Smoothed Adaptive Weighting for Imbalanced Semi-Supervised Learning: Improve Reliability Against Unknown Distribution Data

Zhengfeng Lai [*1] **Chao Wang** [*2] **Henrry Gunawan** [1] **Sen-Ching Cheung** [3] **Chen-Nee Chuah** [1]

Abstract

Despite recent promising results on semi-supervised learning (SSL), data imbalance, particularly in the unlabeled dataset, could significantly impact the training performance of a SSL algorithm if there is a mismatch between the expected and actual class distributions. The efforts on how to construct a robust SSL framework that can effectively learn from datasets with unknown distributions remain limited. We first investigate the feasibility of adding weights to the consistency loss and then we verify the necessity of smoothed weighting schemes. Based on this study, we propose a self-adaptive algorithm, named Smoothed Adaptive Weighting (SAW). SAW is designed to enhance the robustness of SSL by estimating the learning difficulty of each class and synthesizing the weights in the consistency loss based on such estimation. We show that SAW can complement recent consistency-based SSL algorithms and improve their reliability on various datasets including three standard datasets and one gigapixel medical imaging application without making any assumptions about the distribution of the unlabeled set.

1. Introduction

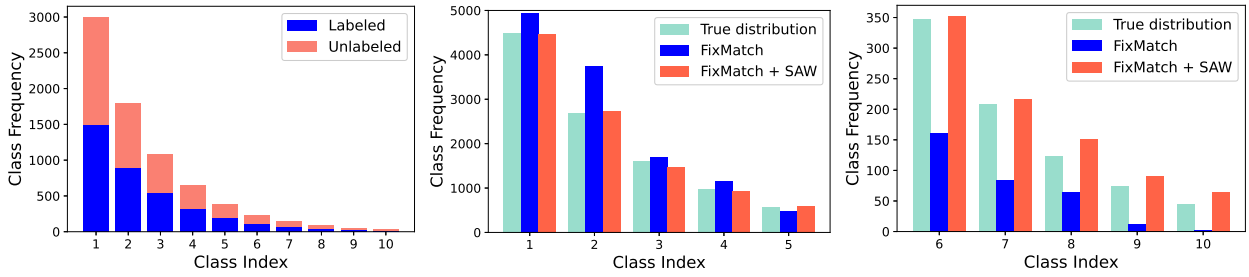
Although deep neural networks have repeatedly achieved promising performance on various tasks (He et al., 2016; Chen et al., 2019; Chan et al., 2019), their success heavily relies on the accessibility of large annotated datasets that typically involve labor-intensive labeling. The labeling cost could be even higher in medical imaging applications where images are at gigapixel level (Lai et al., 2021) and specific domain knowledge is required (Chan et al., 2019). Semi-supervised learning (SSL) is one of the promising methods to reduce the labeling cost by leveraging unlabeled data.

To leverage unlabeled data effectively, recent state-of-the-art SSL algorithms (Arazo et al., 2020; Sohn et al., 2020; Berthelot et al., 2020) produce pseudo labels on unlabeled data, and then train the model by iteratively optimizing consistency loss as a regularization via pseudo labels. These SSL methods typically assume that labeled and unlabeled data are balanced (i.e., equal representation from different classes). One challenge for the wide-adoption of SSL is that the performance may suffer from imbalanced training set (Kim et al., 2020; Calderon-Ramirez et al., 2020; Pulido et al., 2020), such as a long-tailed dataset from CIFAR-10 shown in Figure 1(a) (Krizhevsky, 2009). Specifically, the model's predictions can be biased towards the *majority classes* and away from *minority classes* that have few samples (Dong et al., 2018). This affects the quality of pseudo labels, which could subsequently make the model even more biased with self-reinforcing errors (Tai et al., 2021). Consider the example shown in Figure 1: using FixMatch (Sohn et al., 2020) on the long-tailed CIFAR-10 dataset, Figure 1(c) shows that the numbers of pseudo labels for minority classes $\{6, 7, 8, 9, 10\}$ are significantly smaller than the true distribution. In fact, the smallest class receives **no** pseudo labels at all. Figure 1(b) shows that original SSL model generates more pseudo labels for majority classes. The bias in the pseudo label distribution has prolonged the training and affected the classification performance.

