Appendix for the Paper
“CDMAD: Class-Distribution-Mismatch-Aware Debiasing for Class-Imbalanced Semi-Supervised Learning”

Hyuck Lee Heeyoung Kim
Department of Industrial and Systems Engineering, KAIST
Daejeon 34141, Republic of Korea
{dlgur0921, heeyoungkim}@kaist.ac.kr

A. Core part of the code for CDMAD

```
with torch.no_grad():
    white = torch.ones((1, 3, 32, 32)).cuda()
    biaseddegree, _ = model(white)
    outputs_u, _ = model(inputs_u)
    if epoch>args.debiasstart:
        outputs_u = outputs_u - biaseddegree.detach()
    targets_u2 = F.softmax(outputs_u).detach()
```

Figure 1. Code for refining pseudo-labels using CDMAD

Fig. 1 presents a core part of the code for CDMAD to refine the biased pseudo-labels of the base SSL algorithm. As we can see in Fig. 1, CDMAD is very easy to implement. We simply need to calculate the logits for an image without any patterns (solid color image) and then subtract them from the logits for unlabeled samples. Biased class predictions on test samples are refined in a similar way.

B. Further related works

Semi-supervised learning (SSL) algorithms use unlabeled data for training when labeled samples are insufficient. Entropy minimization [15] encourages the class predictions on unlabeled samples to be confident by directly minimizing entropy or using pseudo-labels [33]. Consistency regularization [40, 42, 46] encourages the class predictions on two augmented versions of an unlabeled sample to be consistent. FixMatch [45] and ReMixMatch [3] conduct entropy minimization and consistency regularization using strong data augmentation techniques [9, 12]. ReMixMatch also conducts Mixup regularization [2, 48] and self-supervised learning with rotation [14]. CoMatch [36] proposed graph-based contrastive learning using embedding and pseudo-label graphs. Recently, curriculum pseudo-labeling that considers the learning status for each class was proposed by FlexMatch [59] and extended in Adsh [16], SoftMatch [7] and FreeMatch [53].

Class-imbalanced learning (CIL) algorithms mitigate class imbalance to improve classification performance for minority classes. Resampling techniques [1, 6, 17, 22] balance the number of each class samples, and reweighting techniques [11, 19, 21, 39, 54] balance the loss for each class. Cao et al. [5] and Ren et al. [43] proposed losses that minimize a generalization error bound, and Kim et al. [27], Yin et al. [57] transferred knowledge from the data of the majority classes to the data of minority classes. Kang et al. [25] decoupled representation and classifier learning. Menon et al. [38] proposed post-hoc logit-adjustment and loss, which is Fisher consistent for minimizing the balanced error. Recently, CIL algorithms based on contrastive learning [10, 23, 24, 24, 37, 49] and multi-expert learning [4, 35, 51, 56, 60, 61] received considerable attention.

C. Data augmentation techniques

CDMAD uses data augmentation techniques utilized in FixMatch, ReMixMatch, and previous CISSL algorithms. Specifically, CDMAD uses random horizontal flipping and random cropping as weak data augmentation techniques and uses Cutout [12] and RandomAugment [9] as strong data augmantation techniques. Random horizontal flipping and cropping flips and crops images, respectively. We implemented these weak data augmentation techniques using torchvision.transforms library. Cutout randomly masks out the square region of the image during training, which prevents the network from focusing on non-general features. The purpose of RandomAugment is to teach the network invariances. RandomAugment is a data augmentation technique that automatically searches for improved augmentation policies, where the search space of the policy consists of many sub-policies, one of which is randomly chosen for each data point at each iteration. A sub-policy is composed of basic data-augmentation techniques, such as shearing, rotation, and translation. We imple-

(a) Random horizontal flipping 1 (b) Random horizontal flipping 2 (c) Random cropping 1 (d) Random cropping 2

(e) Cutout 1 (f) Cutout 2 (g) RandomAugment 1 (h) RandomAugment 2

Figure 2. Example images augmented using each data augmentation technique

mented Cutout and RandomAugment using the code from https://github.com/ildoonet/pytorch-randaugment. Example images augmented using each data augmentation technique are presented in Fig. 2.

D. Training losses of FixMatch [45] and ReMixMatch [3]

Training losses of FixMatch [45] and ReMixMatch [3] on a minibatch for labeled set $\mathcal{MX}$ and a minibatch for unlabeled set $\mathcal{MU}$ can be expressed as follows:

$$\begin{aligned} loss_F\left(\mathcal{MX}, \mathcal{MU}, \hat{q}, \tau; \theta\right) = Con(\mathcal{MU}, \hat{q}, \tau; \theta) \\ + Sup(\mathcal{MX}; \theta), \end{aligned} \tag{1}$$

$$\begin{aligned} loss_R\left(\mathcal{MX}, \mathcal{MU}, \bar{q}; \theta\right) = Mix(\mathcal{MX}, \mathcal{MU}, \bar{q}; \theta) \\ + Con(\mathcal{MU}, \bar{q}; \theta) + Rot(\mathcal{MU}, r; \theta), \end{aligned} \tag{2}$$

where $\hat{q}$ and $\bar{q}$ denote the concatenations of $\hat{q}_b$ and $\bar{q}_b$, $b = 1, \ldots, \mu B$, respectively, $Con(\mathcal{MU}, \hat{q}, \tau; \theta)$ and $Con(\mathcal{MU}, \bar{q}; \theta)$ denote the consistency regularization loss with and without the confidence threshold τ, respectively, $Sup(\mathcal{MX}; \theta)$ denotes the supervised loss for weakly augmented labeled data points, $Mix(\mathcal{MX}, \mathcal{MU}, \bar{q}; \theta)$ denotes the mix-up regularization loss, and $Rot(\mathcal{MU}, r; \theta)$ denotes the rotation loss with the rotated degree r.

