Class-Imbalanced Semi-Supervised Learning with Adaptive Thresholding

Lan-Zhe Guo [1] **Yu-Feng Li** [1]

Abstract

Semi-supervised learning (SSL) has proven to be successful in overcoming the difficulties of data labeling by leveraging unlabeled data. Previous SSL algorithms typically assume a balanced class distribution and exploit unlabeled data by assigning pseudo-labels with a fixed high-confidence prediction. However, it is well-known that real-world dataset is often imbalanced, the performance of existing SSL algorithms is seriously decreased under imbalanced class distribution since pseudo-labels that are generated based on a fixed confidence threshold are biased toward majority classes and result in low recall on minority classes. In this paper, we develop a simple yet powerful framework, whose key idea is to select a subset of pseudo-labeled examples based on thresholds that can be adaptively adjusted for different classes. Specifically, an optimization objective that considers the number of pseudo-labels being selected for each class is proposed and a highly efficient closed-form solution that produces the adaptive thresholds can be derived from the optimization. We empirically demonstrate the effectiveness of the proposal in extensive experimental settings.

1. Introduction

Machine learning, especially deep learning, has been repeatedly reported that can achieve competitive or even better performance than human beings on certain supervised learning tasks (LeCun et al., 2015). These tasks, however, crucially rely on the availability of a large number of labeled training data. In many practical tasks, large-scale well-labeled datasets are difficult to obtain, as the acquisition of labeled data requires huge human labor and financial costs (Zhou, 2017; Li et al., 2019). On the other hand, there are usually abundant unlabeled data. Therefore, it is desirable for machine learning models to work with unlabeled data.

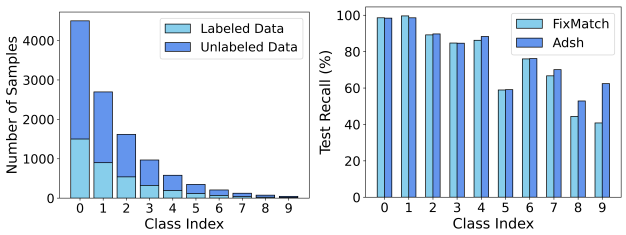
Semi-supervised learning (SSL) is one of the most promising learning paradigms to bypass the labeling cost by leveraging an abundance of unlabeled data (Chapelle et al., 2006). In much recent work, SSL can be categorized into several main classes in terms of the use of unlabeled data, such as entropy minimization (Grandvalet & Bengio, 2005), consistency regularization (Laine & Aila, 2017; Tarvainen & Valpola, 2017; Miyato et al., 2018), pseudo-labeling (Lee, 2013), and their combinations (Berthelot et al., 2019; Sohn et al., 2020; Berthelot et al., 2020; Xu et al., 2021). Due to its capability to handle both labeled and unlabeled data, SSL has been successfully applied into various tasks such as image classification (Sohn et al., 2020), object detection (Jeong et al., 2019), semantic segmentation (Souly et al., 2017), text classification (Miyato et al., 2017), etc. It has been reported in certain cases, such as image classification (Sohn et al., 2020), SSL methods can achieve the performance of purely supervised learning even when a substantial portion of the labels in a given dataset has been discarded.

All of the positive results of SSL, however, are based on a basic assumption that the class distribution is balanced in both labeled and unlabeled data, i.e., the number of examples in each class is nearly the same. Such an assumption is difficult to hold in practical applications. For example, in computer vision tasks, the frequency distribution of visual categories in our daily life is inherently imbalanced (Wang et al., 2017); in medical diagnosis tasks, a malignant lesion is rare compared to benign ones (Johnson & Khoshgoftaar, 2019), etc.

It is well-known that machine learning models suffer severe performance degradation with such an imbalanced class distribution (Dong et al., 2019). Unfortunately, the class imbalance issue can be more problematic for SSL algorithms since they generate pseudo-labels for unlabeled data from the model's biased predictions. Take the SOTA SSL algorithm FixMatch (Sohn et al., 2020) for an example, FixMatch uses the unlabeled examples with a fixed high-confidence prediction (e.g., 0.95) in classification tasks. However, the prediction confidence is biased towards the majority classes under class-imbalanced distribution, adopting a fixed threshold for all classes results in the minority classes losing too

[1]National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China. Correspondence to: Yu-Feng Li <liyf@lamda.nju.edu.cn>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.



(a) Imbalanced Dataset (b) Test Recall (%)

Figure 1. An example of experimental results on Wide ResNet-28-2 for the synthetically class-imbalanced CIFAR-10 dataset. (a) Both labeled and unlabeled datasets are class-imbalanced, where the most majority class has $100\times$ more examples than the most minority class. (b) Recall rate on a balanced test data. FixMatch selects pseudo-labels if its confidence prediction is greater than 0.95 for all classes, while the proposed Adsh algorithm selects pseudo-labels based on an adaptive class-dependent threshold. The results show that our proposal can improve the recall on minority classes, comparing to FixMatch.

many unlabeled examples with correct pseudo-labels, resulting in low recall rates. (see Figure 1). That is to say, it may not be good enough for SSL algorithms to use a fixed threshold to select pseudo-labels for all classes under class-imbalanced data distribution.

Unlike the previous works, we aim to the proposed approach has the ability to adaptively adjust the threshold for each class based on the class distribution. This inspires us to consider answering the following question in this study: **Can we design an SSL algorithm that selects pseudo-labels with adaptive thresholding?**

To this end, we propose a generic SSL algorithm with **ad**aptive thres**h**olding (Adsh) that can adaptively select pseudo-labeled examples based on a class dependent threshold during the training process. Specifically, our high-level idea is to formulate the pseudo-label selection process as an optimization objective and explicitly consider the number of pseudo-labels to be selected for every class in order to overcome the class imbalance. A highly efficient closed-form solution can be derived from the optimization objective. Then, based on the solution we obtain an adaptive thresholding technique that encodes class-wise distribution to obtain class-dependent thresholds. The proposal Adsh can be integrated with existing SSL methods like FixMatch (Sohn et al., 2020). Empirical evaluations on extensive settings demonstrate the effectiveness of Adsh comparing with the state-of-the-art SSL algorithms. For example, experimental results on CIFAR-10, SVHN, STL-10 datasets with different levels of class imbalance and different numbers of labeled data consistently show the performance improvement of our proposal. We also consider class imbalance and class distribution mismatch between labeled and unlabeled data simultaneously. Experimental results on this challenging setting also show the superiority of our proposal.

2. Related Works

This work is mainly related to class-imbalanced learning and semi-supervised learning.

Class-Imbalanced Learning. Real-world datasets usually yield a *class-imbalanced* label distribution (Liu et al., 2019) and make the standard training of machine learning models harder to generalize (Wang et al., 2017). Various algorithms have been proposed so far to address this problem (Buda et al., 2018; Johnson & Khoshgoftaar, 2019). The most commonly adopted approach is to re-balance the training objective with respect to the class-wise sample sizes. Two of such methods are representative: a) *re-weighting*, which influence the loss function by assigning relatively higher costs to examples from minor classes (Cao et al., 2019; Cui et al., 2019; Huang et al., 2019; Khan et al., 2019; 2017; Lin et al., 2017; Ren et al., 2018; Hu et al., 2019); b) *re-sampling*, which directly adjust label distribution by over-sampling for the minority class or under-sampling for the majority class, or both in order to obtain a balanced sampling distribution (Chawla et al., 2002; He & Garcia, 2009; Byrd & Lipton, 2019). However, naively re-balancing the objective usually results in over-fitting to minority classes. Recently, there are also transfer-learning based methods been proposed by transferring features from majority classes to under-represented minority classes (Hariharan & Girshick, 2017; Liu et al., 2019; Yin et al., 2019). Nevertheless, these methods assume all labels are available and can not be applied to SSL scenarios directly.

