Rebalancing Using Estimated Class Distribution for Imbalanced Semi-Supervised Learning under Class Distribution Mismatch

Taemin Park*, Hyuck Lee*, and Heeyoung Kim†

Department of Industrial and Systems Engineering, KAIST
Daejeon 34141, Republic of Korea
{ptm1001,dlgur0921,heeyoungkim}@kaist.ac.kr

Abstract. Despite significant advancements in class-imbalanced semi-supervised learning (CISSL), many existing algorithms explicitly or implicitly assume that the class distribution of unlabeled data matches that of labeled data. However, when this assumption fails in practice, the classification performance of such algorithms may degrade due to incorrectly assigned weight to each class during training. We propose a novel CISSL algorithm called *Rebalancing Using Estimated Class Distribution (RECD)*. RECD estimates the unknown class distribution of unlabeled data through Monte Carlo approximation, leveraging predicted class probabilities for unlabeled samples, and subsequently rebalances the classifier based on the estimated class distribution. Additionally, we propose an extension of feature clusters compression in the context of CISSL to mitigate feature map imbalance by densifying minority class clusters. Experimental results on four benchmark datasets demonstrate that RECD achieves state-of-the-art classification performance in CISSL.

Keywords: Class-imbalanced semi-supervised learning · Long-tailed learning · Auxiliary Balanced Classifier

1 Introduction

In real-world datasets, the number of samples often varies across different classes. Deep neural network algorithms trained on such datasets tend to exhibit bias towards majority classes, which have a larger number of samples in the dataset [15, 31, 41]. To mitigate the bias, various class-imbalanced learning (CIL) techniques have been proposed, including re-sampling [4, 10, 30, 42], re-weighting [8, 29, 36], decoupled learning [19, 35, 53], and multi-expert learning [43, 48, 52].

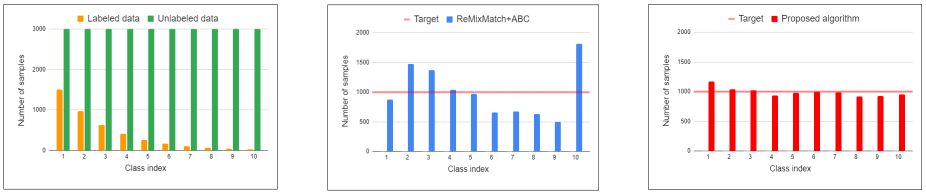
Although the CIL techniques significantly mitigate the bias, they typically assume that the entire training set is labeled. However, in reality, labeling samples requires substantial resources, resulting in many real-world datasets containing

* Equal contribution
† Corresponding author

a significant proportion of unlabeled data. Because the CIL techniques assign importance to each data point based on its corresponding label, they cannot be directly applied when the training set includes unlabeled samples.

Recently, several class-imbalanced semi-supervised learning (CISSL) algorithms, including ABC [27], UDAL [24], CReST [46], DARP [20], DASO [34], CoSSL [11], and SAW [23], have emerged to address the bias towards majority classes in scenarios where the training set consists of both labeled and unlabeled data. However, ABC, UDAL, and CReST explicitly assume that the class distribution of the unlabeled data matches that of the labeled data. Similarly, DARP, DASO, and SAW can be seen as implicitly assuming the class distribution match between the labeled and unlabeled data, as they incorporate CIL techniques such as LA [32] and cRT [19], which rely on the same class distribution between the labeled and unlabeled data. Additionally, CoSSL solely considers the class distribution of labeled data in its main component, TFE.

However, the class distribution of the unlabeled set is often unknown and may differ from that of the labeled set [34, 40, 47]. In this case, the classification performance of the aforementioned algorithms may degrade because the unknown class distribution of the unlabeled set is not considered in the rebalancing process. Fig. 1b illustrates the class predictions on the test set of CIFAR-10 [21] using ReMixMatch+ABC, an existing CISSL algorithm, trained on the long-tailed version of the CIFAR-10 [8] training set where the class distributions of labeled and unlabeled sets mismatch, as presented in Fig. 1a. From Fig. 1b, we can observe that many test samples are mispredicted as Class 10 (the most minor class). This may be because ABC rebalances the classifier excessively, considering only the class distribution of the labeled set.



(a) Class distributions **(b)** ReMixMatch+ABC **(c)** Proposed algorithm

Fig. 1: Comparison of the predictions of ReMixMatch+ABC and the proposed algorithm (ReMixMatch+RECD) trained on an imbalanced version of CIFAR-10 where the class distributions of labeled and unlabeled sets mismatch, as illustrated in Fig. 1a.

To rebalance the classifier appropriately when the class distribution of the unlabeled set mismatches that of the labeled set, we propose a novel CISSL algorithm called *Rebalancing Using Estimated Class Distribution (RECD)*, which estimates the unknown class distribution using Monte Carlo (MC) approximation and rebalances the training loss based on the estimated class distribution. Specifically, RECD is a modular algorithm that operates in conjunction with previous CISSL algorithms. While RECD can generally be integrated with most

CISSL algorithms by providing additional information about the class distribution of the unlabeled set, in this paper, we specifically combine RECD with ABC due to its simple and effective framework compared to other CISSL algorithms.

To estimate the class distribution of unlabeled data, RECD first computes the average class predictions of ABC for the entire unlabeled set. Because generating class predictions for the entire unlabeled set can be computationally intensive, RECD generates class predictions on an unlabeled minibatch for each iteration and updates the exponential moving average (EMA) of the class predictions on the unlabeled set. Utilizing MC approximation, this average class prediction serves as the *classifier's prior*, representing the estimated unknown class distribution of the unlabeled set. Then, RECD rebalances the training loss on unlabeled samples for ABC based on the estimated class distribution of the unlabeled set, and trains ABC using this rebalanced unlabeled loss in addition to the training loss on labeled samples for ABC. As depicted in Fig. 1c, ReMixMatch+RECD generates nearly balanced class predictions on the test set.

In addition to mitigating imbalance in the classifier, RECD also mitigates imbalance in the feature map. While existing CISSL algorithms primarily focus on mitigating classifier bias, there also exists imbalance in the feature map [5,18]. Specifically, feature clusters of minority classes tend to be sparser compared to the dense clusters of majority classes. These sparse features often overlap with clusters of other classes in the feature map, posing challenges in classifying minority class features [28]. To address this issue by densifying the sparse feature clusters of minority classes, we extend feature clusters compression (FCC) [28], a previous CIL technique, into the context of CISSL by leveraging the unknown class distribution of the unlabeled data estimated by RECD. We refer to this extended version of FCC as the adaptively adjusted feature multiplier (AAFM).

Our experimental results on CIFAR-10-LT [8], CIFAR-100-LT [8], STL-10-LT [20], and Small-ImageNet-127 [11] demonstrate the superior performance of the proposed algorithm compared to the baseline algorithms, regardless of whether the class distributions of labeled and unlabeled sets match or mismatch. Through detailed analyses, we verify that RECD effectively estimates the true class distribution of the unlabeled set and uses it to mitigates class imbalance. Additionally, we validate that AAFM not only reduces the test loss for minority classes but also stabilizes the training process. Finally, by conducting experiments that combine RECD with DARP [20], we validate the effectiveness of RECD beyond its application soley with ABC. The source codes for RECD are available at `https://github.com/taemin-park/RECD`.

2 Related work

Many CISSL algorithms have been proposed to address class imbalance in scenarios where the training set consists of both labeled and unlabeled data. For example, ABC [27] trains an auxiliary balanced classifier with a training loss that is rebalanced based on the class distribution of the labeled data. CReST [46] systematically enlarges the labeled dataset by assigning the labels to unlabeled

samples predicted as minority classes more frequently than those predicted as majority classes. UDAL [24] combines progressive distribution alignment and logit-adjustment [32] to adjust the training loss. ABC, CReST, and UDAL have effectively mitigated bias towards majority classes. However, they assumed that the class distribution of the unlabeled data is the same as that of the labeled data, which may pose practical limitation as described in Sec. 1.