Each loss term in Eq (1) and (2) of the main paper is detailed as follows:

$$\begin{aligned} Con(\mathcal{MU}, \hat{q}, \tau; \theta) = \\ \frac{1}{\mu B} \sum_{u_b^m \in \mathcal{MU}} \mathbf{I}(\max(\hat{q}_b) \geq \tau) \mathbf{H}(P_\theta(y|\mathcal{A}(u_b^m)), \hat{q}_b), \end{aligned} \tag{3}$$

$$Con(\mathcal{MU}, \bar{q}; \theta) = \frac{1}{\mu B} \sum_{u_b^m \in \mathcal{MU}} \mathbf{H}(P_\theta(y|\mathcal{A}(u_b^m)), \bar{q}_b), \tag{4}$$

$$Sup(\mathcal{MX}; \theta) = \frac{1}{B} \sum_{x_b^m \in \mathcal{MX}} \mathbf{H}(P_\theta(y|\alpha(x_b^m)), p_b^m), \tag{5}$$

$$\begin{aligned} Mix(\mathcal{MX}, \mathcal{MU}, \bar{q}; \theta) = \frac{1}{B} \sum_{mx_b^m \in \mathcal{MX}'} \mathbf{H}(P_\theta(y|mx_b^m), mp_b^m) + \\ \frac{1}{\mu B} \sum_{mu_b^m \in \mathcal{MU}'} \mathbf{H}(P_\theta(y|mu_b^m), m\bar{q}_b), \end{aligned} \tag{6}$$

$$Rot(\mathcal{MU}, r; \theta) = \frac{1}{\mu B} \sum_{u_b^m \in \mathcal{MU}} \mathbf{H}(P_{\theta'}(d|\mathcal{R}(u_b^m, r)), r), \tag{7}$$

where $\mathbf{H}(\cdot, \cdot)$ denotes the cross-entropy loss, p_b^m is one-hot encoded y_b^m, $\mathcal{MX}'$ and $\mathcal{MU}'$ are generated by mixup operation with strongly augmented $\mathcal{MX}$ and $\mathcal{MU}$, respectively, mx_b^m denotes a mixed-labeled image, mp_b^m denotes a mixed label, mu_b^m denotes a mixed-unlabeled image, $m\bar{q}_b$ denotes a mixed pseudo-label, $\mathcal{R}(u_b^m, r)$ denotes the rotated

u_b^m with degree r, and $P_{\theta'}(\hat{r}|\mathcal{R}(u_b^m, r))$ denotes the prediction of rotated degree r using network parameters θ' that mostly overlap with θ.

E. Illustration of refining biased class predictions on test samples using CDMAD

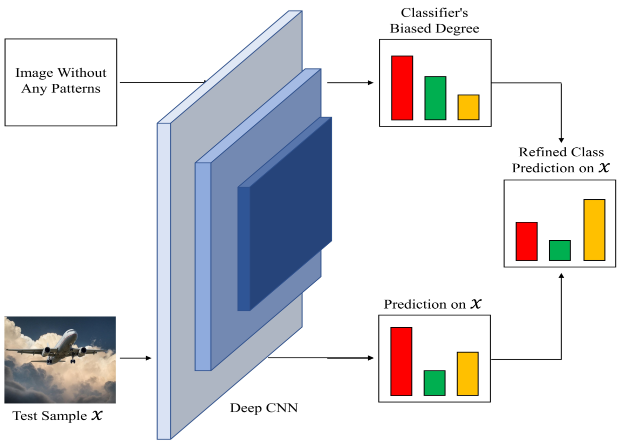


Figure 3. Refinement of biased class predictions on test samples using CDMAD

Fig. 3 presents refinement process of the biased class predictions on test samples using the CDMAD.

F. Pseudo code of the proposed algorithm

The pseudo code that describes both training and test phases of the proposed algorithm is presented in Algorithm 1.

G. Performance measures

Following previous CISSL studies, we used balanced accuracy (bACC) [19], geometric mean (GM) [30] as performance measures for the experiments in Section 4.2. Each performance measure is detailed as follows. **Balanced accuracy (bACC)** is the average of per-class accuracy. When the test set is class-balanced, bACC equals to the overall test accuracy. bACC is also referred to as the averaged class recall in previous CISSL studies [55] and [13]. **Geometric mean (GM)** is obtained by multiplying the Cth root of per-class accuracy, where C denotes the number of classes. GM equals to the overall test accuracy when all classes have the same per-class accuracy.

H. Further details about datasets and experimental setup

CIFAR-10-LT and CIFAR-100-LT are long-tailed datasets artificially generated from CIFAR-10 and CIFAR-100 [29], respectively, with $N_k = N_1 \times (N_C/N_1)^{\frac{k-1}{C-1}}$ and $M_k = M_1 \times (M_C/M_1)^{\frac{k-1}{C-1}}$. For CIFAR-10-LT, we assumed that γ_u is known and equal to γ_l while varying both γ_l and γ_u as 50, 100 and 150. We then assumed that γ_u is unknown and different from γ_l while setting γ_l to 100 and varying γ_u as 1, 50 and 150. We set $N_1 = 1500$ and $M_1 = 3000$. For CIFAR-100-LT, we assumed that γ_u is known and equal to γ_l while varying both γ_l and γ_u as 20, 50 and 100. We set $N_1 = 150$ and **STL-10-LT** is a long-tailed dataset created from STL-10 [8], where the number of labeled samples exponentially decreases from N_1 to N_C. We conducted experiments with unknown γ_u while varying γ_l as 10 and 20. We set N_1 to 450 and used all $100,000$ unlabeled samples. **Small-ImageNet-127** is a down-sampled version of ImageNet-127 [20], created by grouping ImageNet [44] into 127 classes based on WordNet hierarchy. The training set of ImageNet-127 consists of a total of 1,281,167 images and is imbalanced with the class imbalanced ratio of 286. Fan et al. [13] created two versions of this dataset by down-sampling the images to 32×32 and 64×64, and randomly selected 10% of the training samples of each class as a labeled set and used the remaining as an unlabeled set. We conducted experiments on both versions under the assumption that γ_u is known and equal to γ_l. Similar to Fan et al. [13], Wei et al. [55], we conducted experiments using only FixMatch because of an excessive training cost. The test set of Small-ImageNet-127 is also class-imbalanced.