Semi-Supervised Learning. SSL methods that aim to improve model performance by leveraging unlabeled data have a long history of research (Chapelle et al., 2006). Our paper is mainly related to deep SSL that introduces SSL techniques to DNNs and achieved significant advancement in recent years (Berthelot et al., 2019; Grandvalet & Bengio, 2005; Laine & Aila, 2017; Miyato et al., 2018; Sohn et al., 2020; Tarvainen & Valpola, 2017). Typical ways of these SSL methods include training the model to fit *pseudo-labels* or optimizing a well-designed objective that does not rely on labels. For example, pseudo-labeling based methods (Lee, 2013) generate pseudo-labels for unlabeled examples and train model to predict the pseudo-labels in a supervised manner; entropy minimization based methods (Grandvalet & Bengio, 2005) encourage the model's predicted distribution to have low entropy which does not require label information; consistency regularization based methods, e.g., Temporal Ensembling (Laine & Aila, 2017), Mean-Teacher (Tarvainen & Valpola, 2017), VAT (Miyato et al., 2018), etc, produce augmentations for unlabeled examples and optimize the consistency loss between the model output on given examples and it's augmented version. There are also

methods called holistic methods that utilize these techniques simultaneously, such as MixMatch (Berthelot et al., 2019), ReMixMatch (Berthelot et al., 2020) and FixMatch (Sohn et al., 2020). These SSL algorithms are reported to achieve near supervised performance on benchmark tasks. However, in some realistic scenarios, SSL methods suffer poor performance improvement (Guo et al., 2022), e.g, when labeled and unlabeled data distribution is mismatch (Oliver et al., 2018; Guo et al., 2020a;b; Zhou et al., 2021) or when class distribution is imbalanced (Kim et al., 2020; Wei et al., 2021; Guo et al., 2021). In this paper, we mainly focus on the class-imbalanced SSL problem.

Class-Imbalanced Semi-Supervised Learning. Recently, two representative algorithms DARP (Distribution Aligning Refinery of Pseudo-label) (Kim et al., 2020) and CReST (Class-Rebalancing Self-Training) (Wei et al., 2021) have been proposed to address the class-imbalanced semi-supervised learning. Specifically, DARP refines raw biased pseudo-labels to match the true class distribution. However, the process needs to know the ground-truth class distribution of unlabeled data as a prior which is evidently impossible in real tasks. To alleviate this limitation, DARP further proposes to estimate the class distribution by assuming that the confusion matrix between labeled data and unlabeled data are the same. Unfortunately, this assumption is also inappropriate since the trained model tends to overfit the small labeled dataset and obtain a nearly perfect confusion matrix while this can not generalize well to unlabeled data. CReST adopts a self-training manner that retrains the SSL model after adaptively select pseudo-labeled data from the unlabeled set to supplement the original labeled set. Different from the classical self-training strategy, CReST samples pseudo-labels according to the label frequency in order to progressively align the class distribution (i.e., the examples are selected with higher probabilities if they are predicted as minority classes). However, CReST assumes that the class distributions between labeled data and unlabeled data are the same which is difficult to verify since we have no idea about the true class distribution of unlabeled data. These strict assumptions limit their wider applications.

3. Preliminary and Background

This section provides notations used in this paper and gives a brief review of SSL algorithms with a fixed threshold.

3.1. Problem Setting and Notations

For a K-class classification task, we are given a set of training data from an unknown distribution, which includes N labeled examples $\mathcal{D}^l = \{(\mathbf{x}_1^l, \mathbf{y}_1^l), \cdots, (\mathbf{x}_N^l, \mathbf{y}_N^l)\}$ and M unlabeled examples $\mathcal{D}^u = \{\mathbf{x}_1^u, \cdots, \mathbf{x}_M^u\}$ where $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^d$ denote the input d-dimensional feature vector and $\mathbf{y} \in \mathcal{Y}$ are corresponding one-hot label. The number of examples in class k under $\mathcal{D}^l$ and $\mathcal{D}^u$ are denoted by N_k and M_k, respectively, i.e., $\sum_{k=1}^K N_k = N$ and $\sum_{k=1}^K M_k = M$. Without loss of generality, we assume that the classes are sorted in descending order, i.e., $N_1 \geq N_2 \geq, \cdots, \geq N_K$ and $M_1 \geq M_2 \geq, \cdots, \geq M_K$. We measure the degree of class imbalance by *imbalance ratio*, which is defined as $\gamma_l = \frac{N_1}{N_K}$ and $\gamma_u = \frac{M_1}{M_K}$ for labeled and unlabeled data respectively. γ_l and γ_u could be much larger than 1 and it is noteworthy that they are usually not the same in practical tasks. The goal is to learn a model $f(\mathbf{x}; \theta) : \mathcal{X} \to \mathcal{Y}$ that generalizes well under a class-balanced test criterion, where θ is the model parameter.

The training loss of an SSL algorithm usually contains supervised loss $\mathcal{L}_s$ and unsupervised loss $\mathcal{L}_u$ with a trade-off parameter $\lambda_u > 0$: $\mathcal{L}_s + \lambda_u \mathcal{L}_u$, where $\mathcal{L}_s$ is constructed on $\mathcal{D}^l$ and $\mathcal{L}_u$ is constructed on $\mathcal{D}^u$. Typically, $\mathcal{L}_s$ applies the standard cross-entropy loss on labeled examples:

$$\begin{aligned} \mathcal{L}_s &= \frac{1}{N} \sum_{i=1}^{N} H(\mathbf{y}_i, f(\mathbf{y}|\mathbf{x}_i; \theta)) \\ &= \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} -y_{i,k} \log f(\mathbf{y} = k|\mathbf{x}; \theta) \end{aligned} \tag{1}$$

where $f(\mathbf{y}|\mathbf{x}; \theta) \in [0, 1]^K$ is the predicted probabilities produced by the model f with parameter θ for the input $\mathbf{x}$, and $H(\cdot, \cdot)$ is the cross-entropy function.

Different constructions of the unsupervised loss $\mathcal{L}_u$ lead to different SSL methods. Typically, there are two ways of constructing $\mathcal{L}_u$: one is to use pseudo-labels to formulate a "supervised loss" such as the cross-entropy loss (e.g., FixMatch (Sohn et al., 2020)), and another one is to optimize a regularization that does not depend on labels such as consistency regularization (e.g., UDA (Xie et al., 2020)). Next, we will introduce the a recent SSL work to interpret how to generate pseudo-labels and construct unsupervised loss $\mathcal{L}_u$.

3.2. FixMatch: An SSL algorithm with Fixed Thresholding

Due to its simplicity yet empirical success, we select FixMatch (Sohn et al., 2020) as an SSL example in this subsection. Moreover, we consider FixMatch as a warm-up of the proposed algorithm, since FixMatch uses a fixed threshold to select unlabeled examples, it will be used as a comparison with the proposed algorithm.

FixMatch applies weak and strong augmentations to unlabeled examples and generates pseudo-labels using the model's predictions on weakly augmented unlabeled examples. The pseudo-label is only retained if the model produces a high-confidence prediction. The model is then trained to predict the pseudo-label when fed a strongly augmented version of the same example.

Specifically, given a batch of B labeled examples $\{(\mathbf{x}_b^l, \mathbf{y}_b^l) : b \in (1, \cdots, B)\}$ and a batch of μB unlabeled examples $\{\mathbf{x}_b^u : b \in (1, \cdots, \mu B)\}$ where μ determines the relative bath size of labeled and unlabeled data.