DARP [20] solves a convex optimization problem to refine biased pseudo labels. DASO [34] blends class predictions of a similarity-based classifier and a linear classifier to obtain unbiased pseudo-labels. SAW [23] estimates the learning difficulty of each class and smoothly reweights the training loss based on the estimated learning difficulty. However, DARP, DASO, and SAW implicitly assume the same class distributions of the unlabeled and labeled datasets, as they incorporate CIL techniques such as LA [32] and cRT [19], which rebalance the classifier based on the class distribution of the labeled data.

CoSSL [11] integrates Tail-class Feature Enhancement (TFE) and class-balanced sampling to rebalance the classifier. InPL [50] introduces energy-based pseudo-labeling to effectively discern correctly predicted unlabeled samples of minority classes. DePL [44] mitigates pseudo-label bias using counterfactual reasoning and adaptive marginal loss. L2AC [40] mitigates training bias by using a bias adaptive classifier, which consists of a bias attractor and a linear classifier. Adsh [14] sets an adaptive confidence threshold for each class to utilize unlabeled samples predicted as minority classes more frequently than those predicted as majority classes. ACR [47] estimates the class distribution of the unlabeled set by comparing class predictions with class distribution prototypes, and then adaptively adjusts the training loss based on the estimated class distribution. CDMAD [26] refines the predictions for unlabeled and test samples based on the logits for solid color image. Although these algorithms have significantly improved CISSL, this area remains relatively less explored compared to SSL and CIL.

3 Preliminaries

3.1 Problem settings

We have a training dataset that consists of the labeled set $\mathcal{X} = \{(x_n, y_n) : n \in (1, \ldots, N)\}$ and unlabeled set $\mathcal{U} = \{(u_m) : m \in (1, \ldots, M)\}$, where $x_n \in R^d$ and $y_n \in R^K$ are the nth labeled sample and its corresponding label, respectively, and $u_m \in R^d$ is the mth unlabeled sample. For each class k, N_k and M_k denote the numbers of the labeled and unlabeled samples, respectively, i.e., $\sum N_k = N$ and $\sum M_k = M$. In CISSL, it is typically assumed that $N_1 \geq N_2 \geq \ldots \geq N_K$ and $M_1 \geq M_2 \geq \ldots \geq M_K$. We denote the class imbalance ratios of the labeled and unlabeled sets as $\gamma_l = \frac{N_1}{N_K}$ and $\gamma_u = \frac{M_1}{M_K}$, respectively. When M_k is unknown, γ_u would also be unknown, and it may differ from γ_l. For each training iteration, we generate minibatches $\mathcal{MB}_\mathcal{X} = \{(x_b^m, y_b^m) : b \in (1, \ldots, B_l)\} \subset \mathcal{X}$ and $\mathcal{MB}_\mathcal{U} = \{(u_b^m) : b \in (1, \ldots, B_u)\} \subset \mathcal{U}$ from $\mathcal{X}$ and $\mathcal{U}$, respectively, where B_l and B_u are the sizes of the labeled and unlabeled minibatches, respectively. Using $\mathcal{MB}_\mathcal{X}$ and $\mathcal{MB}_\mathcal{U}$, we aim to train an algorithm $f_\theta : R^d \rightarrow R^K$ that accurately predicts the

class probabilities for unseen samples, where θ denotes the network parameters. We denote the extracted features before the classification layer as $\xi(\cdot)$.

3.2 Backbone SSL algorithm

Following previous CISSL algorithms [11, 20, 23, 27, 34, 50], either FixMatch [38] or ReMixMatch [1] is used as a backbone SSL algorithm for RECD. To utilize unlabeled samples, FixMatch and ReMixMatch conduct consistency regularization using weak data augmentation techniques $\alpha(\cdot)$ (flipping and cropping) and strong data augmentation techniques $\mathcal{A}(\cdot)$ (Cutout [9] and RandomAugment [6]).

Specifically, FixMatch first predicts the class probabilities for $\alpha(u_b^m)$ as $f(\alpha(u_b^m))$ and then generates the pseudo-label as $q_b^m = \arg\max_k(f(\alpha(u_b^m))_k)$ when the confidence exceeds the confidence threshold ϵ, i.e., $\max_k(f(\alpha(u_b^m))_k) \geq \epsilon$, where $(\cdot)_k$ denotes the kth element of $(\cdot)$. Then, FixMatch predicts the class probabilities for $\mathcal{A}(u_b^m)$ as $f(\mathcal{A}(u_b^m))$ and ensures $f(\mathcal{A}(u_b^m))$ to be close to the pseudo-label q_b^m by minimizing a loss that penalizes the distance between $f(\mathcal{A}(u_b^m))$ and q_b^m. In addition, FixMatch minimizes the prediction loss calculated between weakly augmented labeled sample $\alpha(x_b^m)$ and p_b^m, which is the one-hot version of y_b^m.

Similarly, ReMixMatch first predicts the class probabilities of $\alpha(u_b^m)$ as $q_b^m = f(\alpha(u_b^m))$. Then ReMixMatch applies a distribution alignment technique as $\tilde{q}_b^m = \text{Normalize}(q_b^m \times q(y)/\tilde{q}(y))$, where $\text{Normalize}(x)_i = x_i / \sum_j x_j$, $\tilde{q}(y)$ is the moving average of class predictions on unlabeled samples, and $q(y)$ indicates the class distribution of the training set. Finally, ReMixMatch sharpens $\tilde{q}_b^m$ as $\bar{q}_b^m = \text{Normalize}((\tilde{q}_b^m)^{1/T})$, where T is the temperature for sharpening. Using $\bar{q}_b^m$ as pseudo-labels, ReMixMatch conducts consistency regularization [33, 37], entropy minimization [13, 25] and mixup regularization [2, 39, 51]. ReMixMatch also minimizes the rotation loss [12] for self-supervised learning. The training loss of the backbone SSL algorithm, L_{back}, is detailed in Appendix B.

3.3 Auxiliary balanced classifier (ABC)

To use a balanced classifier trained on a class-balanced subset of the training set, while benefiting from the high-quality representations learned from the entire training set, Lee and Kim [27] introduced an auxiliary balanced classifier (ABC) attached to the representation layer of a backbone SSL algorithm. To ensure that ABC is trained on a class-balanced subset of the training set, ABC generates a $0/1$ mask $M(\cdot)$ for each training sample, where the $0/1$ mask determines whether each sample contributes to the training loss. Specifically, ABC samples a $0/1$ mask from a Bernoulli distribution for each labeled and unlabeled sample as

$$M(x_b^m) = \mathcal{B}(\frac{N_K}{N_{y_b^m}}) \quad \text{and} \quad M(u_b^m) = \mathcal{B}(\frac{N_K}{N_{\hat{q}_b^m}}). \tag{1}$$

$\mathcal{B}(\cdot)$ denotes a Bernoulli distribution and $\hat{q}_b^m = \arg\max(q_b^m)$, where q_b^m is the class prediction of the ABC for an unlabeled sample u_b^m. Note that in Eq. (1), the probability of sampling 1 is adjusted to be inversely proportional to the number

of training samples belonging to each class. This sampling strategy ensures that ABC is trained with an equal frequency from each class's data.