We used the Adam optimizer [28]. We used the exponential moving average (EMA) of the network parameters for each iteration to evaluate the classification performance. We used Wide ResNet-28-2 [58] as a deep CNN for CIFAR-10-LT, CIFAR-100-LT, and STL-10-LT, and ResNet-50 [18] for Small-ImageNet-127.

For the experiments using FixMatch, we set the minibatch size B to 32, relative size of the unlabeled to labeled minibatches μ to 2, and learning rate of the optimizer to $1.5 * 10^{-3}$. We trained FixMatch for 500 epochs, where 1 epoch= 500 iterations. For the experiments using ReMixMatch, we set the minibatch size B to 64, relative size of the unlabeled to labeled minibatches μ to 2, and learning rate of the optimizer to $2 * 10^{-3}$. We trained ReMixMatch for 300 epochs. For the experiments on CIFAR-100, we set the weight decay parameter of L2 regularization (for EMA parameters) to 0.08 because CIFAR-100 has significantly many classes compared to the total number of training samples. For the experiments on CIFAR-10, STL-10, and Small-ImageNet-127, we set the weight decay parameter of L2 regularization to 0.04 when the number of training samples is smaller than $3 * 10^4$, while we set it to 0.01 and 0.02 for FixMatch and ReMixMatch, respectively, when the number of training samples is larger than $3 * 10^4$, because L2 regularization becomes ineffective as the number of training samples increases. We confirmed that the training of the proposed algorithm took less time than the

Algorithm 1 Pseudo code of the proposed algorithm

Input: Labeled set $\mathcal{X}$, unlabeled set $\mathcal{U}$, test set $\mathcal{X}^{test}$, network parameters θ
Output: Refined class predictions on test samples $f_{\theta}^{*}\left(x_{k}^{test}\right)$ for $k=1,\ldots,K$
while training **do**
  Generate minibatches $\mathcal{MX}=\{(x_b^m, y_b^m): b\in(1,\ldots,B)\}\subset\mathcal{X}$ and $\mathcal{MU}=\{(u_b^m): b\in(1,\ldots,\mu B)\}\subset\mathcal{U}$
  Produce logits for a solid color image $g_{\theta}(\mathcal{I})$
  Produce logits for weakly augmented unlabeled samples $g_{\theta}\left(\alpha\left(u_b^m\right)\right)$ for $b=1,\ldots,\mu B$
  Obtain refined logits $g_{\theta}^{*}\left(\alpha\left(u_b^m\right)\right)=g_{\theta}\left(\alpha\left(u_b^m\right)\right)-g_{\theta}(\mathcal{I})$ for $b=1,\ldots,\mu B$
  Obtain refined pseudo-labels $q_b^{*}=\phi\left(g_{\theta}^{*}\left(\alpha\left(u_b^m\right)\right)\right)$ for $b=1,\ldots,\mu B$
  if Base SSL=='FixMatch' **then**
    $loss_F^{*}=loss_F\left(\mathcal{MX},\mathcal{MU},q^{*},0;\theta\right)$
    $\Delta\boldsymbol{\theta}\propto\nabla_{\boldsymbol{\theta}}loss_F^{*},\quad \boldsymbol{\theta}\leftarrow\boldsymbol{\theta}+\Delta\boldsymbol{\theta}$
  end if
  if Base SSL=='ReMixMatch' **then**
    Produce class probabilities on wealy augmented labeled samples $P_{\theta}\left(y|\alpha\left(x_b^m\right)\right)$ for $b=1,\ldots,B$
    $CEloss=CrossEntropy\left(p_b^m, P_{\theta}\left(y|\alpha\left(x_b^m\right)\right)\right)$
    $loss_R^{*}=loss_R\left(\mathcal{MX},\mathcal{MU},q^{*};\theta\right)+CEloss$
    $\Delta\boldsymbol{\theta}\propto\nabla_{\boldsymbol{\theta}}loss_R^{*},\quad \boldsymbol{\theta}\leftarrow\boldsymbol{\theta}+\Delta\boldsymbol{\theta}$
  end if
end while
Produce logits for a solid color image $g_{\theta}(\mathcal{I})$
Produce logits for test samples $g_{\theta}\left(x_k^{test}\right)$ for $k=1,\ldots,K$
Obtain refined logits $g_{\theta}^{*}\left(x_k^{test}\right)=g_{\theta}\left(x_k^{test}\right)-g_{\theta}(\mathcal{I})$ for $k=1,\ldots,K$
Obtain refined class predictions $f_{\theta}^{*}\left(x_k^{test}\right)=\arg\max_c g_{\theta}^{*}\left(x_k^{test}\right)_c$ for $k=1,\ldots,K$

baseline CISSL algorithms. We used random cropping and horizontal flipping for weak data augmentation and Cutout [12] and RandomAugment [9] for strong data augmentation. These augmentation techniques are detailed in Appendix C. To use CDMAD after network parameters are stabilized, we trained naive ReMixMatch and FixMatch for first 100 epochs, and subsequently used CDMAD to refine pseudo-labels, similar to DARP [26]. We conducted experiments using the GPU server Nvidia Tesla-V100 and 3090ti and used the Python library PyTorch 1.11.0 and 1.12.1. Our experiment results can be reproduced using the code in the supplementary material.