For unlabeled data, FixMatch tries to generate pseudo-labels via the model's predictions. FixMatch first predict the class distribution given a weakly augmented version of an unlabeled example

$$\mathbf{q}_b = f(\mathbf{y}|\alpha(\mathbf{x}_b^u); \theta) \tag{2}$$

where $\alpha(\cdot)$ is the weak augmentation. Then, it creates a pseudo-label by

$$\hat{\mathbf{y}}_b^u = \arg\max(\mathbf{q}_b) \tag{3}$$

Following by (Sohn et al., 2020), the argmax applied to a probability distribution produces a one-hot probability distribution. To construct the unsupervised loss, it computes the model prediction for a strong augmentation $\mathcal{A}$ of the same unlabeled examples $\mathbf{x}_b^u$:

$$f(\mathbf{y}|\mathcal{A}(\mathbf{x}_b^u); \theta) \tag{4}$$

The unsupervised loss is defined as the cross-entropy between $\hat{\mathbf{y}}_b^u$ and $f(\mathbf{y}|\mathcal{A}(\mathbf{x}_b^u); \theta)$:

$$H(\hat{\mathbf{y}}_b^u, f(\mathbf{y}|\mathcal{A}(\mathbf{x}_b^u); \theta)) \tag{5}$$

Eventually, FixMatch only uses the unlabeled examples with a high-confidence prediction by selecting based on a fixed threshold $\tau = 0.95$ for all classes. Therefore, in FixMatch, the unsupervised loss with cross-entropy and confidence threshold is defined as:

$$\mathcal{L}_u = \frac{1}{\mu B} \sum_{b=1}^{\mu B} \mathbb{I}(\max(\mathbf{q}_b) \geq \tau) H(\hat{\mathbf{y}}_b^u, f(\mathbf{y}|\mathcal{A}(\mathbf{x}_b^u); \theta)) \tag{6}$$

where $\mathbb{I}(\cdot)$ is an indicator function.

As we discussed in the introduction, with class-imbalanced training data, adopting a fixed threshold for all classes may lead to the elimination of too many unlabeled examples with correct pseudo-labels in the minority classes (See Figure 1), resulting in low recall rates on minority classes and eventually drop off the overall performance. It is natural to think that: **the threshold should be class-dependent and adaptive to class distributions rather than being fixed for all classes.** Therefore, in the next section, we are going to propose a new SSL scheme having adaptive thresholds for different classes.

4. Adsh: An SSL Algorithm with Adaptive Thresholding

We now turn to the framework we propose in this paper: Adsh, an SSL algorithm with thresholds that can be adaptively adjusted for different classes. The detail algorithm procedure is presented in Algorithm 1.

Algorithm 1 Adsh Algorithm.

Input: Labeled Data $\mathcal{D}^l$, unlabeled data $\mathcal{D}^u$, number of classes K, number of epochs E, number of iterations T, unlabeled loss weight λ_u, unlabeled data ratio μ, class bias $\mathbf{s} \in \mathbb{R}^K$, model parameter θ_0.

$t = 0$
for $e = 1$ **to** E **do**
  for $iter = 1$ **to** $\lfloor T/E \rfloor$ **do**
    Sample $\{(\mathbf{x}_b^l, \mathbf{y}_b^l) : b \in (1, \cdots, B)\}$ from $\mathcal{D}^l$.
    Sample $\{\mathbf{x}_b^u : b \in (1, \cdots, \mu B)\}$ from $\mathcal{D}^u$.
    $\mathcal{L}_s = \frac{1}{B} \sum_{b=1}^{B} H(\mathbf{y}_b^l, f(\mathbf{y}|\alpha(\mathbf{x}_b^l); \theta_t))$ *// Compute cross entropy loss for labeled examples*
    for $b = 1$ **to** μB **do**
      $\mathbf{q}_b = f(\mathbf{y}|\alpha(\mathbf{x}_b^u); \theta_t)$ *// Predicted probability distribution*
      $\hat{\mathbf{y}}_b^u = \arg\max(\mathbf{q}_b)$ *// Pseudo-label for* $\mathbf{x}_b^u$
      $H_b = H(\hat{\mathbf{y}}_b^u, f(\mathbf{y}|\mathcal{A}(\mathbf{x}_b^u); \theta)$ *// Compute cross entropy loss for pseudo-labeled examples*
    end for
    for $k = 1$ **to** K **do**
      $\tau_k = \exp(-s_k)$ *// Class-dependent adaptive thresholds*
    end for
    $\mathcal{L}_u = \frac{1}{\mu B} \sum_{b=1}^{\mu B} \mathbb{I}(\max(\mathbf{q}_b) \geq \tau_{\hat{\mathbf{y}}_b^u}) \mathrm{H}_b$
    $\mathcal{L} = \mathcal{L}_s + \lambda_u \mathcal{L}_u$
    $\theta_{t+1} =$ Optimization Step$(\theta_t, \mathcal{L})$ *// Update model parameter via gradient methods, e.g., SGD*
    $t = t + 1$
  end for
  Update s via algorithm 2 or Eq.(11) *// Update* s
end for
return θ_T.

To alleviate the drawbacks of fixed thresholding on the class-imbalanced datasets, we propose to select pseudo-labels via class-dependent thresholds that adaptively change for each class. Specifically, we formulate the SSL objective as an optimization problem that explicitly encodes the number of pseudo-labels to be selected for each class into the objective:

$$\begin{aligned} \min_{\hat{\mathbf{y}}, \mathbf{s}, \theta} \quad & \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} -y_{i,k} \log f(\mathbf{y} = k|\alpha(\mathbf{x}_i^l); \theta) \\ & + \frac{1}{M} \sum_{i=1}^{M} \sum_{k=1}^{K} [-\hat{y}_{i,k} \log f(\mathbf{y} = k|\alpha(\mathbf{x}_i^u); \theta) \\ & - s_k \hat{y}_{i,k}] \\ \text{s.t.} \quad & \hat{\mathbf{y}}_i = [\hat{y}_{i,1}, \cdots, \hat{y}_{i,K}] \in \{0, 1\}^K \\ & 0 \leq \mathbf{1}^\top \hat{\mathbf{y}}_i \leq 1 \\ & s_k > 0, \ \ \forall 1 \leq k \leq K \end{aligned} \tag{7}$$

Algorithm 2 Algorithm for Computing s.

Input: Model parameter θ, unlabeled data $\mathcal{D}^u = \{\mathbf{x}_i^u\}_{i=1}^M$, number of classes K, user-defined threshold for the most majority class τ_1.

```
Initialize an array C with K rows to save model prediction confidence
for x^u in D^u do
    q = f(y|α(x^u); θ)    // Prediction confidence for unlabeled examples
    ŷ = argmax(q)    // Predicted pseudo-label
    C_ŷ ← Append(max(q))    // Save the maximum probability for each example
end for
ρ = 1.0
Sort C_k in descending order ∀1 ≤ k ≤ K
for len ← 1 to length(C_1) do
    if C_1[len] < τ_1 then
        break
    end if
    ρ = len / length(C_1) × 100%    // Percentage of selected pseudo-labels for the most majority class
end for
for k = 1 to K do
    s_k = −log(C_k[length(C_k) * ρ])    // Determine s_k for other classes
end for
return s.
```

where $\hat{\mathbf{y}} \in \mathbb{R}^{M\times K}$ is the pseudo-label matrix for unlabeled examples and $\hat{\mathbf{y}}_i$ is the pseudo-label vector for example $\mathbf{x}_i^u$. $\hat{\mathbf{y}}_i$ is required to be either a discrete one-hot or a zero vector, assigning $\hat{\mathbf{y}}_i$ as $\mathbf{0}$ leads to ignoring this pseudo-label in the model training. s_k introduces different levels of class-wise bias for pseudo-label selection, and a larger s_k indicates a larger number of pseudo-labeled examples would be selected for class k.

Eq.(7) shows that, on one hand, the pseudo-label $\hat{\mathbf{y}}$ should be consistent with the model prediction, on the other hand, the number of selected pseudo-labels is controlled by s_k explicitly for each class k, rather than based on a fixed threshold τ. Similar ideas to control the number of selected examples have also been applied to other machine learning problems, e.g., domain adaptation (Zou et al., 2018), curriculum learning (Zou et al., 2019). Different from these works, our paper pays attention to class imbalanced semi-supervised learning and presents a general scheme for the pseudo-label selection which is an important part of SSL algorithms. This sheds new light on how to apply SSL to more realistic and challenging scenarios.