With the 0/1 mask, ABC calculates the classification loss, L_{cls}, between weakly augmented labeled sample $\alpha(x_b^m)$ and its corresponding one-hot version label p_b^m. In addition, ABC also calculates a consistency regularization loss, L_{con}, between weakly augmented unlabeled sample $\alpha(u_b^m)$ and strongly augmented unlabeled sample $\mathcal{A}(u_b^m)$, which is minimized to ensure that decision boundaries lie in low-density regions. The training losses L_{cls} and L_{con} are calculated as:

$$L_{cls} = \frac{1}{B_l} \sum_{b=1}^{B_l} M(x_b^m) \mathcal{H}(p_b^m, f_{abc}(\xi\left(\alpha(x_b^m)\right))), \tag{2}$$

$$L_{con} = \frac{1}{B_u} \sum_{b=1}^{B_u} \sum_{i=1}^{2} M(u_b^m) \mathbb{1}(\max(q_b^m) \geq \epsilon) \mathcal{H}(q_b^m, f_{abc}(\xi\left(\mathcal{A}_i(u_b^m)\right))), \tag{3}$$

where $\mathcal{H}$ is the cross-entropy loss, $q_b^m = f_{abc}(\xi(\alpha(u_b^m)))$, f_{abc} denotes the auxiliary classifier attached to the representation layer, and $\mathbb{1}(\cdot)$ denotes an indicator function. With the L_{cls}, L_{con} and the training loss of backbone SSL algorithm L_{back}, total training loss of ABC is calculated as:

$$L_{ABC} = L_{cls} + L_{con} + L_{back}. \tag{4}$$

ABC [27] effectively mitigated class imbalance when the class distributions of the labeled and unlabeled sets match. However, as we can observe in the second equation of the Eq. (1), because the 0/1 masks for unlabeled samples are generated based on the class distribution of the labeled set without considering that of the unlabeled set, f_{abc} may not be properly rebalanced when the unknown class distribution of the unlabeled set significantly mismatches that of the labeled set.

3.4 Feature clusters compression (FCC) [28]

To separate features extracted from samples of minority classes from those of majority classes by densifying the feature clusters, Li *et al.* [28] introduced FCC. For a training sample x of class k, FCC multiplies the extracted feature $\xi(x)$ by a scaling factor τ_k, and uses $\tau_k \times \xi(x)$ as input for the classification layer ψ, as

$$f_\theta(x) = \psi(\tau_k \times \xi(x)), \tag{5}$$

where τ_k decreases from majority to minority classes in a naive manner as follows:

$$\tau_k = 1 + \eta \times (1 - \frac{k}{K}), \;\; k = 1, \ldots, K, \tag{6}$$

where η is a scale hyperparameter. Since the features of majority classes are multiplied by relatively large scaling factors compared to those of minority classes, their feature clusters occupy larger regions on the feature map during training. In this context, to prevent minority class features from crossing the decision

boundary, the feature extractor $\xi(\cdot)$ would be trained to map minority class features into denser regions. Note that during the test phase, the features extracted from test samples are not multiplied by the scaling factor, ensuring that the classifier predicts classes for test samples with densified features. According to Li *et al*. [28], FCC effectively enhances the separability of feature clusters of minority classes in CIL. However, in CISSL, it may fail to appropriately densify minority class features because the class distribution of the unlabeled set cannot be considered in the scaling factor τ_k. In this paper, we refer to the scaling factor in Eq. (6) as the naive feature multiplier.

4 Methodology

4.1 Rebalancing using estimated class distribution (RECD)

As described in Sec. 1, RECD incorporates the unknown class distribution of the unlabeled data by rebalancing the classifier based on the estimated class distribution of the unlabeled data. RECD is integrated with ABC [27], which rebalances the training loss using the 0/1 mask, as described in Eq. (1). Whereas ABC generates 0/1 masks for unlabeled samples based on the class distribution of the labeled set, RECD replaces the parameter of 0/1 masks for unlabeled samples using the estimated class distribution of the unlabeled set.

Specifically, to estimate the unknown class distribution of the unlabeled set, $p(y_u)$, RECD first computes the average class predictions of ABC on the entire unlabeled set as $\frac{1}{M}\sum_{m=1}^{M} p_\theta(y_u|u_m)$, where $p_\theta(y_u|u_m) = f_{abc}(\xi(\alpha(u_m)))$. Then, using MC approximation, $\frac{1}{M}\sum_{m=1}^{M} p_\theta(y_u|u_m)$ becomes $p_\theta(y_u)$ as follows:

$$\frac{1}{M}\sum p_\theta(y_u|u_m) \approx \int p_\theta(y_u|u)p(u)du = p_\theta(y_u). \tag{7}$$

In Eq. (7), $p_\theta(y_u)$ can be regarded as the *classifier's prior* for the unlabeled samples, and we use this classifier's prior as the estimation of the class distribution of the unlabeled set $p(y_u)$ based on that the classifier's prior is highly correlated with the class distribution, as analyzed in Fig. 3.

Using the estimated class distribution $p_\theta(y_u)$, RECD modifies the sampling distribution of the 0/1 mask for unlabeled samples in Eq. (1) as follows:

$$M(u_b^m) = \mathcal{B}\left(\frac{\min(p_\theta(y_u))}{p_\theta(y_u)_{\hat{q}_b^m}}\right). \tag{8}$$

By setting the parameter of the 0/1 mask based on $p_\theta(y_u)$, RECD rebalances the training loss of the unlabeled samples to an appropriate extent even when the class distributions of the labeled and unlabeled sets severely mismatch. The total training loss of RECD is described in detail in Sec. 4.3.

RECD may increase computational complexity compared to the original ABC because, as shown in Eq. (7), computing class predictions of ABC for the entire unlabeled set for each iteration or epoch would require a substantial amount of

computation, especially for large-scale datasets such as LSUN [49] or iNaturallist [7]. To address this issue, RECD replaces the naive average $\frac{1}{M}\sum p_\theta(y_u|u_m)$ with an EMA of the class predictions on the unlabeled samples, which is updated for each iteration of the training as follows:

$$p_\theta(y_u)_{new} = \rho \times p_\theta(y_u)_{old} + (1-\rho) \times p_\theta(y_u)_{batch}, \tag{9}$$

where ρ is a hyperparameter for the EMA update, $p_\theta(y_u)_{old}$ is $p_\theta(y_u)$ before the update, and $p_\theta(y_u)_{batch}$ is the average class prediction on the weakly augmented unlabeled minibatch, $\frac{1}{B_u}\sum_{b=1}^{B_u} q_b^m = \frac{1}{B_u}\sum_{b=1}^{B_u} f_{abc}(\xi(\alpha(u_b^m)))$. Updating the EMA of the class predictions on unlabeled samples adds negligible computational cost compared to the original ABC because RECD reuses q_b^m that is calculated to conduct consistency regularization for ABC, as in Eq. (3).

4.2 Adaptively adjusted feature multiplier (AAFM)

Although the naive feature multiplier effectively densifies feature clusters of the minority classes in CIL, its efficacy may diminish in CISSL due to its lack of consideration for the class distribution of the unlabeled set. In addition, the naive feature multiplier linearly reduces from majority to minority classes as shown in Eq. (6), without considering the degree of class imbalance in the training set. These limitations may lead to inappropriate scaling factors for the extracted features, resulting in insufficient densification or unstable training, as discussed in Sec. 5.3. To address this issue by incorporating both the class imbalance ratio of the labeled set and the unknown class distribution of the unlabeled set, we introduce an advanced version of the naive feature multiplier, AAFM. For the labeled samples, AAFM for the kth class, τ_k^l, is calculated as follows:

$$\tau_k^l = 1 + \eta \times (\frac{N_k}{N_1} \times C_{lk}^t), \;\; k = 1, \ldots, K, \tag{10}$$

where η is a scale hyperparameter similar to η in Eq. (6), and t is a hyperparameter for C_{lk}, which represents the EMA of confidence for labeled samples of the kth class, which is updated for each iteration of the training process as follows:

$$C_{lk} = \omega \times C_{lk}^{old} + (1-\omega) \times C_{lk}^{batch}, \tag{11}$$

where ω is a hyperparameter for the EMA update, C_{lk}^{old} is C_{lk} before the update, and C_{lk}^{batch} is the average confidence for class k on a minibatch, which is calculated as $\frac{1}{n_k}\sum_{b=1}^{B_l} f_{abc}(\xi(\alpha(x_b^m)))_k \times \mathbb{1}(y_b^m = k)$. Here, n_k denotes the number of samples for the kth class on a minibatch. Note that C_{lk} is updated only when $n_k \geq 1$. By multiplying C_{lk}^t to the second term on the right-hand side of Eq. (10), we stabilize the training process, assigning smaller AAFMs to classes whose feature clusters are located near the decision boundary.