I. Description of baseline algorithms

The classification performance of the CDMAD was compared with those of the following algorithms: **1. vanilla algorithm** - Deep CNN trained with cross-entropy loss, **2. CIL algorithms** - Re-sampling [22], LDAM-DRW [5], and cRT [25], **3. SSL algorithms** - FixMatch [45] and ReMixMatch [3], and **4. CISSL algorithms** - DARP, DARP+LA, DARP+cRT [26], CReST, CReST+LA [55], ABC [34], CoSSL [13], DASO [41], SAW, SAW+LA and SAW+cRT [31] combined with FixMatch and ReMixMatch. Adsh [16], DebiasPL [52], UDAL [32] and L2AC [50] combined with FixMatch. We report the performance of the baseline algorithms reported in Tables of Lai et al. [31] and Fan et al. [13] when it is reproducible; the performance measured using the uploaded code was reported otherwise.

J. Further qualitative analysis

J.1. Case of $\gamma_l = \gamma_u$

In Table 1 of Section 4.2, CDMAD performed better than the baseline CISSL algorithms when the class distributions of the labeled and unlabeled sets are assumed to be the same. To verify whether the pseudo-labels and class predictions on test samples refined by CDMAD contributed to its superior performance, we conducted two types of comparison: 1) pseudo-labels refined by CDMAD vs. true labels of unlabeled samples, and 2) class predictions refined by CDMAD vs. true labels of test samples. These results are also compared to those from FixMatch and ReMixMatch.

First, Fig. 4 compares the confusion matrices of pseudo-labels generated by (a) FixMatch, (b) FixMatch+CDMAD, (c) ReMixMatch, and (d) ReMixMatch+CDMAD trained on CIFAR-10-LT under $\gamma_l = 100$ and $\gamma_u = 100$. The value in the ith row and jth column represents the proportion of the ith class samples classified as the jth class. We can observe that the pseudo-labels of FixMatch and ReMixMatch are biased toward the majority classes. Specifically, the data points in the minority classes (e.g., classes 8 and 9) are often misclassified into the majority classes (e.g. classes 0 and 1). In contrast, Fig. 4 (b) and Fig. 4 (d) show that Fix-

Match+CDMAD and ReMixMatch+CDMAD made nearly balanced class predictions.

Second, Fig. 5 compares the confusion matrices of the class predictions on the test set of CIFAR-10 using (a) FixMatch, (b) FixMatch+CDMAD, (c) ReMixMatch, and (d) ReMixMatch+CDMAD trained on CIFAR-10-LT under $\gamma_l = 100$ and $\gamma_u = 100$. Similar to Fig. 4, FixMatch+CDMAD and ReMixMatch+CDMAD made more balanced predictions across classes.

J.2. Case of $\gamma_l \neq \gamma_u$

In Table 2 of Section 4.2, the proposed algorithm performed better than the baseline algorithms when the class distribution of the unlabeled set is assumed to be unknown and actually differs with that of the labeled set. To verify whether the pseudo-labels and class predictions refined by CDMAD contributed to its superior performance, we conducted three types of comparison: 1) pseudo-labels refined by CDMAD vs. true labels of unlabeled samples, 2) representations learned with unrefined pseudo-labels vs. representations learned with pseudo-labels refined by CDMAD, and 3) class predictions refined by CDMAD vs. true labels of test samples. These results are also compared to those from FixMatch and ReMixMatch.

First, Fig. 6 compares the confusion matrices of pseudo-labels generated by (a) FixMatch, (b) FixMatch+CDMAD, (c) ReMixMatch, and (d) ReMixMatch+CDMAD trained on CIFAR-10-LT under $\gamma_l = 100$ and $\gamma_u = 1$. The value in the ith row and jth column represents the proportion of the ith class samples classified as the jth class. We can observe that the pseudo-labels of FixMatch and ReMixMatch are biased toward the majority classes. Specifically, the data points in the minority classes (e.g., classes 7, 8 and 9) are often misclassified into the majority classes (e.g. classes 0 and 1). In contrast, Fig. 6 (b) and Fig. 6 (d) show that FixMatch+CDMAD and ReMixMatch+CDMAD made nearly balanced class predictions.

Second, Fig. 7 compares t-distributed stochastic neighbor embedding (t-SNE) [47] of representations obtained for the test set of CIFAR-10 using FixMatch, FixMatch+CDMAD, ReMixMatch, and ReMixMatch+CDMAD trained on CIFAR-10 with $\gamma_l = 100$ and $\gamma_u = 1$ (unknown γ_u), where different colors indicate different classes in CIFAR-10. We can observe that the representations obtained using FixMatch+CDMAD and ReMixMatch+CDMAD are separated into classes with clearer boundaries compared the those from FixMatch and ReMixMatchFrom in Fig. 7 (a) and Fig. 7 (c). This is probably because CDMAD appropriately refined the biased pseudo-labels and used them for training, whereas FixMatch and ReMixMatch failed to learn the representations properly because they used the biased pseudo-labels for training. These results demonstrate that the quality of representations can be improved by using well refined pseudo-labels (Fig. 6 (b) and Fig. 6 (d)) for training.

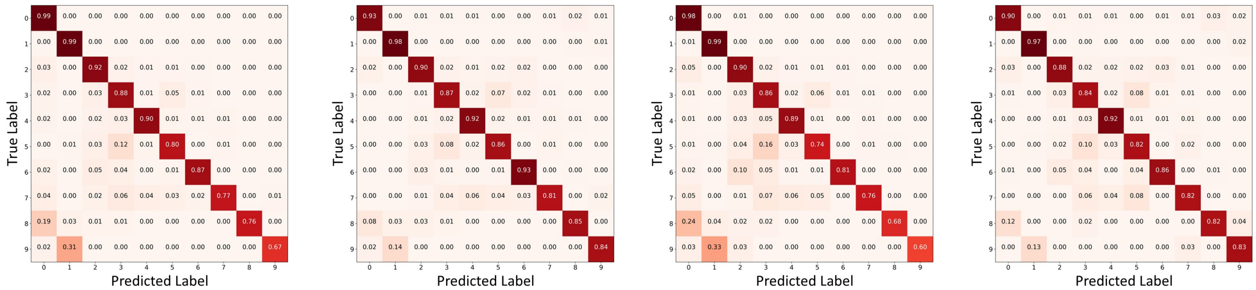
Third, Fig. 8 compares the confusion matrices of the class predictions on the test set of CIFAR-10 using (a) FixMatch and (b) FixMatch+CDMAD trained on CIFAR-10-LT under $\gamma_l = 100$ and $\gamma_u = 1$. Similar to Fig. 6, FixMatch+CDMAD made more balanced predictions across classes compared to the other algorithms. (Note that the results using ReMixMatch and ReMixMatch+CDMAD are presented in Section 4.3.)