In supervised learning (SL), re-weighting (Cao et al., 2019; Tan et al., 2020) and re-sampling (Kang et al., 2020; 2019; Chawla et al., 2002) based methods have shown their success. However, they were designed for SL and require label information of the entire dataset. Hence they may not be directly applicable in SSL settings where the majority samples are unlabeled. A few recent studies (Kim et al., 2020; Yang & Xu, 2020; Hyun et al., 2020; Wei et al., 2021) try to address the issue in SSL settings, but all of these methods may introduce additional heavy computations (e.g. complicated re-sampling (Wei et al., 2021), pseudo label alignment (Kim

[*]Equal contribution [1]Department of Electrical and Computer Engineering, University of California Davis, Davis, CA, USA [2]Department of Statistics and Data Science, Southern University of Science and Technology, Shenzhen 518055, P.R. China (Work done at UC Davis.) [3]Department of Electrical and Computer Engineering, University of Kentucky, Lexington, KY, USA. Correspondence to: Zhengfeng Lai <lzhengfeng@ucdavis.edu>.

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



(a) (b) (c)

Figure 1. Experimental results on CIFAR-10 under imbalanced ratio $\gamma = 100$. (a) Class distribution of labeled and unlabeled data. (b) Estimated distribution of majority classes $\{1, 2, 3, 4, 5\}$. (c) Estimated distribution of minority classes $\{6, 7, 8, 9, 10\}$.

et al., 2020)) and have made assumptions about the distribution of the unlabeled data. For example, CReST (Wei et al., 2021) assumes that unlabeled data have almost the same distribution as labeled data. DARP (Kim et al., 2020) utilizes the confusion matrix from labeled data to estimate unlabeled data distribution. However, in real-world scenarios (such as in gigapixel pathology imaging problems (Lai et al., 2021)), such prior information about the unlabeled data is often unavailable. This is the second challenge SSL faces in realistic tasks.

Contributions. As re-sampling based techniques may introduce significant computational cost (Lee et al., 2021), we focus on the feasibility of adapting re-weighting techniques from SL to SSL settings. We first relax the second challenge by following (Wei et al., 2021) to assume the unlabeled data share similar distribution as the labeled data. Under this setting, we verify the effectiveness and necessity of smoothed weighting schemes on the consistency loss in countering the class imbalance problem. We then turn to the second challenge of unknown distribution of unlabeled data and propose Smoothed Adaptive Weighting (SAW) that can automatically estimate the learning difficulty for each class and assign weights to the consistency loss per epoch.

We demonstrate the compatibility of SAW to any existing pseudo-label based SSL algorithms and the effectiveness of SAW under various imbalanced degrees among the labeled and unlabeled sets. SAW improves both ReMixMatch (Berthelot et al., 2020) and FixMatch (Sohn et al., 2020) by up to 30.0% and 40.5%, respectively, in terms of absolute geometric mean scores (GM) gain. Furthermore, SAW outperforms DARP (Kim et al., 2020), a recent work that specifically tackles the imbalance issue in SSL, by up to 10.9%. To demonstrate the robustness of SAW under different distributions, we also evaluate it on a hold-out test set that has completely different distribution from the training set and achieve up to 15% of absolute performance gain over DARP (Kim et al., 2020). Lastly, we test SAW on a gigapixel-level pathology dataset (Lai et al., 2021) to verify the applicability of our algorithm on a **real use-case in medical imaging**. Our work differs from the prior work in that it deals with multi-class problems without making or relying on any assumptions about the unlabeled data.

2. Related works

Semi-supervised learning (SSL). One branch of SSL is pseudo labeled-based methods (Sohn et al., 2020; Berthelot et al., 2020; Arazo et al., 2020), which generate pseudo labels for unlabeled samples and use them in the following training loop. To promote consistency across predictions over unlabeled samples, consistency regularization can be applied across different augmented views of the same unlabeled training samples (Tarvainen & Valpola, 2017; Miyato et al., 2018). Recent state-of-the-art SSL algorithms such as FixMatch (Sohn et al., 2020) and ReMixMatch (Berthelot et al., 2020) combine both pseudo labels and consistency regularization. However, all of these aforementioned methods assume balanced class distribution in both labeled and unlabeled data and may face performance degradation in real-world imbalanced datasets (Calderon-Ramirez et al., 2020; Pulido et al., 2020).