Eq.(7) can be optimized alternatively: first, solving $\hat{\mathbf{y}}$ and $\mathbf{s}$ given a fixed θ; then, optimizing θ in a supervised manner by leveraging pseudo-labels $\hat{\mathbf{y}}$.

Solving $\hat{\mathbf{y}}$ and $\mathbf{s}$ given a fixed θ. If the model parameter θ is fixed, we have the following theorem to guarantee the solution of $\hat{\mathbf{y}}$.

Theorem 4.1. *Given a learning model $f(\mathbf{x};\theta)$, the pseudo-label $\hat{\mathbf{y}}$ in Eq.(7) has the closed-form solution:*

$$\hat{y}_{i,k} = \begin{cases} 1, & \textbf{\textit{if}}\ k = \operatorname{argmax} \dfrac{f(\mathbf{y}=k|\alpha(\mathbf{x}_i^u);\theta)}{\exp(-s_k)}, \\ & \dfrac{f(\mathbf{y}=k|\alpha(\mathbf{x}_i^u);\theta)}{\exp(-s_k)} \geq 1. \\ 0, & \textbf{\textit{otherwise}}. \end{cases} \tag{8}$$

Theorem 4.1 implies that pseudo-label $\hat{\mathbf{y}}$ is dependent on both model predictions and s_k. Moreover, we can show that under certain conditions, Eq.(8) gives an class-dependent adaptive threshold,

Lemma 4.2. *If $\exp(s_k - s_{k'}) > \frac{f(\mathbf{y}=k'|\alpha(\mathbf{x}_i^u);\theta)}{f(\mathbf{y}=k|\alpha(\mathbf{x}_i^u);\theta)}$ holds for all k and k' that satisfy $f(\mathbf{y}=k|\alpha(\mathbf{x}_i^u);\theta) > f(\mathbf{y}=k'|\alpha(\mathbf{x}_i^u);\theta)$, then we have: $\operatorname{argmax} \frac{f(\mathbf{y}=k|\alpha(\mathbf{x}_i^u);\theta)}{\exp(-s_k)} = \operatorname{argmax} f(\mathbf{y}=k|\alpha(\mathbf{x}_i^u);\theta)$.*

It is noteworthy that the condition in above lemma is easy to satisfy since the model prone to over-confident (Thulasidasan et al., 2019), thus, $f(\mathbf{y}=k'|\alpha(\mathbf{x}_i^u);\theta)/f(\mathbf{y}=k|\alpha(\mathbf{x}_i^u);\theta)$ is relatively small in real tasks.

The above analysis shows that instead of selecting pseudo-labels based on the original prediction confidence and a fixed threshold τ, we select pseudo-label for unlabeled example $\mathbf{x}_i^u$ that are predicted as $\hat{\mathbf{y}}_i^u$ with

$$\mathbb{I}(\max(\mathbf{q}_i) \geq \exp(-s_{\hat{\mathbf{y}}_i^u})) \tag{9}$$

where $\mathbf{q}_i = f(\mathbf{y}|\alpha(\mathbf{x}_i^u);\theta)$ and $\hat{\mathbf{y}}_i^u = \operatorname{argmax}(\mathbf{q}_i)$.

If the ground-truth class distribution of unlabeled data is known, we can solve s_k to make the pseudo-label $\hat{\mathbf{y}}$ has the same class distribution with the ground-truth y^*, i.e.,

$$\frac{\sum_{i=1}^M \hat{y}_{i,k}}{\sum_{i=1}^M \hat{y}_{i,k'}} = \frac{\sum_{i=1}^M y_{i,k}^*}{\sum_{i=1}^M y_{i,k'}^*}, \quad \forall k,k' \in \{1,\cdots,K\} \tag{10}$$

Specifically, we can first set s_1 using a user-defined hyper-parameter, e.g., $s_1 = -\log(0.95)$, then, s_k for $2 \leq k \leq K$ can be computed by:

$$\begin{aligned} & \sum_{i=1}^M \mathbb{I}(f(\mathbf{y}=k|\alpha(\mathbf{x}_i^u);\theta) \geq \exp(-s_k)) \\ = & \frac{\sum_{i=1}^M \mathbb{I}(f(\mathbf{y}=1|\alpha(\mathbf{x}_i^u);\theta) \geq \exp(-s_1))}{\gamma_k} \end{aligned} \tag{11}$$

where $\gamma_k = \frac{\sum_{i=1}^M y_{i,1}^*}{\sum_{i=1}^M y_{i,k}^*}$ indicates the imbalance ratio between class 1 and class k.

Table 1. Comparison of classification performance (Accuracy (%)) on imbalanced CIFAR-10 dataset under three different imbalance ratio: $\gamma = 50, 100, 150$ and two different numbers of labeled data: $N_1 = 1500, M_1 = 3000$ and $N_1 = 500, M_1 = 4000$. The best results are indicated in bold.

Imbalanced CIFAR-10 Dataset						
	$N_1 = 1500, M_1 = 3000$			$N_1 = 500, M_1 = 4000$		
Algorithm	$\gamma = 50$	$\gamma = 100$	$\gamma = 150$	$\gamma = 50$	$\gamma = 100$	$\gamma = 150$
Supervised	65.23 ± 0.05	58.94 ± 0.13	55.63 ± 0.38	51.31 ± 0.34	45.82 ± 0.41	40.90 ± 0.39
CBL	65.52 ± 0.31	58.52 ± 0.45	52.36 ± 0.58	51.94 ± 0.71	46.22 ± 0.92	41.58 ± 1.24
Re-Sampling	64.53 ± 0.39	56.34 ± 0.42	53.21 ± 0.51	51.96 ± 0.65	48.13 ± 1.25	40.26 ± 1.88
cRT	67.82 ± 0.14	63.43 ± 0.45	59.56 ± 0.44	56.28 ± 1.45	48.11 ± 0.79	45.02 ± 1.08
LDAM	68.91 ± 0.10	63.15 ± 0.24	58.68 ± 0.30	56.41 ± 0.92	49.27 ± 0.88	45.10 ± 0.75
Mean-Teacher	68.84 ± 0.82	61.33 ± 0.28	54.79 ± 0.31	56.34 ± 1.68	48.55 ± 0.77	45.32 ± 1.20
MixMatch	73.59 ± 0.46	65.03 ± 0.26	62.71 ± 0.29	65.32 ± 1.20	56.41 ± 1.96	52.38 ± 1.88
ReMixMatch	78.96 ± 0.29	72.88 ± 0.12	68.61 ± 0.40	76.83 ± 0.98	70.12 ± 1.23	59.58 ± 1.30
FixMatch	79.10 ± 0.14	71.50 ± 0.31	68.47 ± 0.15	77.34 ± 0.96	68.45 ± 0.94	60.10 ± 0.82
DARP	81.60 ± 0.31	75.23 ± 0.14	69.31 ± 0.26	76.72 ± 0.46	69.41 ± 0.50	61.23 ± 0.31
CReST	82.03 ± 0.26	75.08 ± 0.41	69.84 ± 0.39	76.18 ± 0.36	69.50 ± 0.70	60.81 ± 0.55
Adsh	$\mathbf{83.38 \pm 0.06}$	$\mathbf{76.52 \pm 0.35}$	$\mathbf{71.49 \pm 0.30}$	$\mathbf{79.27 \pm 0.38}$	$\mathbf{70.97 \pm 0.46}$	$\mathbf{62.04 \pm 0.51}$

However, in many realistic scenarios, the class distribution is unknown, in this case we present a simple and effective alternative strategy to determine s without introducing additional hyper-parameters. The full procedure is presented in algorithm 2. Specifically, the algorithm to determine s_k exploits the class-wise confidence threshold effectively by ranking all the probabilities predicted as class k in descending order and setting s_k such that $\exp(-s_k)$ be equal to the predicted probability ranked at $\rho \times length(C_k)$, where $length(C_k)$ is the number of unlabeled examples predicted as class k and $\rho \times 100\%$ denotes the percentage of selected confident pseudo-labels. Such a strategy takes the predicted probability ranked at $\rho \times 100\%$ separately from each class as a reference for thresholding.