K. Further comparison with LA

Because CDMAD can be viewed as an extension of LA for incorporating awareness of class distribution mismatch, we compared the classification performance of LA and CDMAD for CISSL under the settings that the class distributions of the labeled and unlabeled sets mismatch. To use LA for CISSL, we refined pseudo-labels and class predictions on test samples by LA similar to CDMAD. Experimental results are presented in Tab. 1. ReMixMatch+LA adjusts the logits on inputs by the log of the class distribution of the labeled set by assuming that the class distribution of the unlabeled set is the same as that of the labeled set. ReMixMatch+LA* adjusts the logits on inputs by the log of the class distribution of the whole training set by assuming that the class distribution of the unlabeled set is known, although it differs from that of the labeled set. From Tab. 1, we can observe that ReMixMatch+CDMAD performed significantly better than both ReMixMatch+LA and ReMixMatch+LA*. This may be because CDMAD refined the biased pseudo-labels and class predictions on test samples more effectively than ReMixMatch+LA and ReMixMatch+LA* by incorporating awareness of class distribution mismatch. It should be noted that LA* cannot re-balance the classifier to an appropriate degree even if the class distribution of the unlabeled set is known under the class distribution mismatch setting. This may be because in SSL, each labeled data point is typically used more frequently and importantly than each unlabeled data point. Consequently, the classifier may become biased towards the class distribution of the labeled set to a greater degree than the class distribution of the entire training set, while still being affected by the class distribution of the unlabeled set.

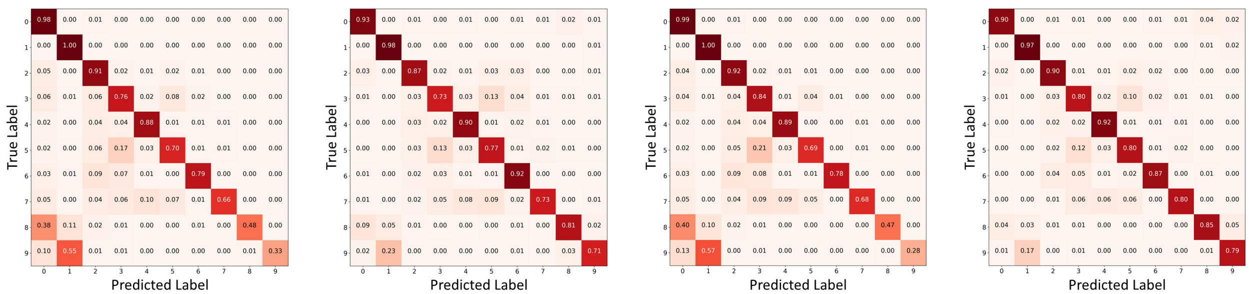
Table 1. bACC/GM on CIFAR-10-LT under $\gamma_l \neq \gamma_u$.

	CIFAR-10-LT ($\gamma_l = 100$)		
Algorithm	$\gamma_u = 1$	$\gamma_u = 50$	$\gamma_u = 150$
ReMixMatch+LA	76.6/ 66.8	69.9/ 52.6	70.5/ 42.7
ReMixMatch+LA*	69.2/ 54.0	73.7/ 70.8	58.3/ 27.4
ReMixMatch+CDMAD	**89.9/ 89.6**	**86.9/ 86.7**	**83.1/ 82.7**



(a) FixMatch (b) FixMatch+CDMAD (c) ReMixMatch (d) ReMixMatch+CDMAD

Figure 4. Confusion matrices of pseudo-labels generated by (a) FixMatch, (b) FixMatch+CDMAD, (c) ReMixMatch, and (d) ReMixMatch+CDMAD trained on CIFAR-10-LT under $\gamma_l = 100$ and $\gamma_u = 100$.



(a) FixMatch (b) FixMatch+CDMAD (c) ReMixMatch (d) ReMixMatch+CDMAD

Figure 5. Confusion matrices of the class predictions on the test set of CIFAR-10 using (a) FixMatch, (b) FixMatch+CDMAD, (c) ReMixMatch, and (d) ReMixMatch+CDMAD trained on CIFAR-10-LT under $\gamma_l = 100$ and $\gamma_u = 100$.

L. Fine grained experimental results

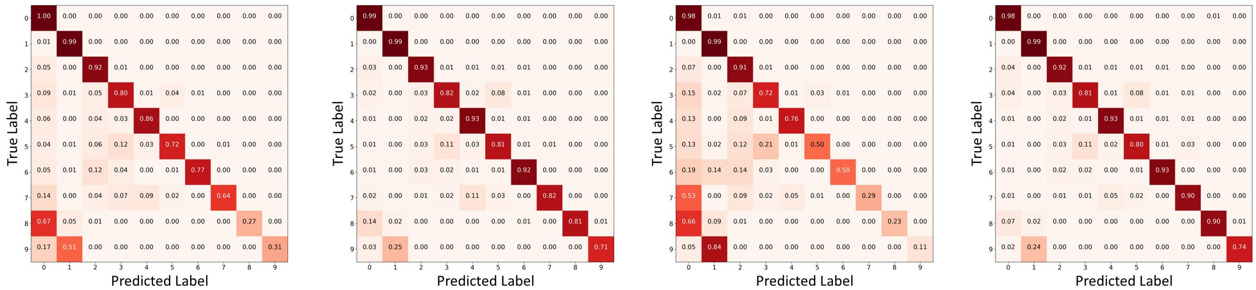
To verify that CDMAD improves classification performance for minority classes, we performed experiments using FixMatch/ ReMixMatch and FixMatch/ReMixMatch+CDMAD on CIFAR-10-LT and measured the accuracy for Many/Medium/Few groups separately (for CIFAR-10-LT, we set the first three classes as many shot groups, then next four classes as medium shot groups, and then last three classes as few shot groups). We also measured the fine grained classification performance of FixMatch/ReMixMatch+CoSSL [13] on CIFAR-10-LT and compared them with those of CDMAD for comparison with a recent CISSL algorithm. The results are summarized in Tab. 2, Tab. 3, and Tab. 4. We can observe that FixMatch+CDMAD and ReMixMatch+CDMAD greatly improved accuracy for few shot groups with only slightly decreased accuracy for many shot groups compared to FixMatch and ReMixMatch. We can also observe that FixMatch/ ReMixMatch+CDMAD achieved better medium and few shot classification accuracies than FixMatch/ ReMixMatch+COSSL. These results demonstrate that CDMAD effectively relieves class imbalance.