We also multiply AAFM τ_k^u with the features extracted from strongly augmented unlabeled samples, where τ_k^u incorporates the unknown class distribution

of the unlabeled set by reusing the estimated class distribution $p_\theta(y_u)$ as follows:

$$\tau_k^u = 1 + \eta \times (\frac{p_\theta(y_u)_k}{\max(p_\theta(y_u))} \times C_{uk}^t), \;\; k = 1, \ldots, K. \tag{12}$$

In Eq. (12), C_{uk} is an EMA of confidence for unlabeled samples predicted as the kth class, which is updated for each iteration of the training process as follows:

$$C_{uk} = \omega \times C_{uk}^{old} + (1 - \omega) \times C_{uk}^{batch}, \tag{13}$$

where $C_{uk}^{batch} = \frac{1}{m_k} \sum_{b=1}^{B_u} f_{abc}(\xi(\alpha(u_b^m)))_k \times \mathbb{1}(\hat{q}_b^m = k)$, with $m_k = \sum_{b=1}^{B_u} \mathbb{1}(\hat{q}_b^m = k)$ and $\hat{q}_b^m = \arg\max_k (q_b^m)_k$. C_{uk} is updated only when $m_k \geq 1$. Since τ_k^u incorporates the unknown class distribution of the unlabeled set, AAFM can densify feature clusters of minority classes to an appropriate extent even when the class distributions of the labeled and unlabeled sets significantly mismatch.

4.3 End-to-end training of RECD

Fig. 2 shows the end-to-end training process of the proposed algorithm for labeled and unlabeled data, respectively. The total training loss of RECD, denoted as L_{RECD}, closely resembles that of ABC in Eq. (4), and is formulated as follows:

$$L_{RECD} = L_{cls} + L_{con} + L_{back}, \tag{14}$$

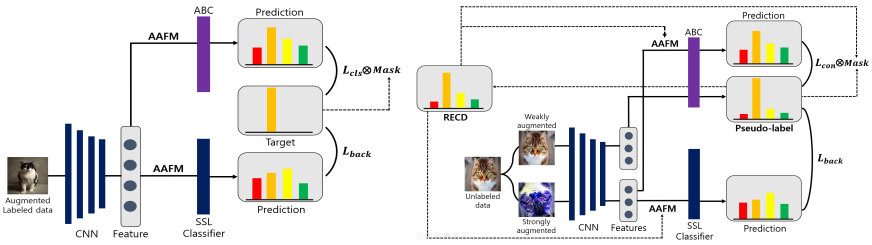
where L_{cls} and L_{con} are improved by applying RECD and AAFM as follows:

$$L_{cls} = \frac{1}{B_l} \sum_{b=1}^{B_l} M(x_b^m) \mathcal{H}(p_b^m, f_{abc}(\xi(\alpha(x_b^m)) \times \tau_{y_b^m}^l)), \tag{15}$$

$$L_{con} = \frac{1}{B_u} \sum_{b=1}^{B_u} \sum_{i=1}^{2} M(u_b^m) \mathcal{H}(q_b^m, f_{abc}(\xi(\mathcal{A}_i(u_b^m)) \times \tau_{\hat{q}_b^m}^u)), \tag{16}$$

$$M(x_b^m) = \mathcal{B}(\frac{N_K}{N_{y_b^m}}) \quad \text{and} \quad M(u_b^m) = \mathcal{B}(\frac{\min(p_\theta(y_u))}{p_\theta(y_u)_{\hat{q}_b^m}}), \tag{17}$$

where p_b^m is the one-hot version of y_b^m, $q_b^m = f_{abc}(\xi(\alpha(u_b^m)))$, and $\tau_{y_b^m}^l$ and $\tau_{\hat{q}_b^m}^u$ are AAFMs corresponding to y_b^m and $\hat{q}_b^m$, respectively. To improve the quality of features learned by the backbone SSL algorithm, L_{back} is also adjusted with two modifications. First, AAFM is applied to further densify feature clusters of minority classes during the learning of the backbone SSL algorithm. Second, to train the backbone algorithm with unbiased pseudo-labels, pseudo-labels of the backbone algorithm are generated by f_{abc} instead of the classifier of the backbone algorithm. The pseudo-code of RECD is presented in Appendix D.



(a) Balanced training with labeled data **(b)** Balanced training with unlabeled data

Fig. 2: End-to-end training of the proposed algorithm

5 Experiments

5.1 Experimental setup

Following previous CISSL studies [11, 20, 23, 34, 40], we conducted experiments on **CIFAR-10-LT [8], CIFAR-100-LT [8], STL10-LT [20]**, and **Small-ImageNet-127 [11]**. The detailed descriptions for the datasets are provided in Appendix E.1. We evaluated the classification performance using balanced accuracy (bACC) (referred to as the averaged class recall in [11]) [15] and geometric mean (GM) [22]. The performance of the proposed algorithm was compared with the following baseline algorithms: **vanilla algorithm**, **CIL algorithms** [3, 16, 19], **SSL algorithms** [1, 38], and **CISSL algorithms** [11, 14, 20, 23, 24, 27, 40, 44, 46]. Algorithms such as DASO [34] and ACR [47], whose classification performances were measured under different experimental settings from ours, were compared with RECD in Appendix G. The detailed descriptions of the baseline algorithms and training setups are provided in Appendix E.2.

5.2 Experimental results

Tab. 1 summarizes the classification performances of the baseline algorithms and RECD on CIFAR-10-LT under $\gamma_l = \gamma_u$. First, the CISSL algorithms achieved higher performances than the vanilla algorithm, CIL algorithms, and SSL algorithms. These results verify the importance of mitigating class imbalance and utilizing unlabeled samples. Furthermore, RECD achieved the highest performance among the CISSL algorithms. Specifically, RECD showed a tendency to outperform other CISSL algorithms by a large margin as the class imbalance ratio increased, indicating its effectiveness in mitigating class imbalance. These results may be attributed to the unique ability of the proposed algorithm to mitigate class imbalance in the feature map, as we analyze in Sec. 5.3.

Tab. 2 summarizes the performances of the baseline CISSL algorithms and RECD on CIFAR-100-LT. RECD outperformed the baseline algorithms. Given that CIFAR-100-LT has 100 classes, these results suggest that RECD may also be suitable for datasets with a large number of classes. Considering that classes with the fewest labeled data points have only three labeled samples under $\gamma = 50$,

Table 1: Comparison of bACC/GM on CIFAR-10-LT under $\gamma = \gamma_l = \gamma_u$