The proportion ρ is computed from the majority class, i.e., the class 1. Specifically, we set s_1 using a user-defined hyper-parameter (e.g., $\tau_1 = 0.95$), then the proportion of pseudo-labels selected for class 1 would be determined as:

$$\rho = \frac{\sum_{i=1}^{M} \mathbb{I}(f(\mathbf{y} = 1|\alpha(\mathbf{x}_i^u); \theta) \geq \tau_1)}{length(C_1)}$$

where $length(C_1) = \sum_{i=1}^{M} \mathbb{I}(\text{argmax}(f(\mathbf{y}|\alpha(\mathbf{x}_i^u); \theta)) = 1)$. This ensures pseudo-labels with the same confidence-level with-in class can be selected for every class.

Solving θ given fixed $\hat{\mathbf{y}}$ and s. With the pseudo-label $\hat{\mathbf{y}}$, we can solve θ in the supervised manner. Same as previous SSL methods, we optimize θ using the SGD algorithm, in which the unsupervised loss $\mathcal{L}_u$ is given by

$$\mathcal{L}_u = \frac{1}{\mu B} \sum_{b=1}^{\mu B} \mathbb{I}(\max(\mathbf{q}_b) \geq \tau_{\hat{\mathbf{y}}_b^u}) H(\hat{\mathbf{y}}_b^u, f(\mathbf{y}|\mathcal{A}(\mathbf{x}_b^u); \theta)) \tag{12}$$

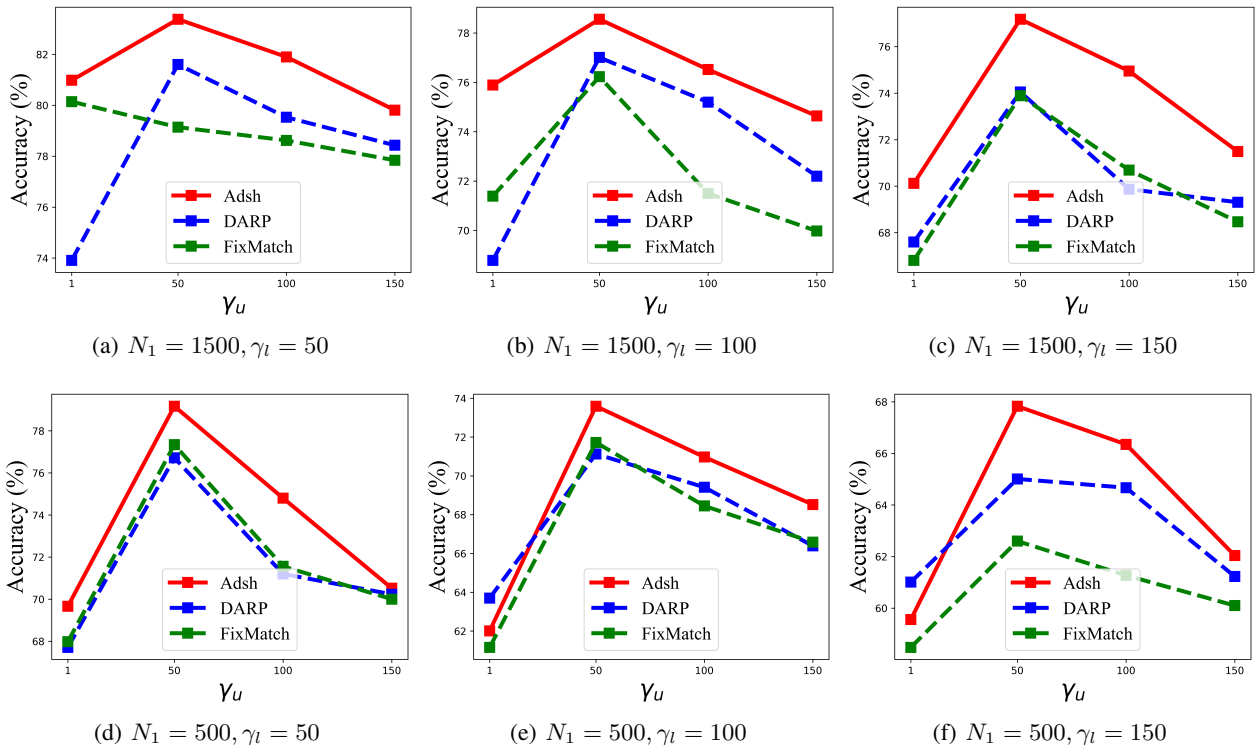
The above unsupervised loss function implies that the pseudo-label selection is not dependent on a fixed threshold. Instead, it is dependent on a threshold that adaptively changes for different classes. Selecting the pseudo-labels by utilizing the adaptive thresholding gives the advantage of selecting examples that have relatively low confidence, but high within-class confidence and thus help alleviate the bias problem of the original prediction under class-imbalanced distribution.

5. Experiments

In this section, we give comprehensive evaluations on various class-imbalanced SSL scenarios. We first describe the experimental setups in Section 5.1. Then, we present empirical results of our proposal and other compared methods under extensive setups in Section 5.2. Finally, we present detailed analyses to help understand the superiority of our proposal in Section 5.3.

5.1. Experimental setup

Imbalanced Datasets. We conduct experiments on long-tailed variants of CIFAR-10 (Krizhevsky & Hinton, 2009), SVHN (Netzer et al., 2011) and STL-10 (Coates et al., 2011) datasets with various levels of class imbalance and different ratios of labeled data. These are all widely adopted datasets to evaluate SSL algorithms. For constructing the class-imbalanced training dataset, we use two parameters γ_l, γ_u to denote the *imbalance ratio* of labeled and unlabeled data, i.e., $\gamma_l = \frac{N_1}{N_K}, \gamma_u = \frac{M_1}{M_K}$. Once γ_l, γ_u and N_1, M_1 are given, we set $N_k = N_1 \cdot \gamma_l^{-\frac{k-1}{K-1}}$ and $M_k = M_1 \cdot \gamma_u^{-\frac{k-1}{K-1}}$ for $1 < k \leq K$. Specifically, we consider two different num-



(a) $N_1 = 1500, \gamma_l = 50$ (b) $N_1 = 1500, \gamma_l = 100$ (c) $N_1 = 1500, \gamma_l = 150$

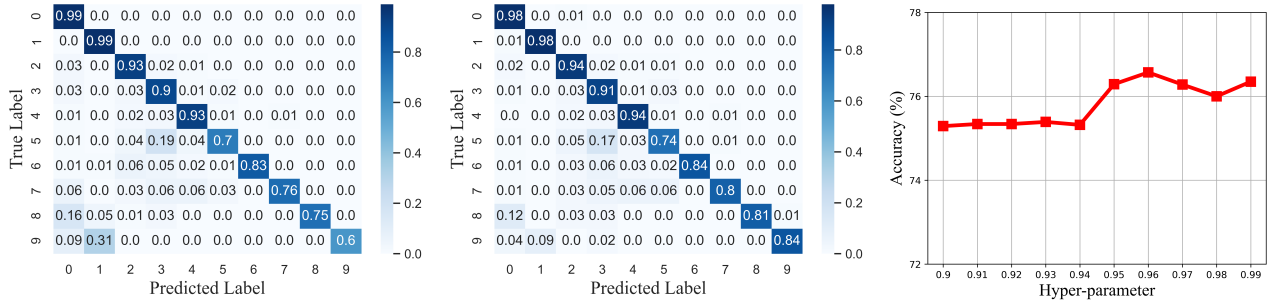
(d) $N_1 = 500, \gamma_l = 50$ (e) $N_1 = 500, \gamma_l = 100$ (f) $N_1 = 500, \gamma_l = 150$

Figure 2. Comparison results of classification performance on CIFAR-10 with 12 different imbalance ratios, i.e., $\gamma_l \in [50, 100, 150]$, $\gamma_u \in [1, 50, 100, 150]$ and 2 different number of labeled examples, i.e., $N_1 = 1500$ (upper), $N_1 = 500$ (lower).

bers of labeled examples, i.e., $N_1 = 500, M_1 = 4000$ and $N_1 = 1500, M_1 = 3000$, and various imbalance ratios, i.e., γ_l and γ_u come from combinations of $[1, 50, 100, 150]$. The test set remains untouched and balanced, so that accuracy is adopted as the evaluation criterion.