Table 2. Fine grained experimental results under $\gamma_l = \gamma_u = 100$.

CIFAR-10-LT ($\gamma_l = \gamma_u = 100$)				
Algorithm	Overall	Many	Medium	Few
FixMatch	72.5	95.0	74.6	47.3
FixMatch+CDMAD	83.6	91.9	82.2	77.2
ReMixMatch	74.3	96.7	77.8	47.2
ReMixMatch+CDMAD	85.5	90.1	84.8	81.8

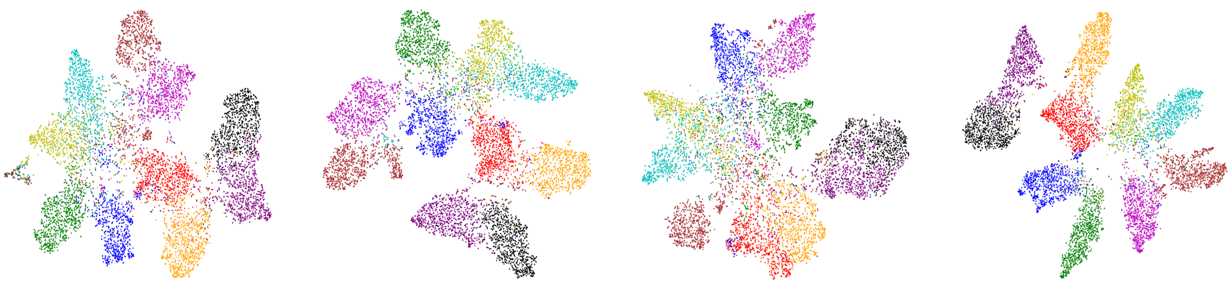
Table 3. Fine grained experimental results under $\gamma_l = 100$, and $\gamma_u = 1$.

CIFAR-10-LT ($\gamma_l = 100$, $\gamma_u = 1$)				
Algorithm	Overall	Many	Medium	Few
FixMatch	70.2	96.3	77.7	34.0
FixMatch+CDMAD	87.5	95.6	86.4	80.9
ReMixMatch	65.4	96.6	70.8	27.0
ReMixMatch+CDMAD	89.9	96.5	87.8	86.0



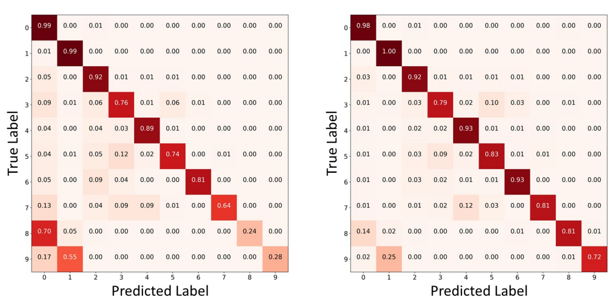
(a) FixMatch (b) FixMatch+CDMAD (c) ReMixMatch (d) ReMixMatch+CDMAD

Figure 6. Confusion matrices of pseudo-labels generated by (a) FixMatch, (b) FixMatch+CDMAD, (c) ReMixMatch, and (d) ReMixMatch+CDMAD trained on CIFAR-10-LT under $\gamma_l = 100$ and $\gamma_u = 1$.



(a) FixMatch (b) FixMatch+CDMAD (c) ReMixMatch (d) ReMixMatch+CDMAD

Figure 7. t-SNE of representations obtained for the test set of CIFAR-10 using (a) FixMatch, (b) FixMatch+CDMAD, (c) ReMixMatch, and (d) ReMixMatch+CDMAD trained on CIFAR-10-LT under $\gamma_l = 100$ and $\gamma_u = 1$.



(a) FixMatch (b) FixMatch+CDMAD

Figure 8. Confusion matrices of the class predictions on the test set of CIFAR-10 using (a) FixMatch and (b) FixMatch+CDMAD trained on CIFAR-10-LT under $\gamma_l = 100$ and $\gamma_u = 1$.

M. Comparing CDMAD with DASO

Because classification performance of DASO were measured under slightly different settings from ours, it was difficult to fairly compare their classification performance with that of CDMAD in the main paper. Nevertheless, in the case of DASO, we conducted experiments in the same setting as ours using the official code in github. The classification performance of DASO is summarized in Tab. 5, Tab. 6, and Tab. 7. From Tab. 5, Tab. 6, and Tab. 7, we can ob-

Table 4. Fine grained experimental results under $\gamma_l = \gamma_u = 100$.

CIFAR-10-LT ($\gamma_l = \gamma_u = 100$)				
Algorithm	Overall	Many	Medium	Few
FixMatch+CoSSL	83.2	93.4	81.1	75.8
FixMatch+CDMAD	83.6	91.9	82.2	77.2
ReMixMatch+CoSSL	84.1	91.7	82.1	79.1
ReMixMatch+CDMAD	85.5	90.1	84.8	81.8

Table 5. bACC/GM on CIFAR-10-LT under $\gamma = \gamma_l = \gamma_u$.