Imbalanced supervised learning. Real-world datasets are typically of "long-tailed" distribution (Liu et al., 2019; Van Horn et al., 2018; Jamal et al., 2020; Liu et al., 2020). Supervised learning on imbalanced distribution has been widely studied. Popular methods to re-balance the class distributions include re-sampling the data (Chawla et al., 2002; Buda et al., 2018; Pouyanfar et al., 2018; Kang et al., 2019) and re-weighting the loss function (Aurelio et al., 2019; Cao et al., 2019; Tan et al., 2020; Khan et al., 2017; Menon et al., 2020). Although these methods perform effectively on **supervised learning** where all samples are labeled, their applicability to SSL is unclear considering the majority of samples are unlabeled in SSL settings.

Imbalanced semi-supervised learning. For multi-class classification, CReST (Wei et al., 2021) is a class-rebalancing self-training framework to expand the labeled data by adding pseudo-labeled samples. One obvious draw-

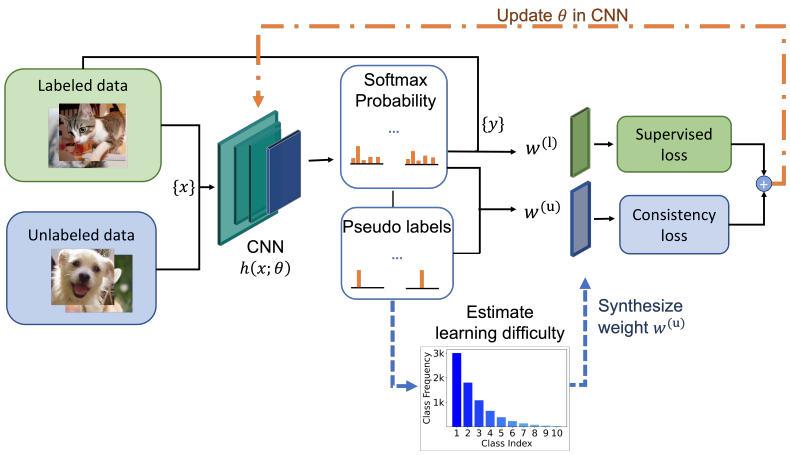


Figure 2. Illustration of the SAW framework.

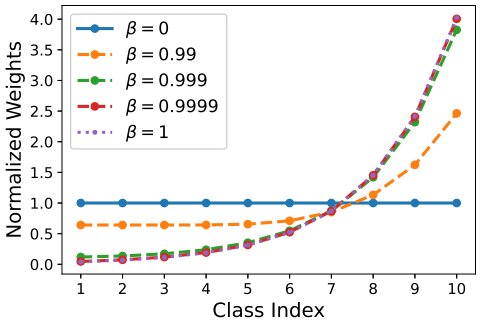


Figure 3. The normalized weights generated from setting different values of β by using the smoothing function (4).

back is that it directly assumes labeled and unlabeled data are of almost the same distribution, which may not be true in realistic scenarios. DARP (Kim et al., 2020) introduced a distribution aligning scheme to softly refine the pseudo labels generated from the model. However, DARP also assumes that labeled and unlabeled data can be sampled from similar input distribution: they approximate the confusion matrix of unlabeled data as that of labeled data. Another limitation of these methods is that they may introduce significantly additional computation cost.