Compared Methods. We compare our Adsh with many methods, including class-imbalanced learning methods, SSL methods, and recently proposed class-imbalanced SSL methods. Specifically, for class-imbalanced learning, we consider a wide range of methods including a) Class-Balanced Loss (CBL) (Cui et al., 2019), a representative re-weighting strategy where labeled examples are re-weighted according to the inverse of the effective number of examples in each class; b) Re-Sampling (Byrd & Lipton, 2019), a typical re-sampling strategy where each labeled example is sampled with probability proportional to the inverse samples of its class; c) classifier Re-Training (cRT) (Kang et al., 2020), which retrains the classifier with a balancing objective after training the whole network to learn a representation under imbalanced distribution; d) Label-Distribution-Aware Margin (LDAM), which imposes a larger margin to minority class in the training process and balancing the objective at the later stage of training. We also evaluate several classic SSL algorithms including a) Mean-Teacher (Tarvainen & Valpola, 2017), which adds a consistency regularization between the prediction of the current model and the ensemble of the model in previous training epochs; b) MixMatch (Berthelot et al., 2019), a holistic SSL method that adopted both pseudo-label and consistency regularization strategies with Mixup augmentations; c) ReMixMatch (Berthelot et al., 2020), which further improves MixMatch by adding an augmentation anchoring and a distribution alignment; d) FixMatch (Sohn et al., 2020), reported as the best performing SSL method, that generate pseudo-labels from the weakly augmented data and applied to strongly augmented data. To further show the efficacy of our proposal, we also compared with recently proposed algorithms that consider SSL and class-imbalance simultaneously, including a) Distribution Aligning Refinery of Pseudo-label (DARP) (Kim et al., 2020), which refines the pseudo-labels generated from the SSL model to match the ground-truth class distribution of unlabeled data; b) Class-Rebalancing Self-Training (CReST) (Wei et al., 2021), a self-training based strategy that selects pseudo-labels according to the inverse of label frequency and align distribution progressively.

Implementation Details. In all experiments, we adopt the Wide ResNet-28-2 (Zagoruyko & Komodakis, 2016) as the backbone since it is commonly adopted in various SSL methods (Oliver et al., 2018). We train the model with batch size



(a) Confusion Matrix of FixMatch (b) Confusion Matrix of Adsh (c) Hyper-Parameter Sensitivity

Figure 3. Detailed analyses of the Adsh. (a) and (b): Confusion matrix on unlabeled data produced by FixMatch (left) and Adsh(right); (c): Performance robustness with hyper-parameter τ_1 changes.

64 for 2^{18} iterations. We adopt Adam (Kingma & Ba, 2015) optimizer with a learning rate 2×10^{-3}. Following (Sohn et al., 2020) and (Kim et al., 2020), the exponential moving average (EMA) technique is applied with a decay rate 0.999. For all algorithms, we evaluate the model on the test dataset every 512 iterations and record the average test accuracy of the last 20 evaluations, following (Kim et al., 2020). Mean $\pm$ std accuracy over five random runs is reported. More details on the implementation are presented in the supplementary material.

5.2. Empirical Results

We first evaluate Adsh with compared methods on the CIFAR-10 dataset under various levels of imbalance ratio and different numbers of labeled examples. In particular, we study two situations: $\gamma_l = \gamma_u$ and $\gamma_l \neq \gamma_u$.

Results on CIFAR-10 with $\gamma_l = \gamma_u$. We first conduct experiments in the case that $\gamma := \gamma_l = \gamma_u$, the most natural scenario that labeled and unlabeled data have the same distribution. Table 1 summarizes the performance of our Adsh and compared methods. From the results, we observe that in most cases SSL methods perform better than class-imbalanced learning methods since they use more unlabeled training data. DARP and CReST methods achieve good performance among compared methods since they consider both unlabeled data exploitation and imbalanced distribution. It is noticeable that our proposal Adsh consistently achieves the best performance in all settings with various imbalance ratios and different numbers of labeled examples.

Results on CIFAR-10 with $\gamma_l \neq \gamma_u$. $\gamma_l \neq \gamma_u$ brings new challenges since the distribution between labeled and unlabeled data is mismatched. We conduct experiments on 24 settings with different imbalance ratios γ_l, γ_u, and different numbers of labeled examples. We report the results of competitive methods FixMatch, DARP, and Adsh. The CReST is omitted since it can not be applied to the mismatched distribution. The results are summarized in Figure 2. An interesting observation is that for a fixed γ_l, all three methods suffer performance degradation when $\gamma_u = 1$, even this is the most balanced unlabeled dataset. One possible reason is that the extent of distribution mismatch prevents performance improvement. The results in Figure 2 show that our Adsh performs better than DARP and FixMatch methods on almost all settings while DARP performs even worse than the FixMatch algorithm in some cases.

Table 2. Comparison of classification performance (Accuracy (%)) on imbalanced SVHN dataset with $\gamma = \gamma_l = \gamma_u = 100$, and STL-10 datasets with $\gamma_l = 10$ or 20 and unknown γ_u. The best results are indicated in bold.

	SVHN	STL-10	
Algorithm	$\gamma = 100$	$\gamma_l = 10$	$\gamma_l = 20$
ReMixMatch	88.91 ± 0.32	67.43 ± 0.43	60.82 ± 0.93
FixMatch	89.34 ± 0.20	73.25 ± 0.21	63.54 ± 0.21
DARP	90.15 ± 0.46	76.97 ± 0.45	68.87 ± 0.66
CReST	89.90 ± 0.64	76.30 ± 0.38	69.43 ± 0.89
Adsh	$\mathbf{92.13 \pm 0.39}$	$\mathbf{79.25 \pm 0.41}$	$\mathbf{71.03 \pm 0.20}$

Results on SVHN and STL-10. We also present experimental results on SVHN and STL-10 datasets. In the case of SVHN, we construct imbalanced dataset as done in Section 5.1 in which 20% are labeled and $\gamma_l = \gamma_u = 100$. For STL-10, we construct a long-tailed variants labeled dataset with $N_1 = 450$ and $\gamma_l = \{10, 20\}$. We fully use the unlabeled data in STL-10 with $M = 100,000$, whose class distribution is imbalanced but the imbalance ratio γ_u is unknown. Therefore, in the case of STL-10, the labeled and unlabeled dataset may not have the same class distribution, i.e., $\gamma_l \neq \gamma_u$. Table 2 summarizes the learning performance on SVHN and STL-10 datasets. Since the simple class-imbalanced learning methods perform significantly worse than SOTA SSL methods and class-imbalanced SSL methods, we omit the results of these methods. From the results, we can see that our proposal consistently improves the performance on both SVHN and STL-10 datasets.

5.3. Detailed Analyses

Quality of pseudo-labels. We evaluate Adsh by measuring the confusion matrix on unlabeled data to show that our Adsh can improve the quality of pseudo-labels. Figure 3(a) and Figure 3(b) visualize the confusion matrix of pseudo-labels using the model trained on CIFAR-10 with $\gamma_l = \gamma_u = 100, N_1 = 1500, M_1 = 3000$. The results show that the raw pseudo-labels generated by FixMatch are biased towards majority classes, for example, there are more than 30% examples that belong to class 9 are predicted wrongly as class 1. On the contrary, our proposal can achieve a more unbiased confusion matrix. These results indicate that the quality of pseudo-labels is actually improved, which can help to improve the generalization performance.

