Three Heads Are Better than One:
Complementary Experts for Long-Tailed Semi-supervised Learning

Chengcheng Ma[1,2*], **Ismail Elezi**[3], **Jiankang Deng**[3], **Weiming Dong**[1†], **Changsheng Xu**[1]

[1]Institute of Automation, Chinese Academy of Sciences, Beijing, China
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China
[3]Huawei Noah's Ark Lab, London, UK

Abstract

We address the challenging problem of Long-Tailed Semi-Supervised Learning (LTSSL) where labeled data exhibit imbalanced class distribution and unlabeled data follow an unknown distribution. Unlike in balanced SSL, the generated pseudo-labels are skewed towards *head* classes, intensifying the training bias. Such a phenomenon is even amplified as more unlabeled data will be mislabeled as *head* classes when the class distribution of labeled and unlabeled datasets are mismatched. To solve this problem, we propose a novel method named ComPlementary Experts (CPE). Specifically, we train multiple experts to model various class distributions, each of them yielding high-quality pseudo-labels within one form of class distribution. Besides, we introduce Classwise Batch Normalization for CPE to avoid performance degradation caused by feature distribution mismatch between head and non-head classes. CPE achieves state-of-the-art performances on CIFAR-10-LT, CIFAR-100-LT, and STL-10-LT dataset benchmarks. For instance, on CIFAR-10-LT, CPE improves test accuracy by over $>2.22\%$ compared to baselines. Code is available at https://github.com/machengcheng2016/CPE-LTSSL.

Introduction

Semi-supervised learning (SSL) is a promising approach for improving the performance of deep neural networks (DNNs). In SSL, only a limited amount of labeled data is available, with the majority of data being unlabeled. Most existing SSL approaches assume that the labeled and unlabeled data follow the same distribution and that the distribution is class-balanced. However, this assumption is often violated in real-world applications, where the class distribution of the labeled data is often long-tailed, with a few (head) classes containing significantly more samples than the other (tail) classes. In long-tailed semi-supervised learning (LTSSL), the distribution of the unlabeled data can take three forms: (i) it can be *long-tailed*, following the same distribution as the labeled data; (ii) it can be *uniform*, with a balanced class distribution; or (iii) it can be *inverse long-tailed*, with the head classes of the labeled data being the tail classes of the unlabeled data, and vice versa.

In the fully-supervised setting, DNNs trained on imbalanced labeled data are prone to mostly predict the head classes. This issue is exacerbated in LTSSL, as the unlabeled data are mostly pseudo-labeled as head classes during training, intensifying the training bias. This can lead to lower test accuracy on tail classes, especially when the class distributions of the labeled and unlabeled data are mismatched. Unfortunately, most of the pioneer LTSSL works assume a similar class distribution between labeled and unlabeled sets, causing lower performances when such assumption is inevitably violated. Very recently, ACR (Wei and Gan 2023) proposes to handle varying class distributions of unlabeled sets by introducing adaptive logit adjustment and achieving state-of-the-art (SOTA) performance. Though effective, its pseudo-labels are generated by a single classifier. As we will show in the experimental section, the pseudo-label quality by ACR can be further improved.

In this paper, we propose a flexible and end-to-end LTSSL algorithm, namely **C**om**P**lementary **E**xperts (CPE). To cope with varying class distributions of unlabeled sets, we train multiple experts where each expert is tasked to model one class group. We do so by using different logit adjustments (Menon et al. 2020) during the computation of the supervised loss. Specifically, the first expert is trained with regular cross entropy loss to fit the skewed class distribution of labeled sets, so it is natural to see it perform well when the distribution of unlabeled and labeled sets are consistent. In contrast, the second and third experts are trained with different intensities of logit adjustments, which are encouraged to yield class-balanced and inversely skewed predictions, respectively. Thus, they can deal with the uniform and inverse long-tailed cases, where the corresponding shape of predictions is just in need. In such a complementary manner, one of three experts in our CPE algorithms can always predict pseudo-labels with higher F1 scores than ACR (see Fig. 1).

Having three experts results in a higher recall in the unlabeled samples belonging to medium and tail classes, especially in the *uniform* and *inverse* cases. Consequently, the extracted features within medium and tail classes tend to follow distribution with larger variances compared with head classes. To prevent such phenomenon from harming model performance, we propose Classwise Batch Normalization (CBN) mechanism to realize classwise feature processing. In CBN, we design three different batch normalization lay-
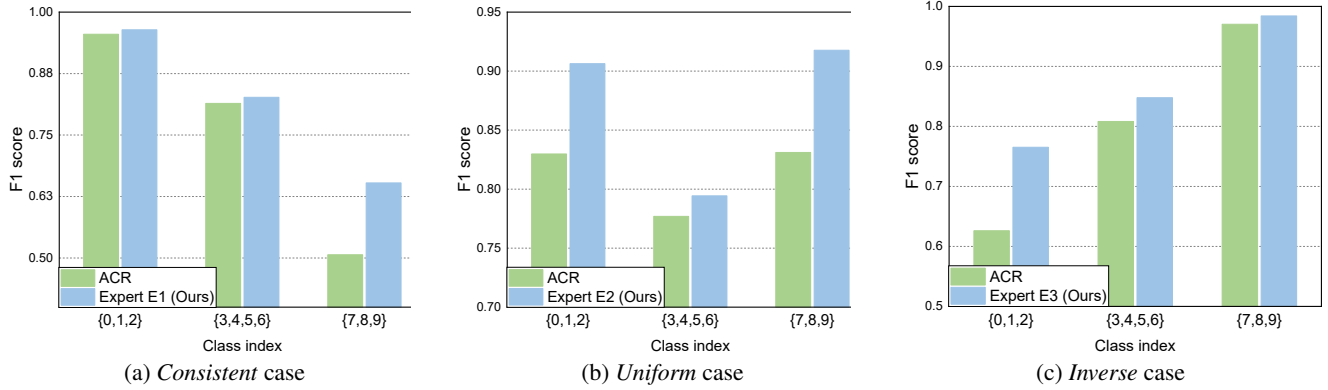
Figure 1: Comparison of F1 score of pseudo-label predictions between ACR (Wei and Gan 2023) and our CPE method under the {"*consistent*", "*uniform*", "*inverse*"} cases. The dataset is CIFAR-10-LT with imbalance ratio γ_l being 100. One of experts in our CPE can generate pseudo-labels with higher quality.

ers for three groups of classes, to affect both the forward and backward calculation during the computation of the unsupervised loss. We will validate the effectiveness of CBN mechanism in the experimental section.