CIFAR-10-LT ($\gamma = \gamma_l = \gamma_u$)			
Algorithm	$\gamma = 50$	$\gamma = 100$	$\gamma = 150$
Vanilla (Cross-Entropy Loss)	$65.2_{\pm 0.05}$/ $61.1_{\pm 0.09}$	$58.8_{\pm 0.13}$/ $58.2_{\pm 0.11}$	$55.6_{\pm 0.43}$/ $44.0_{\pm 0.98}$
Re-sampling [17]	$64.3_{\pm 0.48}$/ $60.6_{\pm 0.67}$	$55.8_{\pm 0.47}$/ $45.1_{\pm 0.30}$	$52.2_{\pm 0.05}$/ $38.2_{\pm 1.49}$
LDAM-DRW [3]	$68.9_{\pm 0.07}$/ $67.0_{\pm 0.08}$	$62.8_{\pm 0.17}$/ $58.9_{\pm 0.60}$	$57.9_{\pm 0.20}$/ $50.4_{\pm 0.30}$
cRT [19]	$67.8_{\pm 0.13}$/ $66.3_{\pm 0.15}$	$63.2_{\pm 0.45}$/ $59.9_{\pm 0.40}$	$59.3_{\pm 0.10}$/ $54.6_{\pm 0.72}$
FixMatch [38]	$79.2_{\pm 0.33}$/ $77.8_{\pm 0.36}$	$71.5_{\pm 0.72}$/ $66.8_{\pm 1.51}$	$68.4_{\pm 0.15}$/ $59.9_{\pm 0.43}$
FixMatch+DARP+cRT [19, 20]	$85.8_{\pm 0.43}$/ $85.6_{\pm 0.56}$	$82.4_{\pm 0.26}$/ $81.8_{\pm 0.17}$	$79.6_{\pm 0.42}$/ $78.9_{\pm 0.35}$
FixMatch+CReST+LA [32, 46]	$85.6_{\pm 0.36}$/ $81.9_{\pm 0.45}$	$81.2_{\pm 0.70}$/ $74.5_{\pm 0.99}$	$71.9_{\pm 2.24}$/ $64.4_{\pm 1.75}$
FixMatch+ABC [27]	$85.6_{\pm 0.26}$/ $85.2_{\pm 0.29}$	$81.1_{\pm 1.14}$/ $80.3_{\pm 1.29}$	$77.3_{\pm 1.25}$/ $75.6_{\pm 1.65}$
FixMatch+CoSSL [11]	$86.8_{\pm 0.30}$/ $86.6_{\pm 0.25}$	$83.2_{\pm 0.49}$/ $82.7_{\pm 0.60}$	$80.3_{\pm 0.55}$/ $79.6_{\pm 0.57}$
FixMatch+SAW+LA [23, 32]	$86.2_{\pm 0.15}$/ $83.9_{\pm 0.35}$	$80.7_{\pm 0.15}$/ $77.5_{\pm 0.21}$	$73.7_{\pm 0.06}$/ $71.2_{\pm 0.17}$
FixMatch+Adsh [14]	$83.4_{\pm 0.06}$/ -	$76.5_{\pm 0.35}$/ -	$71.5_{\pm 0.30}$/ -
FixMatch+DebiasPL [44]	-/ -	$80.6_{\pm 0.50}$/ -	-/ -
FixMatch+UDAL [24]	$86.5_{\pm 0.29}$/ -	$81.4_{\pm 0.39}$/ -	$77.9_{\pm 0.33}$/ -
FixMatch+L2AC [40]	-/ -	$82.1_{\pm 0.57}$/ $81.5_{\pm 0.64}$	$77.6_{\pm 0.53}$/ $75.8_{\pm 0.71}$
FixMatch+RECD	**87.3**$_{\pm 0.18}$/ **87.2**$_{\pm 0.19}$	**84.0**$_{\pm 0.13}$/ **83.6**$_{\pm 0.16}$	**80.6**$_{\pm 0.53}$/ **79.7**$_{\pm 0.66}$
ReMixMatch [1]	$81.5_{\pm 0.26}$/ $80.2_{\pm 0.32}$	$73.8_{\pm 0.38}$/ $69.5_{\pm 0.84}$	$69.9_{\pm 0.47}$/ $62.5_{\pm 0.35}$
ReMixMatch+DARP+cRT [19, 20]	$87.3_{\pm 0.61}$/ $87.0_{\pm 0.11}$	$83.5_{\pm 0.07}$/ $83.1_{\pm 0.09}$	$79.7_{\pm 0.54}$/ $78.9_{\pm 0.49}$
ReMixMatch+CReST+LA [32, 46]	$84.2_{\pm 0.11}$/ -	$81.3_{\pm 0.34}$/ -	$79.2_{\pm 0.31}$/ -
ReMixMatch+ABC [27]	$87.9_{\pm 0.47}$/ $87.6_{\pm 0.51}$	$84.5_{\pm 0.32}$/ $84.1_{\pm 0.36}$	$80.5_{\pm 1.18}$/ $79.5_{\pm 1.36}$
ReMixMatch+CoSSL [11]	$87.7_{\pm 0.21}$/ $87.6_{\pm 0.25}$	$84.1_{\pm 0.56}$/ $83.7_{\pm 0.66}$	$81.3_{\pm 0.83}$/ $80.5_{\pm 0.76}$
ReMixMatch+SAW+cRT [19, 23]	$87.6_{\pm 0.21}$/ $87.4_{\pm 0.26}$	**85.4**$_{\pm 0.32}$/ $83.9_{\pm 0.21}$	$79.9_{\pm 0.15}$/ $79.9_{\pm 0.12}$
ReMixMatch+RECD	**88.1**$_{\pm 0.13}$/ **87.9**$_{\pm 0.12}$	**85.4**$_{\pm 0.40}$/ **85.2**$_{\pm 0.41}$	**82.5**$_{\pm 0.33}$/ **82.1**$_{\pm 0.36}$

these results also demonstrate the possibility that RECD can outperform the baseline algorithms even when labeled samples are extremely scarce.

Tab. 3 summarizes the performances of the baseline CISSL algorithms and RECD on Small-ImageNet-127. For both 32×32 and 64×64 version, RECD significantly outperformed the baseline algorithms. Given that the class distribution of the test set of Small-ImageNet-127 is also imbalanced like the training set, these results show that RECD may also be suitable for datasets with imbalanced test sets. Moreover, given that Small-ImageNet-127 is an large-scale dataset, these results verify that RECD can also be applied to large-scale datasets.

Tab. 4 summarizes the performances of the baseline algorithms and RECD evaluated on CIFAR-10-LT under $\gamma_l \neq \gamma_u$ and STL-10-LT under unknown

Table 2: Comparison of bACC on CIFAR-100-LT under $\gamma = \gamma_l = \gamma_u$

CIFAR-100-LT ($\gamma = \gamma_l = \gamma_u$)		
Algorithm	$\gamma = 20$	$\gamma = 50$
FixMatch [38]	$49.6_{\pm 0.78}$	$42.1_{\pm 0.33}$
FixMatch+DARP [20]	$50.8_{\pm 0.77}$	$43.1_{\pm 0.54}$
FixMatch+DARP+cRT [19, 20]	$51.4_{\pm 0.68}$	$44.9_{\pm 0.54}$
FixMatch+CReST [46]	$51.8_{\pm 0.12}$	$44.9_{\pm 0.50}$
FixMatch+CReST+LA [32, 46]	$52.9_{\pm 0.07}$	$47.3_{\pm 0.17}$
FixMatch+ABC [27]	$53.3_{\pm 0.79}$	$46.7_{\pm 0.26}$
FixMatch+CoSSL [11]	$53.9_{\pm 0.78}$	$47.6_{\pm 0.57}$
FixMatch+RECD	**54.6**$_{\pm 0.36}$	**47.8**$_{\pm 0.17}$
ReMixMatch [1]	$51.6_{\pm 0.43}$	$44.2_{\pm 0.59}$
ReMixMatch+DARP [20]	$51.9_{\pm 0.35}$	$44.7_{\pm 0.66}$
ReMixMatch+DARP+cRT [19, 20]	$54.5_{\pm 0.42}$	$48.5_{\pm 0.91}$
ReMixMatch+CReST [46]	$51.3_{\pm 0.34}$	$45.5_{\pm 0.76}$
ReMixMatch+CReST+LA [32, 46]	$51.9_{\pm 0.60}$	$46.6_{\pm 1.14}$
ReMixMatch+ABC [27]	$55.6_{\pm 0.35}$	$47.9_{\pm 0.10}$
ReMixMatch+CoSSL [11]	$55.8_{\pm 0.62}$	$48.9_{\pm 0.61}$
ReMixMatch+RECD	**55.9**$_{\pm 0.36}$	**49.5**$_{\pm 0.21}$

Table 3: Comparison of bACC on Small-ImageNet-127

Small-ImageNet-127 ($\gamma = \gamma_l = \gamma_u$)		
Algorithm	32×32	64×64
FixMatch [38]	29.7	42.3
FixMatch+DARP [20]	30.5	42.5
FixMatch+DARP +cRT [19, 20]	39.7	51.0
FixMatch+CReST+ [46]	32.5	44.7
FixMatch+CReST++LA [32, 46]	40.9	55.9
FixMatch+CoSSL [11]	43.7	53.8
FixMatch+RECD	**47.3**	**59.5**

γ_u. RECD outperformed the baseline algorithms by large margins. These results demonstrate that RECD effectively mitigates class imbalance even when $\gamma_l \neq \gamma_u$, by considering the unknown class distribution of the unlabeled set. Notably, ReMixMatch+RECD outperformed the baseline algorithms that use ReMixMatch* [20] (ReMixMatch that uses the estimated class distribution of the unlabeled set) as the backbone SSL algorithm. These results implicitly demonstrate that RECD estimates class distribution better than the previous estimation algorithm. Additionally, under severe class distribution mismatch between the labeled and unlabeled sets, LA degraded the performance of ReMixMatch*+DARP and ReMixMatch*+SAW. This may be because the unknown class distribution of the unlabeled set cannot be incorporated into LA, emphasizing the importance of considering the class distribution of the unlabeled set.