CIFAR-10-LT ($\gamma = \gamma_l = \gamma_u$)			
Algorithm	$\gamma = 50$	$\gamma = 100$	$\gamma = 150$
FixMatch+DASO	81.8/ 81.0	75.7/ 74.0	72.0/ 68.9
FixMatch+DASO+LA	84.1/ 83.7	79.4/ 78.8	76.5/ 75.5
FixMatch+CDMAD	**87.3/87.0**	**83.6/83.1**	**80.8/79.9**
ReMixMatch+DASO	82.5/ 81.9	76.0/ 73.9	70.8/ 66.5
ReMixMatch+DASO+LA	85.9/ 85.7	82.8/ 82.4	79.0/ 78.4
ReMixMatch+CDMAD	**88.3/ 88.1**	85.5/ **85.3**	**82.5/ 82.0**

Table 6. Comparison of bACC/GM on CIFAR-10-LT and STL-10-LT under $\gamma_l \neq \gamma_u$.

	CIFAR-10-LT ($\gamma_l = 100$)			STL-10-LT (γ_u =Unknown)	
Algorithm	$\gamma_u = 1$	$\gamma_u = 50$	$\gamma_u = 150$	$\gamma_l = 10$	$\gamma_l = 20$
FixMatch+DASO	86.4/ 86.0	79.1/ 78.2	74.2/ 71.6	68.4/ 65.3	62.1/ 58.9
FixMatch+DASO+LA	86.2/ 85.8	81.7/ 81.2	78.0/ 77.0	68.9/ 66.3	66.0/ 64.6
FixMatch+CDMAD	**87.5/ 87.1**	**85.7/ 85.3**	**82.3/ 81.8**	**79.9/ 78.9**	**75.2/ 73.5**
ReMixMatch+DASO	89.6/ 89.3	79.6/ 77.8	72.3/ 69.0	75.1/ 73.6	66.8/ 61.8
ReMixMatch+DASO+LA	80.6/ 77.7	84.8/ 84.5	79.7/ 79.2	78.1/ 77.3	75.3/ 74.0
ReMixMatch+CDMAD	**89.9/ 89.6**	**86.9/ 86.7**	**83.1/ 82.7**	**83.0/82.1**	**81.9/80.9**

Table 7. Comparison of bACC on CIFAR-100-LT.

	CIFAR-100-LT ($\gamma = \gamma_l = \gamma_u$)		
Algorithm	$\gamma = 20$	$\gamma = 50$	$\gamma = 100$
FixMatch+DASO	45.8	39.2	33.9
FixMatch+DASO+LA	46.2	39.9	34.5
FixMatch+CDMAD	**54.3**	**48.8**	**44.1**
ReMixMatch+DASO	51.5	43.0	38.2
ReMixMatch+DASO+LA	52.8	45.5	40.3
ReMixMatch+CDMAD	**57.0**	**51.1**	**44.9**

serve that the proposed algorithm outperforms DASO. From Tab. 6, we can also observe that combining DASO with LA degrades the classification performance when the class distributions of the labeled and unlabeled sets severely differ. This may be because the LA considers only the class distribution of the labeled set when the class distribution of the unlabeled set is unknown. These results show the importance of re-balancing the classifier by considering the class distribution of the unlabeled set. These results demonstrate the effectiveness of CDMAD.

References

[1] Ricardo Barandela, E Rangel, José Salvador Sánchez, and Francesc J Ferri. Restricted decontamination for the imbalanced training sample problem. In *Iberoamerican congress on pattern recognition*, pages 424–431. Springer, 2003. 1

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

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

[4] Jiarui Cai, Yizhou Wang, and Jenq-Neng Hwang. Ace: Ally complementary experts for solving long-tailed recognition in one-shot. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 112–121, 2021. 1

[5] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2019. 1, 4

[6] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002. 1

[7] H Chen, R Tao, Yue Fan, Y Wang, M Savvides, J Wang, B Raj, X Xie, and Bernt Schiele. Softmatch: Addressing the quantity-quality tradeoff in semi-supervised learning. In *Eleventh International Conference on Learning Representations*. OpenReview. net, 2023. 1

[8] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011. 3

[9] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 702–703, 2020. 1, 4

[10] Jiequan Cui, Zhisheng Zhong, Shu Liu, Bei Yu, and Jiaya Jia. Parametric contrastive learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 715–724, 2021. 1

[11] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9268–9277, 2019. 1

[12] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. 1, 4

[13] Yue Fan, Dengxin Dai, Anna Kukleva, and Bernt Schiele. Cossl: Co-learning of representation and classifier for imbalanced semi-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14574–14584, 2022. 3, 4, 6

[14] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations*, 2018. 1

[15] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *Advances in Neural Information Processing Systems*. MIT Press, 2005. 1

[16] Lan-Zhe Guo and Yu-Feng Li. Class-imbalanced semi-supervised learning with adaptive thresholding. In *International Conference on Machine Learning*, pages 8082–8094. PMLR, 2022. 1, 4

[17] Haibo He and Edwardo A Garcia. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9):1263–1284, 2009. 1

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3

[19] Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. Learning deep representation for imbalanced classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5375–5384, 2016. 1, 3

[20] Minyoung Huh, Pulkit Agrawal, and Alexei A Efros. What makes imagenet good for transfer learning? *arXiv preprint arXiv:1608.08614*, 2016. 3

[21] Muhammad Abdullah Jamal, Matthew Brown, Ming-Hsuan Yang, Liqiang Wang, and Boqing Gong. Rethinking class-balanced methods for long-tailed visual recognition from a domain adaptation perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7610–7619, 2020. 1

[22] N JAPKOWICZ. The class imbalance problem: Significance and strategies. In *Proc. 2000 International Conference on Artificial Intelligence*, pages 111–117, 2000. 1, 4