In summary, our **contribution** is as follows:

- We determine that a single classifier cannot model all possible distributions and propose to model the distributions by three experts trained with different logit adjustments, namely ComPlementary Experts (CPE).
- To deal with different feature distributions, we propose Classwise Batch Normalization (CBN) mechanism for CPE training.
- We reach the new state-of-the-art performances on three popular LTSSL benchmarks.

Related Work

Semi-Supervised Learning

The underlying objective of Semi-Supervised Learning (SSL) is to enhance the performance of Deep Neural Networks (DNNs) through the incorporation of supplementary unlabeled training data, particularly in scenarios where the availability of labeled samples is notably constrained. Until recently, FixMatch (Sohn et al. 2020) and its derivative methodologies such as FlexMatch (Zhang et al. 2021), FreeMatch (Wang et al. 2022b), and SoftMatch (Chen et al. 2022b) have emerged as among the most efficacious algorithms for the task of classification. These methodologies adhere to the foundational principle of self-training (Pise and Kulkarni 2008), wherein model predictions on unlabeled data are utilized as pseudo-labels during the training process. These approaches concentrate on refining the quality of pseudo-labels through strategies such as curriculum learning (Zhang et al. 2021), adaptive thresholding (Wang et al. 2022b), and sample weighting (Chen et al. 2022b). However, many existing SSL approaches make the presumption of balanced class distributions within both the labeled and unlabeled training sets. This assumption, however, tends to be incompatible with real-world scenarios, where the class

distribution of natural data frequently adheres to Zipf's law, manifesting a long-tailed distribution (Johnson and Khoshgoftaar 2019; Liu et al. 2019; Reed 2001). As corroborated by subsequent experiments, the performance of conventional FixMatch deteriorates notably, particularly when the unlabeled training set conforms to an inverse class distribution in relation to the labeled set.

Long-Tailed Learning

Since the class distribution of the training set is usually imbalanced, existing algorithms in long-tailed learning attempt to improve the generalization ability of DNN towards the *medium* and *tail* classes. Generally, the framework of existing methods can be summarized into four categories: *Resampling* (Chawla et al. 2002; Buda, Maki, and Mazurowski 2018) conducts under-sampling on head classes or oversampling on tail classes to balance the training data; *Reweighting* (Khan et al. 2017; Cui et al. 2019) adjusts the class weight or sample weight to maintain a balanced training process; *Ensemble learning* (Wang et al. 2020; Cai, Wang, and Hwang 2021; Zhang et al. 2022) is based on multiple experts (classifier heads) to enhance the representation learning; *Loss modification* (Cao et al. 2019; Hong et al. 2021) modifies the logit value by margins in either the training or inference stage and the typical work "logit adjustment" (Menon et al. 2020) which has been proved as the Bayes-optimal solution of the long-tailed problem. Nonetheless, the current approaches are predominantly tailored for the fully-supervised context, rendering them unsuitable for semi-supervised learning where the class distribution is unspecified.

Long-Tailed Semi-Supervised Learning

Due to its practicality, LTSSL has reached increasing attention in recent years. Different from balanced SSL, the unlabeled data are prone to be pseudo-labeled as head classes during training, leading to low recall on tail classes. To address the challenge of LTSSL, previous works draw inspiration from existing solutions to long-tailed classification to

generate unbiased and accurate pseudo-labels, such as re-sampling (Lee, Shin, and Kim 2021; Guo and Li 2022), re-weighting (Lai et al. 2022), transfer learning (Fan et al. 2022), and logit adjustment (Wang et al. 2022a). However, these works usually assume a similar class distribution between labeled and unlabeled set and show inferior performances when such assumption is violated. Another branch of LTSSL works follows the popular idea of SSL to augment the labeled training set with the most reliable unlabeled data multiple times (Wei et al. 2021; Chen et al. 2022a). However, such approaches are not in an end-to-end fashion and require multiple re-training costs. Very recently, ACR (Wei and Gan 2023) is proposed to dynamically adjust the intensity of logit adjustment by measuring the distribution divergence between labeled and unlabeled sets, which can handle varying class distributions of unlabeled set and achieves the state-of-the-art (SOTA) performances. Though effective, its pseudo-labels are generated by one single classifier leading to sub-optimal performance. We will show in the following sections that the pseudo-label quality can still be improved.

Methodology

Preliminaries

Problem Formulation. Long-tailed semi-supervised learning, also known as imbalanced semi-supervised learning, involves a labeled training set $\mathcal{D}_l = \{\mathbf{x}_i^l, y_i\}_{i=1}^N$ and an unlabeled training set $\mathcal{D}_u = \{\mathbf{x}_j^u\}_{j=1}^M$ sharing the same class set, where $\mathbf{x}$ represents a training sample and $y \in \{1, \ldots, C\}$ represents a class label. Let N_k and M_k be the number of samples within the k class in $\mathcal{D}_l$ and $\mathcal{D}_u$, respectively, so $\sum_{k=1}^C N_k = N$ and $\sum_{k=1}^C M_k = M$. As the class distribution of $\mathcal{D}_l$ is skewed, the classes are usually sorted in descending order, *i.e.*, $N_1 > N_2 > \cdots > N_C$. Correspondingly, the imbalance ratio of $\mathcal{D}_l$ is defined as $\gamma_l = N_1/N_C$. As for $\mathcal{D}_u$, the class distribution is unknown because all labels are unavailable. In this paper, we follow previous works (Oh, Kim, and Kweon 2022; Wei and Gan 2023) to consider three cases of class distribution of unlabeled set, that are {"*consistent*", "*uniform*", "*inverse*"}. In *consistent* case, we have $M_1 > M_2 > \cdots > M_C$ and $\gamma_u = \gamma_l$, where γ_u is defined as M_1/M_C. In *uniform* case, we have $M_1 = M_2 = \cdots = M_C$ and $\gamma_u = 1$. In *inverse* case, we have $M_1 < M_2 < \cdots < M_C$ and $\gamma_u = 1/\gamma_l$.