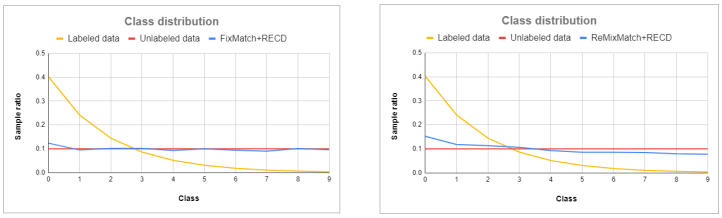
Table 4: Comparison of bACC/GM on CIFAR-10-LT and STL-10-LT under $\gamma_l \neq \gamma_u$.

	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 [38]	$68.9_{\pm 1.95}$/ $42.8_{\pm 8.11}$	$73.9_{\pm 0.25}$/ $70.5_{\pm 0.52}$	$69.6_{\pm 0.60}$/ $62.6_{\pm 1.11}$	$72.9_{\pm 0.09}$/ $69.6_{\pm 0.01}$	$63.4_{\pm 0.21}$/ $52.6_{\pm 0.09}$
/+DARP [20]	$85.4_{\pm 0.55}$/ $85.0_{\pm 0.65}$	$77.3_{\pm 0.17}$/ $75.5_{\pm 0.21}$	$72.9_{\pm 0.24}$/ $69.5_{\pm 0.18}$	$77.8_{\pm 0.33}$/ $76.5_{\pm 0.40}$	$69.9_{\pm 1.77}$/ $65.4_{\pm 3.07}$
/+DARP+LA [20, 32]	$86.6_{\pm 1.11}$/ $86.2_{\pm 1.15}$	$82.3_{\pm 0.32}$/ $81.5_{\pm 0.29}$	$78.9_{\pm 0.23}$/ $77.7_{\pm 0.06}$	$78.6_{\pm 0.30}$/ $77.4_{\pm 0.40}$	$71.9_{\pm 0.49}$/ $68.7_{\pm 0.51}$
/+DARP+cRT [19, 20]	$87.0_{\pm 0.70}$/ $86.8_{\pm 0.67}$	$82.7_{\pm 0.21}$/ $82.3_{\pm 0.25}$	$80.7_{\pm 0.44}$/ $80.2_{\pm 0.61}$	$79.3_{\pm 0.23}$/ $78.7_{\pm 0.21}$	$74.1_{\pm 0.61}$/ $73.1_{\pm 1.21}$
/+ABC [27]	$82.7_{\pm 0.49}$/ $81.9_{\pm 0.68}$	$82.7_{\pm 0.64}$/ $82.0_{\pm 0.76}$	$78.4_{\pm 0.87}$/ $77.2_{\pm 1.07}$	$79.1_{\pm 0.46}$/ $78.1_{\pm 0.57}$	$73.8_{\pm 0.15}$/ $72.1_{\pm 0.15}$
/+SAW [23]	$81.2_{\pm 0.68}$/ $80.2_{\pm 0.91}$	$79.8_{\pm 0.25}$/ $79.1_{\pm 0.32}$	$74.5_{\pm 0.97}$/ $72.5_{\pm 1.37}$	$78.3_{\pm 0.25}$/ $77.0_{\pm 0.19}$	$71.9_{\pm 0.81}$/ $69.0_{\pm 0.81}$
/+SAW+LA [23, 32]	$84.5_{\pm 0.68}$/ $84.1_{\pm 0.78}$	$82.9_{\pm 0.38}$/ $82.6_{\pm 0.38}$	$79.1_{\pm 0.81}$/ $78.6_{\pm 0.91}$	-/ -	-/ -
/+SAW+cRT [19, 23]	$84.6_{\pm 0.23}$/ $84.4_{\pm 0.26}$	$81.6_{\pm 0.38}$/ $81.3_{\pm 0.32}$	$77.6_{\pm 0.40}$/ $77.1_{\pm 0.41}$	-/ -	-/ -
/+RECD	$\mathbf{90.2}_{\pm 0.57}$/ $\mathbf{90.0}_{\pm 0.64}$	$\mathbf{85.6}_{\pm 0.15}$/ $\mathbf{85.3}_{\pm 0.15}$	$\mathbf{82.3}_{\pm 0.39}$/ $\mathbf{81.8}_{\pm 0.45}$	$\mathbf{81.4}_{\pm 0.28}$/ $\mathbf{80.6}_{\pm 0.38}$	$\mathbf{79.0}_{\pm 0.48}$/ $\mathbf{78.1}_{\pm 0.54}$
ReMixMatch [1]	$48.3_{\pm 0.14}$/ $19.5_{\pm 0.85}$	$75.1_{\pm 0.43}$/ $71.9_{\pm 0.77}$	$72.5_{\pm 0.10}$/ $68.2_{\pm 0.32}$	$67.8_{\pm 0.45}$/ $61.1_{\pm 0.92}$	$60.1_{\pm 1.18}$/ $44.9_{\pm 1.52}$
/+ABC [27]	$76.4_{\pm 5.34}$/ $74.8_{\pm 6.05}$	$85.2_{\pm 0.20}$/ $84.7_{\pm 0.25}$	$80.4_{\pm 0.40}$/ $80.0_{\pm 0.44}$	$76.8_{\pm 0.52}$/ $74.8_{\pm 0.64}$	$71.2_{\pm 1.37}$/ $67.4_{\pm 1.89}$
ReMixMatch* [1]	$85.0_{\pm 1.35}$/ $84.3_{\pm 1.55}$	$77.0_{\pm 0.12}$/ $74.7_{\pm 0.04}$	$72.8_{\pm 0.10}$/ $68.8_{\pm 0.21}$	$76.7_{\pm 0.15}$/ $73.9_{\pm 0.32}$	$67.7_{\pm 0.46}$/ $60.3_{\pm 0.76}$
/+DARP [20]	$86.9_{\pm 0.10}$/ $86.4_{\pm 0.15}$	$77.4_{\pm 0.22}$/ $75.0_{\pm 0.25}$	$73.2_{\pm 0.11}$/ $69.2_{\pm 0.31}$	$79.4_{\pm 0.07}$/ $78.2_{\pm 0.10}$	$70.9_{\pm 0.44}$/ $67.0_{\pm 1.62}$
/+DARP+LA [20, 32]	$81.8_{\pm 0.45}$/ $80.9_{\pm 0.40}$	$83.9_{\pm 0.42}$/ $83.4_{\pm 0.45}$	$81.1_{\pm 0.20}$/ $80.3_{\pm 0.26}$	$80.6_{\pm 0.45}$/ $79.6_{\pm 0.55}$	$76.8_{\pm 0.60}$/ $74.8_{\pm 0.68}$
/+DARP+cRT [19, 20]	$88.7_{\pm 0.25}$/ $88.5_{\pm 0.25}$	$83.5_{\pm 0.53}$/ $83.1_{\pm 0.51}$	$80.9_{\pm 0.25}$/ $80.3_{\pm 0.31}$	$80.9_{\pm 0.53}$/ $80.0_{\pm 0.46}$	$76.7_{\pm 0.50}$/ $74.9_{\pm 0.70}$
/+SAW [23]	$87.0_{\pm 0.75}$/ $86.4_{\pm 0.85}$	$80.6_{\pm 1.57}$/ $79.2_{\pm 2.19}$	$77.6_{\pm 0.76}$/ $76.0_{\pm 0.93}$	$82.0_{\pm 0.55}$/ $81.0_{\pm 0.64}$	$79.2_{\pm 0.44}$/ $77.9_{\pm 0.52}$
/+SAW+LA [23, 32]	$74.2_{\pm 1.49}$/ $65.1_{\pm 2.36}$	$84.8_{\pm 1.07}$/ $82.4_{\pm 2.32}$	$81.3_{\pm 2.42}$/ $80.9_{\pm 2.47}$	-/ -	-/ -
/+SAW+cRT [19, 23]	$88.8_{\pm 0.79}$/ $88.6_{\pm 0.83}$	$84.5_{\pm 0.78}$/ $83.6_{\pm 1.27}$	$82.4_{\pm 0.10}$/ $82.0_{\pm 0.10}$	-/ -	-/ -
ReMixMatch+RECD	$\mathbf{90.3}_{\pm 0.40}$/ $\mathbf{90.2}_{\pm 0.41}$	$\mathbf{86.8}_{\pm 0.17}$/ $\mathbf{86.6}_{\pm 0.18}$	$\mathbf{83.9}_{\pm 0.11}$/ $\mathbf{83.7}_{\pm 0.12}$	$\mathbf{84.9}_{\pm 0.41}$/ $\mathbf{84.4}_{\pm 0.47}$	$\mathbf{82.5}_{\pm 0.27}$/ $\mathbf{81.7}_{\pm 0.31}$

