28-2 architecture trained from scratch, we measured performance under consistent and inverse distributions as well as uniform and Gaussian distributions. The results are shown in table 1, table 2 and table 3.

**Why are classes simply divided into head and non-head types? — What if the unlabeled data distribution is chaotic?** Due to the current research focusing on long-tailed distributions, there is a significant disparity in the amount of labeled data between head and tail classes. Our goal is to enable non-head classes to leverage unlabeled data to enhance generalization performance, so we have roughly categorized them into two groups. This distinction is sufficient for the distributions discussed in previous work. Our experimental results in table 4 also support this when we try finer-grained  $\rho$  initialization. When the unlabeled data is chaotic, as the example in fig. 2 shows, there is no significant difference in quantity between non-head and head classes. We can treat them as uniformly distributed, assuming  $c = 5$ , since the dynamic sampling probabilities will come into play. The experimental results are shown in table 5.

**What is the impact of ignoring certain factors?** All three key factors are essential. Without  $\gamma$ , it is impossible to initialize the sampling settings; when  $c$  is small, excessively high sampling probabilities can lead to confirmation errors. If  $\Delta p$  is not used, it becomes challenging to address the impact of bias in the labeled data itself. Furthermore, without  $\rho$ , there is no way to dynamically adjust the sampling probabilities, as  $\Delta p$  is determined based on the labeled data, making it difficult to estimate the effect of unlabeled data consistency regularization on it. The experimental results ( $N : M = 1500 : 3000$ ) are shown in table 6.