The goal of LTSSL is to train a DNN classifier $f(\cdot)$ parameterized by θ on $\mathcal{D}_l$ and $\mathcal{D}_u$, and to achieve good performances on a class-balanced test set $\mathcal{D}_{test}$. Generally, the training loss of LTSSL is similar to that of SSL and is composed of a supervised term and an unsupervised term:

$$L = \frac{1}{B_l} \sum_{i=1}^{B_l} \ell_{CE}\big(f(\mathbf{x}_i^l), y_i\big)$$
$$+ \frac{\lambda}{B_u} \sum_{j=1}^{B_u} \ell_{CE}\big(f(\mathbf{x}_j^u), \hat{y}_j\big) \cdot \mathbb{I}\Big(\big\|\sigma\big(f(\mathbf{x}_j^u)\big)\big\|_\infty > \rho\Big), \quad (1)$$

where $\hat{y}_j$ denotes the pseudo-label of unlabeled sample $\mathbf{x}_j^u$, usually being the model prediction $\hat{y}_j = \arg\max f(\mathbf{x}_j^u)$. $\sigma(\cdot)$ denotes the softmax function, and $\mathbb{I}(\cdot)$ denotes a binary sample mask to select samples with confidence larger

Distribution case	τ	F1 score of pseudo-labels		
		Head	Medium	Tail
Consistent ($\gamma_l = \gamma_u = 100$)	**0**	**0.96**	**0.81**	**0.65**
	2	0.95	0.80	0.51
	4	0.28	0.80	0.27
Uniform ($\gamma_l = 100, \gamma_u = 1$)	0	0.88	0.79	0.87
	2	**0.91**	**0.80**	**0.92**
	4	0.28	0.79	0.73
Inverse ($\gamma_l = 100, \gamma_u = \frac{1}{100}$)	0	0.41	0.82	0.92
	2	0.59	0.83	0.97
	4	**0.77**	**0.85**	**0.98**

Table 1: F1 score of pseudo-labels under three different cases with varying τ in Eq. (2). The dataset is CIFAR-10-LT. τ being 2 means balanced training on labeled set, while τ being 0 or 4 means under- or over- balancing. We clearly see that each τ can only generate high-quality pseudo-labels in one distribution case.

than the threshold ρ. $\ell_{CE}(\cdot, \cdot)$ denotes the cross entropy loss. B_l and B_u denote the size of labeled and unlabeled data batches, respectively. λ is a hyper-parameter that controls the strength of the unsupervised loss.

Logit Adjustment for Long-Tailed Recognition. The logit adjustment (Menon et al. 2020) is theoretically proven as the Bayes-optimal solution to the long-tailed recognition task. Formally, the output logits are added to the logarithm of prior π during training, formulating the balanced cross entropy loss:

$$L_{bal} = \frac{1}{B} \sum_{i=1}^B \ell_{CE}\big(f(\mathbf{x}_i) + \tau \cdot \log \pi, \ y_i\big). \quad (2)$$

The prior π is pre-computed as the label frequencies, *i.e.*, $\pi_i = N_i/N$. The positive scaling factor τ controls the intensity of adjustment. Intuitively, the value of $\log \pi_i$ lies in $(-\infty, 0)$, and is close to $-\infty$ for the tail classes. By adjustment, the logit values of tail classes are encouraged to cancel out the value of $\log \pi_i$ during training, thus becomes larger than those of head classes ultimately. On the other hand, the DNN classifier trained on imbalanced dataset tends to yield larger logit values for head classes, thus the logit values for all classes can be balanced if classifier is trained by L_{bal}, leading to class-balanced predictions during test time.

However, under the LTSSL setting, the unknown class distribution of unlabeled set makes the logit adjustment algorithm somewhat problematic. To be specific, if we still pre-compute π of $\mathcal{D}_l$ and replace ℓ_{CE} with L_{bal} in the first term of Eq. (1), then only in the *uniform* case can the classifier generate high-quality pseudo-labels (see Table 1 for an empirical evidence). This is because L_{bal} always prefers class-balanced predictions. In contrast, if the class distribution of $\mathcal{D}_u$ is also skewed such as in the *consistent* or *inverse* case, then the classifier needs to be under-balanced or over-balanced to cope with the corresponding case (with lower or higher τ), which contradicts the supervised loss on labeled set in turn. We will show how to address such a dilemma in the next section.
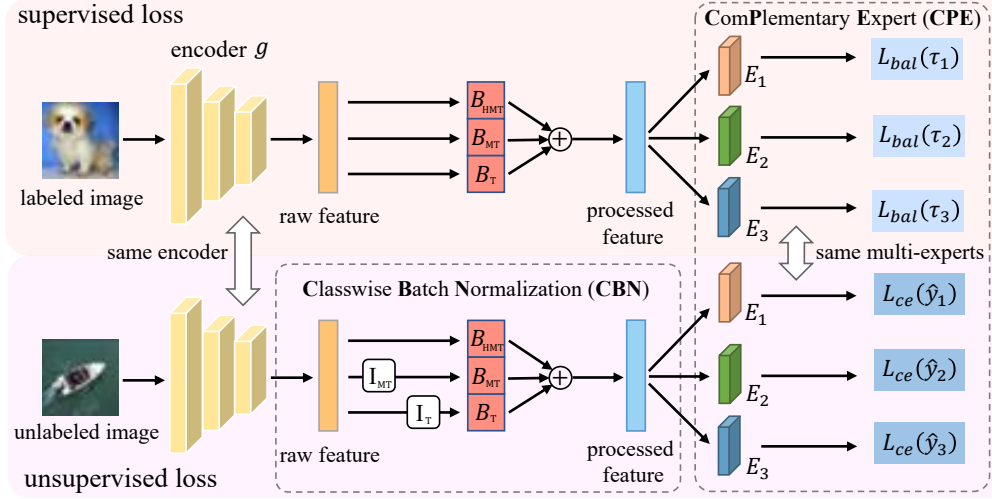
Figure 2: Overview of our CPE algorithm together with the CBN mechanism. I_{MT} and I_T represent the class mask in Eq. (5). $L_{bal}(\tau)$ is short for balanced cross entropy loss in Eq. (2). $L_{ce}(\hat{y})$ denotes the cross entropy loss with pseudo-label $\hat{y}$.

ComPlementary Experts (CPE)

The two-stage training (Kang et al. 2019) is a recognized approach for long-tailed fully-supervised learning. It involves separate training of the feature extractor and classification head in two stages. However, due to its computational demands, in LTSSL, prior works (Fan et al. 2022; Wei and Gan 2023) often opt for a single-stage training using a dual-head DNN architecture. Typically, these algorithms update the shared feature extractor by aggregating gradients from the first head to facilitate representation learning. The second head is then updated using techniques tailored for long-tailed scenarios, aiming to achieve balanced predictions. An example is substituting ℓ_{CE} with L_{bal} in the first term of Eq. (1), which is finally employed for evaluation.