Hyper-Parameter Sensitivity. We also study the performance sensitivity of Adsh to different values of hyper-parameter τ_1. The results of model trained on CIFAR-10 dataset with $N_1 = 1500, M_1 = 3000, \gamma_l = \gamma_u = 100$ are presented in Figure 3(c). When τ_1 is set as 0.96, the model achieves the best performance while changing it to others did not hurt much. These results show that our proposal Adsh is robust to the hyper-parameter selection. Based on the results, to use the Adsh approach, we suggest setting τ_1 as 0.96 first, and further optimize it from $\{0.95, 0.96, 0.97, 0.98, 0.99\}$.

6. Conclusions

In this paper, we tackle an important problem of SSL, that is, SSL in the presence of class imbalanced distribution. We propose a novel Adsh approach that adaptively selects pseudo-labels to train models based on a class-dependent threshold. We formulate the pseudo-label selection into an optimization objective by explicitly considering the number of pseudo-labels to be selected for each class and derive a highly efficient closed-form solution. The proposed Adsh method is a generic scheme that can be easily integrated with existing SSL methods. We demonstrate the use of adaptively class-dependent thresholding can help to the performance of the SOTA SSL method FixMatch in extensive experiments, indicating the importance of adaptive threshold in class-imbalanced SSL.

How to construct robust SSL models in realistic scenarios has attracted great attention in recent years. Class-imbalanced SSL is a representative problem that brings robustness threats to SSL while is still understudied. Our work puts a promising scheme in this direction. One limitation of our scheme is it does not have theoretical guarantees. We will put efforts into this direction in future work, such as giving convergence analysis of SSL algorithms that use fixed thresholds and adaptive thresholds. The code of this paper has been released on `http://www.lamda.nju.edu.cn/code_ADSH.ashx`.

Acknowledgements

This research was supported by the National Science Foundation of China (62176118, 61921006), and the Huawei Cooperation Fund.

References

Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., and Raffel, C. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*, pp. 5050–5060, 2019.

Berthelot, D., Carlini, N., Cubuk, E. D., Kurakin, A., Sohn, K., Zhang, H., and Raffel, C. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. In *Proceedings of the 8th International Conference on Learning Representations*, 2020.

Buda, M., Maki, A., and Mazurowski, M. A. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249–259, 2018.

Byrd, J. and Lipton, Z. What is the effect of importance weighting in deep learning? In *Proceedings of the 36th International Conference on Machine Learning*, pp. 872–881, 2019.

Cao, K., Wei, C., Gaidon, A., Aréchiga, N., and Ma, T. Learning imbalanced datasets with label-distribution-aware margin loss. In *Advances in Neural Information Processing Systems*, pp. 1565–1576, 2019.

Chapelle, O., Scholkopf, B., and Zien, A. *Semi-supervised learning*. MIT Press, 2006.

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

Coates, A., Ng, A. Y., and Lee, H. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, pp. 215–223, 2011.

Cui, Y., Jia, M., Lin, T.-Y., Song, Y., and Belongie, S. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9268–9277, 2019.

Dong, Q., Gong, S., and Zhu, X. Imbalanced deep learning by minority class incremental rectification. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 41 (6):1367–1381, 2019.

Grandvalet, Y. and Bengio, Y. Semi-supervised learning by entropy minimization. In *Advances in Neural Information Processing Systems*, pp. 529–536, 2005.

Guo, L.-Z., Zhang, Z.-Y., Jiang, Y., Li, Y.-F., and Zhou, Z.-H. Safe deep semi-supervised learning for unseen-class unlabeled data. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 3897–3906, 2020a.

Guo, L. Z., Zhou, Z., and Li, Y. F. RECORD: resource constrained semi-supervised learning under distribution shift. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1636–1644, 2020b.

Guo, L. Z., Zhou, Z., Shao, J. J., Zhang, Q., Kuang, F., Li, G. L., Liu, Z. X., Wu, G., Ma, N., Li, Q., and Li, Y. F. Learning from imbalanced and incomplete supervision with its application to ride-sharing liability judgment. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 487–495, 2021.

Guo, L. Z., Zhou, Z., and Li, Y. F. Robust deep semi-supervised learning: A brief introduction. *CoRR*, abs/2202.05975, 2022.

Hariharan, B. and Girshick, R. Low-shot visual recognition by shrinking and hallucinating features. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3018–3027, 2017.

He, H. and Garcia, E. A. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, 2009.

Hu, Z., Tan, B., Salakhutdinov, R., Mitchell, T. M., and Xing, E. P. Learning data manipulation for augmentation and weighting. In *Advances in Neural Information Processing Systems*, pp. 15738–15749, 2019.

Huang, C., Li, Y., Loy, C. C., and Tang, X. Deep imbalanced learning for face recognition and attribute prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(11):2781–2794, 2019.

Jeong, J., Lee, S., Kim, J., and Kwak, N. Consistency-based semi-supervised learning for object detection. In *Advances in Neural Information Processing Systems*, pp. 10759–10768, 2019.

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

Kang, B., Xie, S., Rohrbach, M., Yan, Z., Gordo, A., Feng, J., and Kalantidis, Y. Decoupling representation and classifier for long-tailed recognition. In *Proceedings of the 8th International Conference on Learning Representations*, 2020.

Khan, S., Hayat, M., Zamir, S. W., Shen, J., and Shao, L. Striking the right balance with uncertainty. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 103–112, 2019.

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

Kim, J., Hur, Y., Park, S., Yang, E., Hwang, S. J., and Shin, J. Distribution aligning refinery of pseudo-label for imbalanced semi-supervised learning. In *Advances in Neural Information Processing Systems*, pp. 14567–14579, 2020.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations*, 2015.

Krizhevsky, A. and Hinton, G. Learning multiple layers of features from tiny images. *Technical Report*, 2009.

Laine, S. and Aila, T. Temporal ensembling for semi-supervised learning. In *Proceedings of the 5th International Conference on Learning Representations*, 2017.

LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *Nature*, 521(7553):436–444, 2015.

Lee, D.-H. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *ICML Workshop on Challenges in Representation Learning*, pp. 2–8, 2013.

Li, Y.-F., Guo, L.-Z., and Zhou, Z.-H. Towards safe weakly supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(1):334–346, 2019.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2980–2988, 2017.

Liu, Z., Miao, Z., Zhan, X., Wang, J., Gong, B., and Yu, S. X. Large-scale long-tailed recognition in an open world. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2537–2546, 2019.

Miyato, T., Dai, A. M., and Goodfellow, I. J. Adversarial training methods for semi-supervised text classification. In *Proceedings of the 5th International Conference on Learning Representations*, 2017.

Miyato, T., Maeda, S.-i., Koyama, M., and 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.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. Reading digits in natural images with unsupervised feature learning. In *Proceedings of the NeurIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

Oliver, A., Odena, A., Raffel, C. A., Cubuk, E. D., and Goodfellow, I. Realistic evaluation of deep semi-supervised learning algorithms. In *Advances in Neural Information Processing Systems*, pp. 3235–3246, 2018.

Ren, M., Zeng, W., Yang, B., and Urtasun, R. Learning to reweight examples for robust deep learning. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 4331–4340, 2018.

Sohn, K., Berthelot, D., Carlini, N., Zhang, Z., Zhang, H., Raffel, C., Cubuk, E. D., Kurakin, A., and Li, C. L. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *Advances in Neural Information Processing Systems*, pp. 596–608, 2020.

Souly, N., Spampinato, C., and Shah, M. Semi supervised semantic segmentation using generative adversarial network. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5688–5696, 2017.

Tarvainen, A. and Valpola, H. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems*, pp. 1195–1204, 2017.

Thulasidasan, S., Chennupati, G., Bilmes, J. A., Bhattacharya, T., and Michalak, S. On mixup training: Improved calibration and predictive uncertainty for deep neural networks. In *Advances in Neural Information Processing Systems*, pp. 13888–13899, 2019.

Wang, Y.-X., Ramanan, D., and Hebert, M. Learning to model the tail. In *Advances in Neural Information Processing Systems*, pp. 7032–7042, 2017.