[23] Ziyu Jiang, Tianlong Chen, Bobak J Mortazavi, and Zhangyang Wang. Self-damaging contrastive learning. In *International Conference on Machine Learning*, pages 4927–4939. PMLR, 2021. 1

[24] Bingyi Kang, Yu Li, Sa Xie, Zehuan Yuan, and Jiashi Feng. Exploring balanced feature spaces for representation learning. In *International Conference on Learning Representations*, 2020. 1

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

[26] Jaehyung Kim, Youngbum Hur, Sejun Park, Eunho Yang, Sung Ju Hwang, and Jinwoo Shin. Distribution aligning refinery of pseudo-label for imbalanced semi-supervised learning. In *Advances in Neural Information Processing Systems*, pages 14567–14579. Curran Associates, Inc., 2020. 4

[27] Jaehyung Kim, Jongheon Jeong, and Jinwoo Shin. M2m: Imbalanced classification via major-to-minor translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13896–13905, 2020. 1

[28] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR, 2015*, 2015. 3

[29] A Krizhevsky. Learning multiple layers of features from tiny images. *Technical report, Department of Computer Science, University of Toronto*, 2009. 3

[30] Miroslav Kubat, Stan Matwin, et al. Addressing the curse of imbalanced training sets: one-sided selection. In *Icml*, page 179. Citeseer, 1997. 3

[31] Zhengfeng Lai, Chao Wang, Henrry Gunawan, Sen-Ching S Cheung, and Chen-Nee Chuah. Smoothed adaptive weighting for imbalanced semi-supervised learning: Improve reliability against unknown distribution data. In *International Conference on Machine Learning*, pages 11828–11843. PMLR, 2022. 4

[32] Justin Lazarow, Kihyuk Sohn, Chen-Yu Lee, Chun-Liang Li, Zizhao Zhang, and Tomas Pfister. Unifying distribution alignment as a loss for imbalanced semi-supervised learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5644–5653, 2023. 4

[33] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, 2013. 1

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

[35] Bolian Li, Zongbo Han, Haining Li, Huazhu Fu, and Changqing Zhang. Trustworthy long-tailed classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6970–6979, 2022. 1

[36] Junnan Li, Caiming Xiong, and Steven CH Hoi. Comatch: Semi-supervised learning with contrastive graph regularization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9475–9484, 2021. 1

[37] Tianhong Li, Peng Cao, Yuan Yuan, Lijie Fan, Yuzhe Yang, Rogerio S Feris, Piotr Indyk, and Dina Katabi. Targeted supervised contrastive learning for long-tailed recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6918–6928, 2022. 1

[38] Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. In *International Conference on Learning Representations*, 2020. 1

[39] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2013. 1

[40] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993, 2018. 1

[41] Youngtaek Oh, Dong-Jin Kim, and In So Kweon. Daso: Distribution-aware semantics-oriented pseudo-label for imbalanced semi-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9786–9796, 2022. 4

[42] Sungrae Park, JunKeon Park, Su-Jin Shin, and Il-Chul Moon. Adversarial dropout for supervised and semi-supervised learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018. 1

[43] Jiawei Ren, Cunjun Yu, shunan sheng, Xiao Ma, Haiyu Zhao, Shuai Yi, and hongsheng Li. Balanced meta-softmax for long-tailed visual recognition. In *Advances in Neural Information Processing Systems*, pages 4175–4186. Curran Associates, Inc., 2020. 1

[44] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 3

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

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

[47] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9 (11), 2008. 5

[48] Vikas Verma, Alex Lamb, Juho Kannala, Yoshua Bengio, and David Lopez-Paz. Interpolation consistency training for semi-supervised learning. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 3635–3641. International Joint Conferences on Artificial Intelligence Organization, 2019. 1

[49] Peng Wang, Kai Han, Xiu-Shen Wei, Lei Zhang, and Lei Wang. Contrastive learning based hybrid networks for long-tailed image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 943–952, 2021. 1

[50] Renzhen Wang, Xixi Jia, Quanziang Wang, Yichen Wu, and Deyu Meng. Imbalanced semi-supervised learning with bias adaptive classifier. In *The Eleventh International Conference on Learning Representations*, 2023. 4

[51] Xudong Wang, Long Lian, Zhongqi Miao, Ziwei Liu, and Stella Yu. Long-tailed recognition by routing diverse distribution-aware experts. In *International Conference on Learning Representations*, 2021. 1

[52] Xudong Wang, Zhirong Wu, Long Lian, and Stella X Yu. Debiased learning from naturally imbalanced pseudo-labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14647–14657, 2022. 4

[53] Yidong Wang, Hao Chen, Qiang Heng, Wenxin Hou, Yue Fan, Zhen Wu, Jindong Wang, Marios Savvides, Takahiro Shinozaki, Bhiksha Raj, Bernt Schiele, and Xing Xie. Freematch: Self-adaptive thresholding for semi-supervised learning. In *The Eleventh International Conference on Learning Representations*, 2023. 1

[54] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Learning to model the tail. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017. 1

[55] Chen Wei, Kihyuk Sohn, Clayton Mellina, Alan Yuille, and Fan Yang. Crest: A class-rebalancing self-training framework for imbalanced semi-supervised learning. *arXiv preprint arXiv:2102.09559*, 2021. 3, 4

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

[57] Xi Yin, Xiang Yu, Kihyuk Sohn, Xiaoming Liu, and Manmohan Chandraker. Feature transfer learning for deep face recognition with under-represented data. *arXiv e-prints*, pages arXiv–1803, 2018. 1

[58] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016. 3

[59] Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. *Advances in Neural Information Processing Systems*, 34:18408–18419, 2021. 1

[60] Yifan Zhang, Bryan Hooi, Lanqing Hong, and Jiashi Feng. Self-supervised aggregation of diverse experts for test-agnostic long-tailed recognition. *Advances in Neural Information Processing Systems*, 35:34077–34090, 2022. 1

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