However, we notice that all existing methods only utilize one single head for pseudo-label generation, which cannot perfectly fit the varying class distributions of unlabeled set. According to Table 1, the quality of pseudo-labels can be less satisfying if there exists a mismatch between logit adjustment intensity τ and imbalance ratio γ_u. To this end, we propose to concatenate the shared feature extractor $g(\cdot)$ with three different heads, namely experts $\{E_1, E_2, E_3\}$. Specifically, each of experts is trained by balanced cross entropy loss L_{bal} (see Eq. (2)) with a specified intensity τ_i, and the total supervised loss is formulated as

$$
L_{CPE}^{sup} = \frac{1}{B_l} \sum_{i=1}^{B_l} \ell_{CE}\Big(E_1\big(g(\mathbf{x}_i^l)\big) + \tau_1 \cdot \log \pi, \ y_i \Big)
$$
$$
+ \frac{1}{B_l} \sum_{i=1}^{B_l} \ell_{CE}\Big(E_2\big(g(\mathbf{x}_i^l)\big) + \tau_2 \cdot \log \pi, \ y_i \Big)
$$
$$
+ \frac{1}{B_l} \sum_{i=1}^{B_l} \ell_{CE}\Big(E_3\big(g(\mathbf{x}_i^l)\big) + \tau_3 \cdot \log \pi, \ y_i \Big). \quad (3)
$$

Correspondingly, the formulation of total unsupervised loss follows that of Eq. (1):

$$
L_{CPE}^{unsup}
$$
$$
= \frac{\lambda}{B_u} \sum_{j=1}^{B_u} \ell_{CE}\Big(E_1\big(g(\mathbf{x}_j^u)\big), \hat{y}_{j(1)} \Big) \cdot \mathbb{I}\Big(\|\sigma(E_1(g(\mathbf{x}_j^u)))\|_\infty > \rho \Big)
$$
$$
+ \frac{\lambda}{B_u} \sum_{j=1}^{B_u} \ell_{CE}\Big(E_2\big(g(\mathbf{x}_j^u)\big), \hat{y}_{j(2)} \Big) \cdot \mathbb{I}\Big(\|\sigma(E_2(g(\mathbf{x}_j^u)))\|_\infty > \rho \Big)
$$
$$
+ \frac{\lambda}{B_u} \sum_{j=1}^{B_u} \ell_{CE}\Big(E_3\big(g(\mathbf{x}_j^u)\big), \hat{y}_{j(3)} \Big) \cdot \mathbb{I}\Big(\|\sigma(E_3(g(\mathbf{x}_j^u)))\|_\infty > \rho \Big),
$$
$$
(4)
$$

where $\hat{y}_{(i)}$ denotes the pseudo-label predicted by E_i. Then the total loss of CPE is $L_{CPE} = L_{CPE}^{sup} + L_{CPE}^{unsup}$.

In training stage, one of the three experts in CPE can always generate high-quality pseudo-labels for each class, complementing the defects of other two experts and ensuring the quality of representation learning. In inference stage, CPE assumes the test set is class-balanced and utilizes the second expert E_2 for evaluation. The overview of CPE is illustrated in Fig. 2.

Classwise Batch Normalization for CPE

Despite our CPE method being effective, having three experts results with a higher recall in the unlabeled samples belonging to medium and tail classes, especially in the *uniform* and *inverse* cases. Consequently, the variance of extracted features in medium and tail classes tends to become larger than that of head classes, as we empirically validate in Fig. 3. Specifically, we record the average and standard deviation of each feature channel in the last layer of the feature extractor by class. Such a phenomenon can somehow harm the model performance as previously discussed in adversarial learning (Xie et al. 2020) and fully-supervised long-tailed learning (Cui et al. 2022).

To address this issue, inspired by (Cui et al. 2022), we propose Classwise Batch Normalization (CBN) mechanism
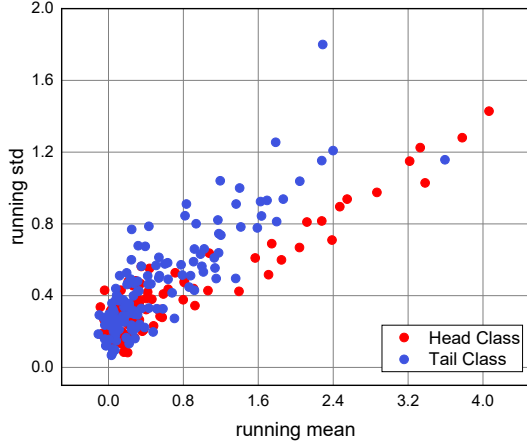
Figure 3: Statistics of extractor features within head and tail classes in CIFAR-10-LT with $(\gamma_l, \gamma_u) = (100, 1/100)$. Each dot represents the running mean and standard deviation of a channel in the BN layer. After pseudo-labeling, features of unlabeled data within tail classes have a higher variance and follow a different distribution from that of the head classes.

to realize classwise feature processing. CBN involves three different BN layers to re-scale the features output by the shared encoder $g(\cdot)$, denoted by $\{\text{BN}_{\text{HMT}}, \text{BN}_{\text{MT}}, \text{BN}_{\text{T}}\}$. Concretely, BN_{HMT} processes the features belonging to all classes, while BN_{MT} processes features in medium and tail classes (denoted by $\mathcal{M}$ and $\mathcal{T}$), and BN_{T} only processes features in tail classes. Formally, the cross entropy loss of expert E_i on unlabeled data $\mathbf{x}_u$ in Eq. (4) is re-formulated as

$$
L_{CPE,i}^{unsup} = \frac{1}{S}\Bigg[\ell_{CE}\Big(E_i\big(\text{BN}_{\text{HMT}}(g(\mathbf{x}^u))\big), \hat{y}_i \Big)
$$
$$
+ \ell_{CE}\Big(E_i\big(\text{BN}_{\text{MT}}(g(\mathbf{x}^u))\big), \hat{y}_i \Big) \cdot \mathbb{I}\Big(\hat{y}_i \in \mathcal{M} \cup \mathcal{T} \Big)
$$
$$
+ \ell_{CE}\Big(E_i\big(\text{BN}_{\text{T}}(g(\mathbf{x}^u))\big), \hat{y}_i \Big) \cdot \mathbb{I}\Big(\hat{y}_i \in \mathcal{T} \Big)\Bigg]. \tag{5}
$$

$\mathbb{I}(\cdot)$ denotes the class mask, and S equals the number of terms that used in $L_{CPE,i}^{unsup}$. Note that we only utilize CBN in computing unsupervised loss because the class distribution of labeled set is always fixed. The illustration of CBN mechanism can be seen in Fig. 2. We will show the effectiveness of CBN in experimental sections.