## References

- Berthelot, D.; Carlini, N.; Cubuk, E. D.; Kurakin, A.; Sohn, K.; Zhang, H.; and Raffel, C. 2019a. ReMixMatch: Semi-Supervised Learning with Distribution Alignment and Augmentation Anchoring. *ArXiv*, abs/1911.09785.
- Berthelot, D.; Carlini, N.; Goodfellow, I. J.; Papernot, N.; Oliver, A.; and Raffel, C. 2019b. MixMatch: A Holistic Approach to Semi-Supervised Learning. *ArXiv*, abs/1905.02249.
- Coates, A.; Ng, A.; and Lee, H. 2011. An Analysis of Single-Layer Networks in Unsupervised Feature Learning. In *International Conference on Artificial Intelligence and Statistics*.
- Du, C.; Han, Y.; and Huang, G. 2024. SimPro: A Simple Probabilistic Framework Towards Realistic Long-Tailed Semi-Supervised Learning. *ArXiv*, abs/2402.13505.
- Fan, Y.; Dai, D.; and Schiele, B. 2021. CoSSL: Co-Learning of Representation and Classifier for Imbalanced Semi-Supervised Learning. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 14554–14564.
- Guo, L.-Z.; and Li, Y.-F. 2022. Class-Imbalanced Semi-Supervised Learning with Adaptive Thresholding. In *International Conference on Machine Learning*.
- He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778.
- 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. *ArXiv*, abs/2007.08844.
- Krizhevsky, A. 2009. Learning Multiple Layers of Features from Tiny Images.
- Lai, Z.; Wang, C.; Gunawan, H.; Cheung, S.-C. S.; and Chuah, C.-N. 2022. Smoothed Adaptive Weighting for Imbalanced Semi-Supervised Learning: Improve Reliability Against Unknown Distribution Data. In *International Conference on Machine Learning*.
- Lee, H.; and Kim, H. 2024. CDMAD: Class-Distribution-Mismatch-Aware Debiasing for Class-Imbalanced Semi-Supervised Learning. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 23891–23900.
- Lee, H.; Shin, S.; and Kim, H. 2021. ABC: Auxiliary Balanced Classifier for Class-imbalanced Semi-supervised Learning. In *Neural Information Processing Systems*.
- Li, J.; Tan, Z.; Wan, J.; Lei, Z.; and Guo, G. 2022. Nested Collaborative Learning for Long-Tailed Visual Recognition. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 6939–6948.
- Li, J.; Xiong, C.; and Hoi, S. C. H. 2020. CoMatch: Semi-supervised Learning with Contrastive Graph Regularization. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 9455–9464.
- Li, L.; Tao, B.; Han, L.; Zhan, D.-C.; and Ye, H.-J. 2024. Twice Class Bias Correction for Imbalanced Semi-Supervised Learning. *ArXiv*, abs/2312.16604.
- Ma, C.; Elezi, I.; Deng, J.; Dong, W.; and Xu, C. 2024. Three Heads Are Better Than One: Complementary Experts for Long-Tailed Semi-supervised Learning. In *AAAI Conference on Artificial Intelligence*.
- Menon, A. K.; Jayasumana, S.; Rawat, A. S.; Jain, H.; Veit, A.; and Kumar, S. 2020. Long-tail learning via logit adjustment. *ArXiv*, abs/2007.07314.
- Miyato, T.; Ichi Maeda, S.; Koyama, M.; and Ishii, S. 2017. Virtual Adversarial Training: A Regularization Method for Supervised and Semi-Supervised Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41: 1979–1993.
- Oh, Y.; Kim, D.-J.; and Kweon, I.-S. 2021. DASO: Distribution-Aware Semantics-Oriented Pseudo-label for Imbalanced Semi-Supervised Learning. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 9776–9786.
- Selvaraju, R. R.; Das, A.; Vedantam, R.; Cogswell, M.; Parikh, D.; and Batra, D. 2016. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. *International Journal of Computer Vision*, 128: 336 – 359.
- Sohn, K.; Berthelot, D.; Li, C.-L.; Zhang, Z.; Carlini, N.; Cubuk, E. D.; Kurakin, A.; Zhang, H.; and Raffel, C. 2020. FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence. *ArXiv*, abs/2001.07685.
- Tarvainen, A.; and Valpola, H. 2017. Weight-averaged consistency targets improve semi-supervised deep learning results. *ArXiv*, abs/1703.01780.
- Wang, R.; Jia, X.; Wang, Q.; Wu, Y.; and Meng, D. 2022a. Imbalanced Semi-supervised Learning with Bias Adaptive Classifier. In *International Conference on Learning Representations*.
- Wang, X.; Lian, L.; Miao, Z.; Liu, Z.; and Yu, S. X. 2020. Long-tailed Recognition by Routing Diverse Distribution-Aware Experts. *ArXiv*, abs/2010.01809.
- Wang, Y.; Chen, H.; Heng, Q.; Hou, W.; Savvides, M.; Shinzaki, T.; Raj, B.; Wu, Z.; and Wang, J. 2022b. FreeMatch: Self-adaptive Thresholding for Semi-supervised Learning. *ArXiv*, abs/2205.07246.
- Wei, C.; Shen, K.; Chen, Y.; and Ma, T. 2020. Theoretical Analysis of Self-Training with Deep Networks on Unlabeled Data. *ArXiv*, abs/2010.03622.
- Wei, C.; Sohn, K.; Mellina, C.; Yuille, A. L.; and Yang, F. 2021. CReST: A Class-Rebalancing Self-Training Framework for Imbalanced Semi-Supervised Learning. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 10852–10861.
- Wei, T.; and Gan, K. 2023. Towards Realistic Long-Tailed Semi-Supervised Learning: Consistency is All You Need. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 3469–3478.
- Xiang, L.; and Ding, G. 2020. Learning From Multiple Experts: Self-paced Knowledge Distillation for Long-tailed Classification. In *European Conference on Computer Vision*.

Yang, Y.; and Xu, Z. 2020. Rethinking the Value of Labels for Improving Class-Imbalanced Learning. *ArXiv*, abs/2006.07529.

Yu, Z. L.; Li, Y.; and Lee, Y. J. 2023. InPL: Pseudo-labeling the Inliers First for Imbalanced Semi-supervised Learning. *ArXiv*, abs/2303.07269.

Zagoruyko, S.; and Komodakis, N. 2016. Wide Residual Networks. *ArXiv*, abs/1605.07146.

Zheng, H.; Zhou, L.; Li, H.; Su, J.; Wei, X.; and Xu, X. 2024. BEM: Balanced and Entropy-Based Mix for Long-Tailed Semi-Supervised Learning. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 22893–22903.