In addition to the above results, we demonstrate the effectiveness of RECD with fewer labeled samples in Appendix J. Furthermore, we verify that RECD can be effectively combined with recent SSL algorithms by conducting experiments using FreeMatch [45] as the backbone SSL algorithm in Appendix I.

5.3 Detailed analyses on the proposed algorithm

We argued that RECD effectively estimates the unknown class distribution of the unlabeled set and mitigates class imbalance using this estimation. To verify this argument, we plotted the true class distribution of the unlabeled set, the class distribution of the unlabeled set estimated by RECD, and the class distribution of the labeled set in Fig. 3. We can observe that RECD accurately estimated the unknown class distribution of the unlabeled set, even when it significantly differed from that of the labeled set. This accurate class distribution estimation enabled the proposed algorithm to be trained with a properly rebalanced training loss and accurately calculated AAFM, mitigating class imbalance as in Fig. 4.



(a) FixMatch+RECD **(b)** RemixMatch+RECD

Fig. 3: True class distributions of the labeled and unlabeled sets, along with the class distribution of the unlabeled set estimated by RECD. FixMatch+RECD and RemixMatch+RECD were trained on CIFAR-10-LT with $\gamma_l = 100$ and $\gamma_u = 1$.

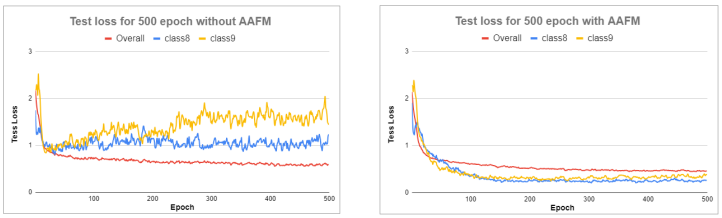
Fig. 4 presents the confusion matrices of class predictions on the test set of CIFAR-10 generated by each algorithm trained on the training set of CIFAR-10-LT under $\gamma_l = 100$ and $\gamma_u = 1$. The (i,j) position of each confusion matrix represents the proportion of the ith class samples classified as the jth class. In the figures, the left three algorithms often misclassified minority classes as majority classes. On the other hand, with both RECD and AAFM, the algorithm classified most samples of minority classes accurately. These results demonstrate that both RECD and AAFM effectively mitigate class imbalance when the class distribution of the labeled and unlabeled sets significantly mismatch.

Additionally, we argued that AAFM not only reduces the test losses for minority classes but also stabilizes the training process. To verify this, we measured the overall test loss and the test losses of minority classes for FixMatch+RECD without AAFM and FixMatch+RECD with AAFM trained on CIFAR-10-LT. From Fig. 5, we can observe that the test losses for minority classes tended to increase as the training proceeds when AAFM was not used. On the other hand, the overall test loss and the test losses for the minority classes consistently decreased when AAFM was used. We can also observe that whereas the test losses for minority classes fluctuated when AAFM was not used, they stably decreased when AAFM was used. In addition, we compared the performance of the naive feature multiplier and AAFM with FixMatch+RECD in Appendix K. Finally,

(a) FixMatch **(b)** Without RECD **(c)** Without AAFM **(d)** RECD+AAFM

Fig. 4: Confusion matrices of class predictions on the test set of CIFAR-10 generated by each algorithm (FixMatch+RECD with the described condition) trained on the training set of CIFAR-10-LT under $\gamma_l = 100$ and $\gamma_u = 1$.

we argue that RECD also can be combined with various CISSL algorithms. To verify this, we conducted additional experiments using DARP, and demonstrated RECD-based DARP outperformed the original DARP in Appendix F.



(a) Without AAFM **(b)** With AAFM

Fig. 5: Overall test loss and test losses for minority classes (Classes 8 and 9). The algorithms were trained on CIFAR-10-LT under $\gamma_l = 100$ and $\gamma_u = 1$.

6 Conclusion

We proposed a novel CISSL algorithm, RECD, which addresses the class distribution mismatch between labeled and unlabeled sets. RECD estimates the class distribution of the unlabeled set through MC approximation and uses it for rebalancing the classifier. Additionally, RECD reuses the estimated class distribution for AAFM to mitigate class imbalance in the feature map by densifying feature clusters of minority classes. Experimental results demonstrate that RECD outperforms the baseline algorithms under both scenarios where the class distributions of the labeled and unlabeled sets match and mismatch. Detailed analyses on RECD verify that the effectiveness of each component of RECD for mitigating class imbalance. However, the estimated class distribution of unlabeled set through MC approximation might be inaccurate when unlabeled training samples are scarce. In future work, we aim to improve the robustness of class distribution estimation, even with a limited amount of unlabeled samples.

Acknowledgements

This research was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (2023R1A2C2005453, RS-2023-00218913).

References

1. Berthelot, D., Carlini, N., an Alex Kurakin, E.D.C., Sohn, K., Zhang, H., Raffel, C.: Remixmatch: Semi-supervised learning with distribution matching and augmentation anchoring. In: ICLR (2020)
2. Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., Raffel, C.: Mixmatch: A holistic approach to semi-supervised learning. In: NeurIPS (2019)
3. Cao, K., Wei, C., Gaidon, A., Arechiga, N., Ma, T.: Learning imbalanced datasets with label-distribution-aware margin loss. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 32. Curran Associates, Inc. (2019)
4. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research pp. 321–357 (2002)
5. Chu, P., Bian, X., Liu, S., Ling, H.: Feature space augmentation for long-tailed data. In: ECCV. vol. 12374, pp. 694–710 (2020)
6. Cubuk, E.D., Zoph, B., Shlens, J., Le, Q.: Randaugment: Practical automated data augmentation with a reduced search space. In: NeurIPS (2020)
7. Cui, Y., Jia, M., Lin, T.Y., Song, Y., Belongie, S.: Large scale fine-grained categorization and domain-specific transfer learning. In: CVPR. pp. 4109–4118 (2018)
8. Cui, Y., Jia, M., Lin, T.Y., Song, Y., Belongie, S.: Class-balanced loss based on effective number of samples. In: CVPR. pp. 9268–9277 (2019)
9. DeVries, T., Taylor, G.W.: Improved regularization of convolutional neural networks with cutout. arXiv preprint arXiv:1708.04552 (2017)
10. Estabrooks, A., Jo, T., Japkowicz, N.: A multiple resampling method for learning from imbalanced data sets. Computational Intelligence **20**(1), 18–36 (2004)
11. Fan, Y., Dai, D., Kukleva, A., Schiele, B.: Cossl: Co-learning of representation and classifier for imbalanced semi-supervised learning. In: CVPR. pp. 14574–14584 (2022)
12. Gidaris, S., Singh, P., Komodakis, N.: Unsupervised representation learning by predicting image rotations. In: ICLR (2018)
13. Grandvalet, Y., Bengio, Y.: Semi-supervised learning by entropy minimization. In: NeurIPS (2004)
14. Guo, L.Z., Li, Y.F.: Class-imbalanced semi-supervised learning with adaptive thresholding. In: ICML. vol. 162, pp. 8082–8094 (2022)
15. Huang, C., Li, Y., Loy, C.C., Tang, X.: Learning deep representation for imbalanced classification. In: CVPR. pp. 5375–5384 (2016)
16. JAPKOWICZ, N.: The class imbalance problem: Significance and strategies. In: Proc. 2000 International Conference on Artificial Intelligence. vol. 1, pp. 111–117 (2000)
17. Japkowicz, N.: The class imbalance problem: Significance and strategies. In: In Proceedings of the International Conference on Artificial Intelligence (2000)