Experiments

We follow previous LTSSL works to evaluate our CPE approach on various dataset benchmarks, and make comparisons with baseline algorithms. Also, we conduct ablation studies on CPE and provide necessary analysis.

Experimental Setting

Dataset We evaluate on three widely-used LTTSL datasets: CIFAR-10-LT (Krizhevsky et al. 2009), CIFAR-100-LT (Krizhevsky et al. 2009), and STL-10 (Coates, Ng, and Lee 2011). We follow the settings in DASO (Oh, Kim, and Kweon 2022) and ACR (Wei and Gan 2023).

- **CIFAR-10-LT.** We test with four settings in the *consistent* case: $(N_1, M_1) = (1500, 3000)$ and $(N_1, M_1) = (500, 4000)$, with common imbalance ratio γ being 100 or 150. In the *uniform* case, we test with $(N_1, M_1) = (1500, 300)$ and $(N_1, M_1) = (500, 400)$, with γ_l being 100 and γ_u being 1. In the *inverse* case, we test with $(N_1, M_1) = (1500, 30)$ and $(N_1, M_1) = (500, 40)$, with γ_l being 100 and γ_u being $1/100$.

- **CIFAR-100-LT.** We evaluate on CIFAR-100-LT in both *consistent* and *inverse* cases. For the former, we set $(N_1, M_1) = (150, 300)$ with γ being 10 or 15. For the latter, $(N_1, M_1) = (150, 30)$ with (γ_l, γ_u) being $(10, 1/10)$.

- **STL-10-LT.** We set N_1 as 150 and imbalance ratio γ_l as 10 or 20 for the labeled set, and use the original unlabeled set which is already imbalanced and the ground-truth labels of which are not provided by the authors.

There is no overlap between labeled and unlabeled sets.

Baselines We compare with several LTSSL algorithms published in top-conferences/journals in the past few years. These baseline algorithms include DARP (Kim et al. 2020), CReST (Wei et al. 2021) and its variant CReST+ (Wei et al. 2021), ABC (Lee, Shin, and Kim 2021), DASO (Oh, Kim, and Kweon 2022), CoSSL (Fan et al. 2022), SAW (Lai et al. 2022), Adsh (Guo and Li 2022), DePL (Wang et al. 2022a), RDA (Duan et al. 2022), and ACR (Wei and Gan 2023). For a fair comparison, we test these baselines and our CPE algorithm on the widely-used codebase USB[1]. We use the same dataset splits and common hyper-parameters for training.

Implementation Details

We follow the default hyper-parameters in USB and set the size of labeled and unlabeled data batch as 64 and 128, respectively. We resize all input images to 32×32. We fix the loss weight λ at 2, and fix the confidence threshold ρ to 0.95. We use the SGD optimizer with learning rate $3e-2$, momentum 0.9, and weight decay $5e-4$. As DNN architecture, we use WRN-28-2 (Zagoruyko and Komodakis 2016) without any pre-training. We set the intensities of logit adjustment (τ_1, τ_2, τ_3) as $(0, 2, 4)$ for all datasets and experimental settings in our CPE algorithm. For CBN mechanism in CPE, we consider the first one-third of all classes as head classes, the last one-third as tail classes, and the remaining as medium classes. We repeat each experiment over three different random seeds, and report the mean and standard deviation of performances. We conduct the experiments on a single GPU of NVIDIA RTX A6000 using PyTorch $v1.10$.

Main Results

Consistent case. We consider the *consistent* case on CIFAR-10-LT and CIFAR-100-LT. We compare CPE and baseline algorithms based on the widely-used SSL method, FixMatch (Sohn et al. 2020).As can be seen in Table 2, our CPE method achieves better performance than all the previous baselines on CIFAR-10-LT under various settings. For example, given $N_1 = 1500$, $M_1 = 3000$, and $\gamma = 150$,

---

[1]https://github.com/microsoft/Semi-supervised-learning

Dataset	CIFAR-10-LT				CIFAR-100-LT	
N_1	1500		500		150	
M_1	3000		4000		300	
$\gamma = \gamma_l = \gamma_u$	100	150	100	150	10	15
Supervised	$63.62_{\pm0.40}$	$59.82_{\pm0.32}$	$47.62_{\pm0.87}$	$43.88_{\pm1.61}$	$48.01_{\pm0.45}$	$45.37_{\pm0.54}$
FixMatch (Sohn et al. 2020)	$76.49_{\pm0.72}$	$72.15_{\pm0.94}$	$73.14_{\pm1.03}$	$65.68_{\pm0.67}$	$57.56_{\pm0.47}$	$53.97_{\pm0.17}$
w/ DARP (Kim et al. 2020)	$77.37_{\pm0.50}$	$74.02_{\pm0.06}$	$71.12_{\pm0.82}$	$65.63_{\pm0.63}$	$56.14_{\pm0.46}$	$52.81_{\pm0.50}$
w/ CReST (Wei et al. 2021)	$79.90_{\pm0.33}$	$74.70_{\pm0.53}$	$77.69_{\pm0.71}$	$68.20_{\pm0.33}$	$58.56_{\pm0.34}$	$55.43_{\pm0.17}$
w/ CReST+ (Wei et al. 2021)	$79.60_{\pm0.06}$	$75.39_{\pm0.42}$	$78.70_{\pm0.40}$	$72.73_{\pm2.26}$	$58.19_{\pm0.37}$	$55.39_{\pm0.23}$
w/ ABC (Lee, Shin, and Kim 2021)	$84.01_{\pm0.15}$	$80.94_{\pm0.85}$	$79.40_{\pm0.88}$	$69.50_{\pm1.86}$	$58.25_{\pm0.20}$	$55.38_{\pm0.47}$
w/ DASO (Oh, Kim, and Kweon 2022)	$78.87_{\pm0.80}$	$74.92_{\pm0.36}$	$73.63_{\pm0.46}$	$67.13_{\pm1.06}$	$58.16_{\pm0.21}$	$54.82_{\pm0.53}$
w/ CoSSL (Fan et al. 2022)	$82.35_{\pm0.79}$	$79.00_{\pm0.41}$	$75.82_{\pm0.61}$	$70.56_{\pm0.55}$	$58.00_{\pm0.39}$	$55.49_{\pm0.43}$
w/ SAW (Lai et al. 2022)	$80.93_{\pm0.31}$	$77.67_{\pm0.14}$	$75.20_{\pm1.01}$	$68.51_{\pm0.16}$	$57.55_{\pm0.45}$	$54.00_{\pm0.65}$
w/ Adsh (Guo and Li 2022)	$78.43_{\pm0.54}$	$73.96_{\pm0.47}$	$75.97_{\pm0.68}$	$66.55_{\pm2.94}$	$58.65_{\pm0.36}$	$54.55_{\pm0.40}$
w/ DePL (Wang et al. 2022a)	$80.65_{\pm0.52}$	$76.58_{\pm0.12}$	$76.98_{\pm1.70}$	$71.95_{\pm2.54}$	$57.08_{\pm0.29}$	$53.89_{\pm0.44}$
w/ RDA (Duan et al. 2022)	$78.13_{\pm0.27}$	$72.88_{\pm0.35}$	$74.59_{\pm0.92}$	$68.59_{\pm0.61}$	$58.13_{\pm0.45}$	$54.54_{\pm0.50}$
w/ ACR (Wei and Gan 2023)	$84.18_{\pm0.52}$	$81.81_{\pm0.49}$	$\mathbf{81.01_{\pm0.42}}$	$76.72_{\pm1.13}$	$59.83_{\pm0.07}$	$56.91_{\pm0.09}$
w/ **CPE (Ours)**	$\mathbf{84.44_{\pm0.29}}$	$\mathbf{82.25_{\pm0.34}}$	$\underline{80.68_{\pm0.96}}$	$\mathbf{76.77_{\pm0.53}}$	$\mathbf{59.83_{\pm0.16}}$	$\mathbf{57.00_{\pm0.51}}$