Wei, C., Sohn, K., Mellina, C., Yuille, A. L., and Yang, F. Crest: A class-rebalancing self-training framework for imbalanced semi-supervised learning. In *Proceedings of the IEEE International Conference on Computer Vision*, 2021.

Xie, Q., Dai, Z., Hovy, E. H., Luong, T., and Le, Q. Unsupervised data augmentation for consistency training. In *Advances in Neural Information Processing Systems*, pp. 6256–6268, 2020.

Xu, Y., Shang, L., Ye, J., Qian, Q., Li, Y.-F., Sun, B., Li, H., and Jin, R. Dash: Semi-supervised learning with dynamic thresholding. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 11525–11536, 2021.

Yin, X., Yu, X., Sohn, K., Liu, X., and Chandraker, M. Feature transfer learning for face recognition with under-represented data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5704–5713, 2019.

Zagoruyko, S. and Komodakis, N. Wide residual networks. In *Proceedings of the British Machine Vision Conference*, 2016.

Zhou, Z., Guo, L. Z., Cheng, Z., Li, Y. F., and Pu, S. STEP: out-of-distribution detection in the presence of limited in-distribution labeled data. In *Advances in Neural Information Processing Systems*, pp. 29168–29180, 2021.

Zhou, Z.-H. A brief introduction to weakly supervised learning. *National Science Review*, 5(1):44–53, 2017.

Zou, Y., Yu, Z., Kumar, B. V. K. V., and Wang, J. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *Proceedings of the European Conference on Computer Vision*, pp. 297–313, 2018.

Zou, Y., Yu, Z., Liu, X., Kumar, B. V. K. V., and Wang, J. Confidence regularized self-training. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5981–5990, 2019.

A. Theorem Proof

Theorem A.1. *The objective formulation*

$$\begin{aligned} &\textit{minimize}_{\hat{\mathbf{y}}} && \mathcal{L}_u = \frac{1}{M}\sum_{i=1}^{M}\sum_{k=1}^{K}[-\hat{y}_{i,k}\log(f(\mathbf{y}=k|\alpha(\mathbf{x}_i^u);\theta)) - s_k\hat{y}_{i,k}] \\ &\textit{subject to} && \hat{\mathbf{y}}_i = [\hat{y}_{i,1},\cdots,\hat{y}_{i,K}] \in \{0,1\}^K, \;\; 0 \le \boldsymbol{I}^\top\hat{\mathbf{y}}_i \le 1 \\ & && s_k > 0, \;\; \forall 1 \le k \le K \end{aligned} \tag{13}$$

has the closed-form solution:

$$\hat{y}_{i,k} = \begin{cases} 1, & \textbf{\textit{if}}\ k = \operatorname{argmax} \dfrac{f(\mathbf{y}=k|\alpha(\mathbf{x}_i^u);\theta)}{\exp(-s_k)}, \\ & \dfrac{f(\mathbf{y}=k|\alpha(\mathbf{x}_i^u);\theta)}{\exp(-s_k)} \ge 1. \\ 0, & \textbf{\textit{otherwise}}. \end{cases} \tag{14}$$

Proof. To select pseudo-label $\hat{y}_{i,k} = 1$ for $\mathbf{x}_i^u$, two conditions need to be satisfied, first

$$-\log(f(\mathbf{y}=k|\alpha(\mathbf{x}_i^u);\theta)) - s_k < -\log(f(\mathbf{y}=k'|\alpha(\mathbf{x}_i^u);\theta)) - s_{k'}$$

From the above inequality, we can derive that

$$\frac{f(\mathbf{y}=k|\alpha(\mathbf{x}_i^u);\theta)}{\exp(-s_k)} > \frac{f(\mathbf{y}=k'|\alpha(\mathbf{x}_i^u);\theta))}{\exp(-s_{k'})}$$

for all other class k'.

Then, the second condition is,

$$-\log(f(\mathbf{y}=k|\alpha(\mathbf{x}_i^u);\theta)) - s_k \le 0$$

and we can obtain that,

$$\frac{f(\mathbf{y}=k|\alpha(\mathbf{x}_i^u);\theta))}{\exp(-s_k)} \ge 1$$

Therefore, the closed-formed solution for our objective function is Eq.(14). □

B. Implementation Details

In all experiments, we adopt the Wide ResNet-28-2 as the backbone. We train the model with batch size 64 for 2^{18} training iterations. For training with semi-supervised learning algorithms, we adopt Adam optimizer with a learning rate of 2×10^{-3}. For the hyper-parameters of Adam, we use $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$ which is the default choice. The exponential moving average (EMA) technique is applied with a decay rate of 0.999. For training with re-balancing algorithms, we use SGD with a learning rate of 0.1, momentum 0.9, and weight decay 5×10^{-4}. The learning rate of SGD decays by 0.01 at the time step 80% and 90% iterations. For all algorithms, we evaluate the model on the test dataset every 512 iterations and record the average test accuracy of the last 20 evaluations. Mean $\pm$ std accuracy over five random runs is reported. All experiments are conducted on Tesla V100 GPUs.

For Mean-Teacher, the consistency coefficient λ_u is set to 50 and the EMA model used for the evaluation is reused for the consistency regularization. We ramped up the consistency coefficient starting from 0 to λ_u using a sigmoid schedule so that it achieves the maximum value at 1.0×10^5 iterations. For MixMatch, we set temperature T as 0.5, the number of augmentation K as 2, the parameter for beta distribution α as 0.75, and the consistency coefficient λ_u as 75. The consistency coefficient is linearly increased to λ_u started from 0. For ReMixMatch, we set $K = 2$ for the number of augmentations to balance the improvement from an augmentation anchoring and a computational cost, suggested by (Kim et al., 2020). We use RandAugment as a strong augmentation. Other hyper-parameters are as same as the original paper. For FixMatch, we

use $\mu = 2$ to determine the ratio of unlabeled data and set $\lambda_u = 1$, $\tau = 0.95$ as the original paper. For DARP, we adopt the the official code and recommended parameters[1]. For cReST, we set the hyper-parameter as the original paper. FixMatch is adopted as the backbone SSL algorithm for DARP and cReST. For our Adsh we set $\tau_1 = 0.95$ as FixMatch and update s every 512 iterations.

C. Combination of Class-Imbalanced Learning and SSL

We also conduct experiments by combining the class-imbalanced learning method and SSL methods. Specifically, we examine Adsh and FixMatch by combining with the classifier re-training (cRT) algorithm (Kang et al., 2020), which is a recently introduced state-of-the-art re-balancing algorithm for the class-imbalanced dataset. These algorithms are denoted by "FixMatch + cRT" and "Adsh+ cRT", respectively. Table 3 summarized the performance of FixMatch and Adsh with/without cRT. From the results, we can observe that combining with cRT can further the performance of Adsh. Moreover, with cRT, our proposal Adsh still achieves better performance than FixMatch.

Table 3. Comparison of classification performance (Accuracy (%)) on imbalanced CIFAR-10 dataset under three different class-imbalance ratio $\gamma = \gamma_l = \gamma_u$. The best results are indicated in bold.

	Imbalanced CIFAR-10		
Algorithm	$\gamma = 50$	$\gamma_l = 100$	$\gamma_l = 150$
Supervised	65.23 ± 0.05	58.94 ± 0.13	55.63 ± 0.38
cRT	67.82 ± 0.14	63.43 ± 0.45	59.56 ± 0.44
FixMatch	79.10 ± 0.14	71.50 ± 0.31	68.47 ± 0.15
Adsh	83.38 ± 0.06	76.52 ± 0.35	71.49 ± 0.30
FixMatch + cRT	84.32 ± 0.40	78.39 ± 0.45	73.26 ± 0.23
Adsh + cRT	$\mathbf{86.21 \pm 0.24}$	$\mathbf{79.82 \pm 0.24}$	$\mathbf{75.48 \pm 0.31}$

[1] https://github.com/bbuing9/DARP