18. Kang, B., Li, Y., Xie, S., Yuan, Z., Feng, J.: Exploring balanced feature spaces for representation learning. In: ICLR (2021)
19. Kang, B., Xie, S., Rohrbach, M., Yan, Z., Gordo, A., Feng, J., Kalantidis, Y.: Decoupling representation and classifier for long-tailed recognition. In: ICLR (2020)
20. Kim, J., Hur, Y., Park, S., Yang, E., Hwang, S., Shin, J.: Distribution aligning refinery of pseudo-label for imbalanced semi-supervised learning. In: NeurIPS (2020)
21. Krizhevsky, A.: Learning multiple layers of features from tiny images. Technical report, Department of Computer Science, University of Toronto (2009)
22. Kubat, M., Matwin, S., et al.: Addressing the curse of imbalanced training sets: one-sided selection. In: Icml. vol. 97, p. 179. Citeseer (1997)
23. Lai, Z., Wang, C., Gunawan, H., Cheung, S.C., Chuah, C.N.: Smoothed adaptive weighting for imbalanced semi-supervised learning: Improve reliability against unknown distribution data. In: ICML. vol. 162, pp. 11828–11843 (2022)
24. Lazarow, J., Sohn, K., Lee, C.Y., Li, C.L., Zhang, Z., Pfister, T.: Unifying distribution alignment as a loss for imbalanced semi-supervised learning. Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) pp. 5644–5653 (2023)
25. Lee, D.H.: Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In: In Workshop on challenges in representation learning (ICML) (2013)
26. Lee, H., Kim, H.: Cdmad: Class-distribution-mismatch-aware debiasing for class-imbalanced semi-supervised learning. In: CVPR. pp. 23891–23900 (2024)
27. Lee, H., Shin, S., Kim, H.: Abc: Auxiliary balanced classifier for class-imbalanced semi-supervised learning. In: NeurIPS (2021)
28. Li, J., Meng, Z., Shi, D., Song, R., Diao, X., Wang, J., , Xu, H.: Fcc: Feature clusters compression for long-tailed visual recognition. In: CVPR. pp. 24080–24089 (2023)
29. Li, K., Kong, X., Lu, Z., Wenyin, L., Yin, J.: Boosting weighted elm for imbalanced learning. Neurocomputing **128**, 15–21 (2014)
30. Liu, X.Y., Wu, J., Zhou, Z.H.: Exploratory undersampling for class-imbalance learning. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) **39**(2), 539–550 (2008)
31. Liu, Z., Miao, Z., Zhan, X., Wang, J., Gong, B., Yu., S.X.: Large-scale long-tailed recognition in an open world. In: CVPR. pp. 2537–2546 (2019)
32. Menon, A.K., Jayasumana, S., Rawat, A.S., Jain, H., Veit, A., Kumar, S.: Long-tail learning via logit adjustment. In: ICLR (2021)
33. Miyato, T., ichi Maeda, S., Koyama, M., Ishii, S.: 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)
34. Oh, Y., Kim, D.J., Kweon, I.S.: Daso: Distribution-aware semantics-oriented pseudo-label for imbalanced semi-supervised learning. In: CVPR. pp. 9786–9796 (2022)
35. Ren, J., Yu, C., Sheng, S., Ma, X., Zhao, H., Yi, S., Li, H.: Balanced meta-softmax for long-tailed visual recognition. In: NeurIPS (2020)
36. Ren, M., Zeng, W., Yang, B., Urtasun, R.: Learning to reweight examples for robust deep learning. In: ICML. vol. 80, pp. 4334–4343 (2018)
37. Sajjadi, M., Javanmardi, M., Tasdizen, T.: Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In: NeurIPS (2016)
38. Sohn, K., Berthelot, D., Li, C.L., Zhang, Z., Carlini, N., Cubuk, E.D., Kurakin, A., Zhang, H., Raffel, C.: Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In: NeurIPS (2020)

39. Verma, V., Kawaguchi, K., Lamb, A., Kannala, J., Solin, A., Bengio, Y., Lopez-Paz, D.: Interpolation consistency training for semi-supervised learning. Neural Networks **145**, 90–106 (2022)
40. Wang, R., Jia, X., Wang, Q., Wu, Y., Meng, D.: Imbalanced semi-supervised learning with bias adaptive classifier. In: ICLR (2023)
41. Wang, S., Minku, L.L., Yao, X.: A learning framework for online class imbalance learning. IEEE Symposium on Computational Intelligence and Ensemble Learning (CIEL) pp. 36–45 (2013)
42. Wang, S., Minku, L.L., Yao, X.: Resampling-based ensemble methods for online class imbalance learning. IEEE Transactions on Knowledge and Data Engineering **27**(5), 1356–1368 (2015)
43. Wang, X., Lian, L., Miao, Z., Liu, Z., Yu., S.X.: Long-tailed recognition by routing diverse distribution-aware experts. In: ICLR (2021)
44. Wang, X., Wu, Z., Lian, L., Yu, S.X.: Debiased learning from naturally imbalanced pseudo-labels. In: CVPR. pp. 14647–14657 (2022)
45. Wang, Y., Chen, H., Heng, Q., Hou, W., Fan, Y., Wu, Z., Wang, J., Savvides, M., Shinozaki, T., Raj, B., Schiele, B., Xie, X.: Freematch: Self-adaptive thresholding for semi-supervised learning. In: ICLR (2023)
46. Wei, C., Sohn, K., Mellina, C., Yuille, A., Yang, F.: Crest: A class-rebalancing self-training framework for imbalanced semi-supervised learning. In: CVPR. pp. 10857–10866 (2021)
47. Wei, T., Gan, K.: Towards realistic long-tailed semi-supervised learning: Consistency is all you need. In: CVPR. pp. 3469–3478 (2023)
48. Xiang, L., Ding, G., Han, J.: Learning from multiple experts: Self-paced knowledge distillation for long-tailed classification. In: ECCV. pp. 247–263 (2020)
49. Yu, F., Seff, A., Zhang, Y., Song, S., Funkhouser, T., Xiao, J.: Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. arXiv preprint arXiv:1506.03365 (2015)
50. Yu, Z., Li, Y., Lee, Y.J.: Inpl: Pseudo-labeling the inliers first for imbalanced semi-supervised learning. In: ICLR (2023)
51. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. In: ICLR (2018)
52. Zhang, Y., Hooi, B., Hong, L., Feng, J.: Self-supervised aggregation of diverse experts for test-agnostic long-tailed recognition. In: NeurIPS (2022)
53. Zhong, Z., Cui, J., Liu, S., Jia, J.: Improving calibration for long-tailed recognition. In: CVPR. pp. 16489–16498 (2021)