Table 2: Top-1 test set accuracy of previous LTSSL algorithms and our proposed CPE under consistent class distributions, *i.e.*, $\gamma_l = \gamma_u$, 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 in underline.

Dataset	CIFAR-10-LT				CIFAR-100-LT	STL-10-LT	
N_1	1500		500		150	150	
M_1	300	30	400	40	30	$\approx$100k	
γ_l	100				10	10	20
γ_u	1	1/100	1	1/100	1/10	N/A	N/A
Supervised	$63.62_{\pm0.40}$	$63.62_{\pm0.40}$	$47.62_{\pm0.87}$	$47.62_{\pm0.87}$	$48.01_{\pm0.45}$	$46.85_{\pm1.65}$	$41.60_{\pm0.77}$
FixMatch (Sohn et al. 2020)	$73.27_{\pm1.25}$	$68.92_{\pm0.79}$	$66.47_{\pm0.84}$	$62.52_{\pm0.93}$	$57.56_{\pm0.47}$	$66.56_{\pm1.02}$	$56.29_{\pm4.00}$
w/ DARP (Kim et al. 2020)	$73.52_{\pm1.19}$	$70.36_{\pm1.55}$	$65.80_{\pm0.67}$	$62.16_{\pm1.10}$	$56.40_{\pm0.21}$	$63.74_{\pm0.54}$	$56.03_{\pm1.81}$
w/ CReST (Wei et al. 2021)	$\underline{85.14_{\pm0.19}}$	$86.71_{\pm0.39}$	$68.20_{\pm0.33}$	$76.37_{\pm3.84}$	$60.07_{\pm0.24}$	$65.52_{\pm1.01}$	$61.38_{\pm1.75}$
w/ CReST+ (Wei et al. 2021)	$82.68_{\pm0.93}$	$73.98_{\pm1.34}$	$72.73_{\pm2.26}$	$63.72_{\pm0.87}$	$59.53_{\pm0.34}$	$66.27_{\pm0.59}$	$62.63_{\pm1.69}$
w/ ABC (Lee, Shin, and Kim 2021)	$81.45_{\pm0.43}$	$83.45_{\pm0.54}$	$69.50_{\pm1.86}$	$79.21_{\pm0.44}$	$59.24_{\pm0.17}$	$70.64_{\pm0.89}$	$65.68_{\pm3.06}$
w/ DASO (Oh, Kim, and Kweon 2022)	$76.01_{\pm0.59}$	$74.47_{\pm0.60}$	$67.54_{\pm0.60}$	$65.08_{\pm0.90}$	$59.25_{\pm0.23}$	$69.31_{\pm0.91}$	$62.45_{\pm2.23}$
w/ CoSSL (Fan et al. 2022)	$79.90_{\pm0.76}$	$77.47_{\pm0.56}$	$74.31_{\pm0.98}$	$73.26_{\pm0.78}$	$57.77_{\pm0.31}$	$71.44_{\pm0.45}$	$69.01_{\pm0.80}$
w/ SAW (Lai et al. 2022)	$81.50_{\pm0.44}$	$76.73_{\pm0.66}$	$68.51_{\pm0.16}$	$70.04_{\pm1.58}$	$58.12_{\pm0.34}$	$69.30_{\pm0.69}$	$65.80_{\pm1.22}$
w/ Adsh (Guo and Li 2022)	$76.55_{\pm0.13}$	$70.45_{\pm0.70}$	$66.55_{\pm2.94}$	$65.64_{\pm1.82}$	$55.49_{\pm0.55}$	$69.35_{\pm1.12}$	$64.82_{\pm1.41}$
w/ DePL (Wang et al. 2022a)	$72.99_{\pm0.45}$	$74.53_{\pm0.61}$	$71.95_{\pm2.54}$	$69.67_{\pm1.34}$	$57.31_{\pm0.55}$	$69.46_{\pm0.62}$	$65.93_{\pm1.22}$
w/ RDA (Duan et al. 2022)	$78.41_{\pm0.19}$	$72.62_{\pm0.55}$	$68.59_{\pm0.61}$	$68.99_{\pm0.26}$	$57.79_{\pm0.43}$	$72.63_{\pm0.26}$	$\underline{69.24_{\pm1.69}}$
w/ ACR (Wei and Gan 2023)	$84.61_{\pm0.50}$	$86.29_{\pm0.19}$	$\underline{80.10_{\pm1.21}}$	$82.65_{\pm0.31}$	$\mathbf{60.92_{\pm0.40}}$	$\mathbf{73.40_{\pm0.73}}$	$67.51_{\pm1.63}$
w/ **CPE (Ours)**	$\mathbf{85.86_{\pm0.40}}$	$\mathbf{87.09_{\pm0.14}}$	$\mathbf{82.32_{\pm0.43}}$	$\mathbf{83.88_{\pm0.18}}$	$\underline{60.83_{\pm0.30}}$	$\underline{73.07_{\pm0.47}}$	$\mathbf{69.60_{\pm0.20}}$

Table 3: Top-1 test set accuracy of previous LTSSL algorithms and our proposed CPE under uniform and inverse class distributions, *i.e.*, $\gamma_l \neq \gamma_u$, on CIFAR10-LT, CIFAR100-LT, and STL-10-LT benchmarks. The network architecture is WRN-28-2 trained from scratch. We highlight the best number in bold and the second best in underline.

CPE surpasses the previous SOTA method (ACR) by 0.44 percentage points (*pp*), and all other baselines by 1.31 *pp*. On CIFAR-100-LT, the performances of CPE is on par with ACR (since the imbalance ratio reduces to 10 and the task becomes easier), but beats other baselines by >1.51 *pp*.

Uniform and inverse case. We consider the *uniform* and *inverse* case, *i.e.*, $\gamma_l \neq \gamma_u$. In Table 3, we show that for almost all baselines, the more γ_u differs from γ_l, the worse is the

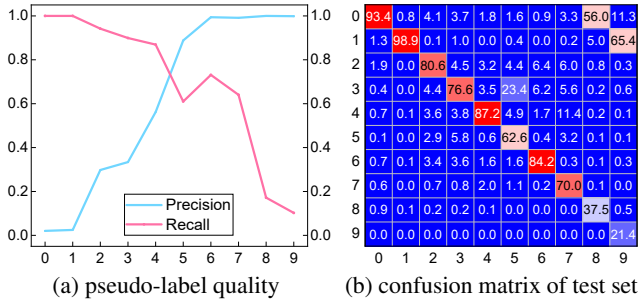(a) pseudo-label quality   (b) confusion matrix of test set

Figure 4: Case study: The first head trained with regular loss generates imbalanced pseudo-labels (low recall on tail classes), leads to a degeneration on the representation ability of tail classes, and finally hurt the test accuracy on tail classes. In contrast, low precision on head classes has much less impacts on the test accuracy.

performance. This is consistent with our findings that DNN classifier tends to predict unlabeled data as head classes, thus resulting in more erroneous when tail classes *w.r.t.* labeled set are no longer tail classes in unlabeled set. In comparison, our CPE method can consistently and largely outperform baseline algorithms on CIFAR-10-LT, validating the effectiveness of CPE to cope with varying class distributions of unlabeled set. Concretely, CPE surpasses ACR by 1.22 *pp* and other baselines by up to >8.01 *pp* with $(N_1, M_1) = (500, 400)$ and $(\gamma_l, \gamma_u) = (100, 1)$. On CIFAR-100-LT and STL-10-LT, CPE achieves comparable performances with ACR, surpassing other baselines by >1 *pp*.

Ablation Studies and Analysis

The effect of each block. We examine the effectiveness of CPE in Table 4. We observe that the multi-expert architecture brings significant improvements over FixMatch. The effectiveness of CBN is better reflected in the *inverse* case, where our method improves over FixMatch by 18 *pp*. In Table 4, we show the effects of multi-experts and CBN, showing that each block is important to reach the best results.

How do the low-precision pseudo-labels affect representation learning? Considering that CPE utilizes the pseudo-labels predicted by all three experts for representation learning, one may argue that CPE will incur unnecessary low-precision pseudo-labels and may harm the performance of the feature extractor. This is because only one expert can generate high-quality pseudo-labels while the others cannot (see Table 1). Thus, we conduct an experiment to observe how the low-precision pseudo-labels affect representation learning. Following the two-head LTSSL training paradigm, we train the first head with regular cross entropy loss, and train the second head with logit adjustment while preventing its gradient back-propagating to the shared feature extractor. We evaluate on CIFAR-10-LT in the *inverse* case with (γ_l, γ_u) being $(100, 1/100)$, and choose the best τ for the logit adjustment of second head via grid-search.

As we illustrate in Fig. 4, the first head predominantly predicts unlabeled data as head classes, yielding low precision for head classes and reduced recall for tail classes. Consequently, tail class representation capability diminishes, lead-
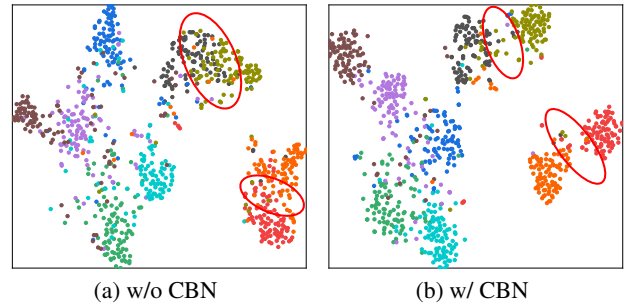

(a) w/o CBN   (b) w/ CBN

Figure 5: T-SNE visualization of extracted features of unlabeled data in CIFAR-10-LT with $\gamma_l = 100$ and $\gamma_u = 1/100$. We can see that the Classwise Batch Normalization (CBN) mechanism can lead to more compact feature clusters, which can be classified easier (highlighted by red circles).

FixMatch		γ_u	
w/ Three Experts	w/ CBN	100	1/100
		76.49	68.92
✓		84.02	86.04
	✓	79.00	81.22
✓	✓	**84.79**	**87.28**

Table 4: Ablation on the effectiveness of multiple experts and CBN. The dataset is CIFAR-10-LT with $(N_1, M_1) = (1500, 3000)$ and the imbalance ratio $\gamma_l = 100$.

ing to lower tail class accuracy. Notably, the modest precision drop for head classes minimally affects head class accuracy, implying that low-precision pseudo-labels barely harm the learning in CPE.

Effect of CBN mechanism. As explained in Sec , we propose CBN mechanism to handle varying feature distributions for CPE. To illustrate the effect of CBN, we visualize the extracted features of all classes in Fig. 5. As can be seen, CBN helps construct a discriminative feature space, which can boost the classification performances.

Conclusion

In this work, we address the LTSSL task, focusing on the influence of pseudo-labels on representation learning. Contrary to conventional methods reliant on a single classifier for pseudo-label generation, we address the challenge of adapting to varying class distributions within the unlabeled training set. To improve pseudo-label quality, we introduce the CPE algorithm. It comprises three experts working within a shared feature extractor framework, each dedicated to modeling a distinct class distribution. We achieve this specialization by applying varying logit adjustments. Furthermore, we introduce classwise batch-normalization that effectively handles feature distribution mismatch between head and tail classes. We validate our proposed model's efficacy through extensive experimentation across three LTSSL datasets, improving the state-of-the-art performance by up to 2.2 *pp*.

Acknowledgements

References

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.

Cai, J.; Wang, Y.; and Hwang, J.-N. 2021. Ace: Ally complementary experts for solving long-tailed recognition in one-shot. In *International Conference on Computer Vision (ICCV)*, 112–121.

Cao, K.; Wei, C.; Gaidon, A.; Arechiga, N.; and Ma, T. 2019. Learning imbalanced datasets with label-distribution-aware margin loss. *Neural Information Processing Systems (NeurIPS)*, 32.

Chawla, N. V.; Bowyer, K. W.; Hall, L. O.; and Kegelmeyer, W. P. 2002. SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16: 321–357.

Chen, H.; Fan, Y.; Wang, Y.; Wang, J.; Schiele, B.; Xie, X.; Savvides, M.; and Raj, B. 2022a. An embarrassingly simple baseline for imbalanced semi-supervised learning. *arXiv preprint arXiv:2211.11086*.

Chen, H.; Tao, R.; Fan, Y.; Wang, Y.; Wang, J.; Schiele, B.; Xie, X.; Raj, B.; and Savvides, M. 2022b. SoftMatch: Addressing the quantity-quality tradeoff in semi-supervised Learning. In *International Conference on Learning Representations (ICLR)*.

Coates, A.; Ng, A.; and Lee, H. 2011. An analysis of single-layer networks in unsupervised feature learning. In *Artificial Intelligence and Statistics (AISTATS)*, 215–223. JMLR Workshop and Conference Proceedings.

Cui, J.; Liu, S.; Tian, Z.; Zhong, Z.; and Jia, J. 2022. Reslt: Residual learning for long-tailed recognition. *IEEE transactions on pattern analysis and machine intelligence*, 45(3): 3695–3706.

Cui, Y.; Jia, M.; Lin, T.-Y.; Song, Y.; and Belongie, S. 2019. Class-balanced loss based on effective number of samples. In *Computer Vision and Pattern Recognition (CVPR)*, 9268–9277.

Duan, Y.; Qi, L.; Wang, L.; Zhou, L.; and Shi, Y. 2022. RDA: Reciprocal distribution alignment for robust semi-supervised learning. In *European Conference on Computer Vision (ECCV)*, 533–549. Springer.

Fan, Y.; Dai, D.; Kukleva, A.; and Schiele, B. 2022. Cossl: Co-learning of representation and classifier for imbalanced semi-supervised learning. In *Computer Vision and Pattern Recognition (CVPR)*, 14574–14584.

Guo, L.-Z.; and Li, Y.-F. 2022. Class-imbalanced semi-supervised learning with adaptive thresholding. In *International Conference on Machine Learning (ICML)*, 8082–8094. PMLR.

Hong, Y.; Han, S.; Choi, K.; Seo, S.; Kim, B.; and Chang, B. 2021. Disentangling label distribution for long-tailed visual recognition. In *Computer Vision and Pattern Recognition (CVPR)*, 6626–6636.

Johnson, J. M.; and Khoshgoftaar, T. M. 2019. Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1): 1–54.

Kang, B.; Xie, S.; Rohrbach, M.; Yan, Z.; Gordo, A.; Feng, J.; and Kalantidis, Y. 2019. Decoupling representation and classifier for long-tailed recognition. In *International Conference on Learning Representations (ICLR)*.

Khan, S. H.; Hayat, M.; Bennamoun, M.; Sohel, F. A.; and Togneri, R. 2017. Cost-sensitive learning of deep feature representations from imbalanced data. *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, 29(8): 3573–3587.

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. *Neural Information Processing Systems (NeurIPS)*, 33: 14567–14579.

Krizhevsky, A.; et al. 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 (ICML)*, 11828–11843. PMLR.

Lee, H.; Shin, S.; and Kim, H. 2021. Abc: Auxiliary balanced classifier for class-imbalanced semi-supervised learning. *Neural Information Processing Systems (NeurIPS)*, 34: 7082–7094.

Liu, Z.; Miao, Z.; Zhan, X.; Wang, J.; Gong, B.; and Yu, S. X. 2019. Large-scale long-tailed recognition in an open world. In *Computer Vision and Pattern Recognition (CVPR)*, 2537–2546.

Menon, A. K.; Jayasumana, S.; Rawat, A. S.; Jain, H.; Veit, A.; and Kumar, S. 2020. Long-tail learning via logit adjustment. In *International Conference on Learning Representations (ICLR)*.

Oh, Y.; Kim, D.-J.; and Kweon, I. S. 2022. Daso: Distribution-aware semantics-oriented pseudo-label for imbalanced semi-supervised learning. In *Computer Vision and Pattern Recognition (CVPR)*, 9786–9796.

Pise, N. N.; and Kulkarni, P. 2008. A survey of semi-supervised learning methods. In *Computational Intelligence and Security (CIS)*, volume 2, 30–34. IEEE.

Reed, W. J. 2001. The Pareto, Zipf and other power laws. *Economics letters*, 74(1): 15–19.

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. *Neural Information Processing Systems (NeurIPS)*, 33: 596–608.

Wang, X.; Lian, L.; Miao, Z.; Liu, Z.; and Yu, S. 2020. Long-tailed recognition by routing diverse distribution-aware experts. In *International Conference on Learning Representations (ICLR)*.

Wang, X.; Wu, Z.; Lian, L.; and Yu, S. X. 2022a. Debiased learning from naturally imbalanced pseudo-labels. In *Computer Vision and Pattern Recognition (CVPR)*, 14647–14657.

Wang, Y.; Chen, H.; Heng, Q.; Hou, W.; Fan, Y.; Wu, Z.; Wang, J.; Savvides, M.; Shinozaki, T.; Raj, B.; et al. 2022b. FreeMatch: Self-adaptive thresholding for semi-supervised learning. In *International Conference on Learning Representations (ICLR)*.

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 *Computer Vision and Pattern Recognition (CVPR)*, 10857–10866.

Wei, T.; and Gan, K. 2023. Towards realistic long-tailed semi-supervised learning: Consistency is all you need. In *Computer Vision and Pattern Recognition (CVPR)*, 3469–3478.

Xie, C.; Tan, M.; Gong, B.; Wang, J.; Yuille, A. L.; and Le, Q. V. 2020. Adversarial examples improve image recognition. In *Computer Vision and Pattern Recognition (CVPR)*, 819–828.

Zagoruyko, S.; and Komodakis, N. 2016. Wide residual networks. In *British Machine Vision Conference (BMVC)*. British Machine Vision Association.

Zhang, B.; Wang, Y.; Hou, W.; Wu, H.; Wang, J.; Okumura, M.; and Shinozaki, T. 2021. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. *Neural Information Processing Systems (NeurIPS)*, 34: 18408–18419.

Zhang, Y.; Hooi, B.; Hong, L.; and Feng, J. 2022. Self-supervised aggregation of diverse experts for test-agnostic long-tailed recognition. *Neural Information Processing Systems (NeurIPS)*, 35: 34077–34090.