3. Methodology

3.1. Preliminaries

Consider a multi-class classification problem with C classes ($C > 2$). An input vector and corresponding one-hot label are denoted $x \in \mathbb{R}^d$ and $y \in \{0,1\}^C$, respectively. Here d is the dimension of the input. The training data contains m labeled instances $\mathcal{D}^{(l)} = \left\{(x_i^{(l)}, y_i^{(l)})\right\}_{i=1}^m$ and n unlabeled instances $\mathcal{D}^{(u)} = \left\{x_j^{(u)}\right\}_{j=1}^n$, with $m \ll n$. The number of training labeled instances in $\mathcal{D}^{(l)}$ of class k is denoted as m_k, i.e., $\sum_{k=1}^C m_k = m$. Similarly, we have n_k for unlabeled instances in $\mathcal{D}^{(u)}$ of class k with $\sum_{k=1}^C n_k = n$. In general, the distribution of unlabeled data $\{n_k\}_{k=1}^C$ is unknown. Without loss of generality, we assume $\{m_k\}$ and $\{n_k\}$ are sorted by cardinality in descending order, i.e., $m_1 \geq m_2 \geq \cdots \geq m_C$, and $n_1 \geq n_2 \geq \cdots \geq n_C$. In a class-imbalanced dataset, the class distribution is skewed, i.e., $m_1 \gg m_C$ or $n_1 \gg n_C$. Here we define $\gamma_l = \frac{m_1}{m_C}$ and $\gamma_u = \frac{n_1}{n_C}$ to measure the degree of class imbalance in labeled and unlabeled data, respectively.

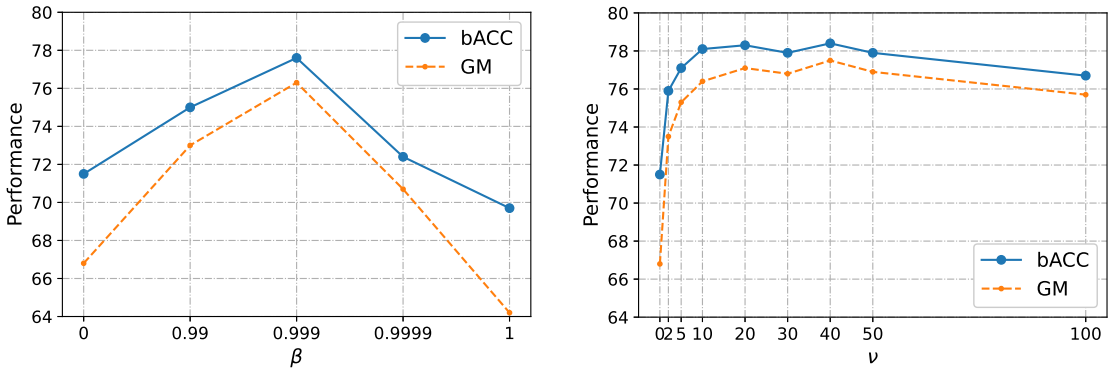
The goal of SSL is to train a classifier $h(x;\theta) : \mathbb{R}^d \to [0,1]^C$ using the above training data. Here $h(x;\theta)$ is parameterized by $\theta \in \Theta$ and its k-component $h(x;\theta)_k \in [0,1]$ is the predictive probability for the k-th class given an input x. The training process is to minimize the loss in both labeled and unlabeled data

$$\min_{\theta\in\Theta} \sum_{i=1}^m \mathcal{L}_l(x_i^{(l)}, y_i^{(l)}; \theta) + \sum_{j=1}^n \Omega(x_j^{(u)}; \theta), \qquad (1)$$

where $\mathcal{L}_l$ is the per-sample supervised loss (e.g., cross-entropy) for labeled data and Ω is the per-sample regularization for unlabeled data. Note that some SSL algorithms (e.g., FixMatch (Sohn et al., 2020)) include both labeled and unlabeled data in the regularization term, where we can simply replace the second term of (1) by $\sum_{j=1}^{n+m} \Omega(x_j;\theta)$, for $x_j \in \mathcal{D}^{(l)} \cup \mathcal{D}^{(u)}$. One of the widely used regularization is consistency loss (Tarvainen & Valpola, 2017; Sohn et al., 2020; Miyato et al., 2018). This kind of regularization forces SSL to predict consistency on different views of the same sample. To utilize $\mathcal{D}^{(u)}$ effectively, many SSL algorithms infer their labels by some pseudo labeling scheme p defined as $\left\{p(x_j^{(u)};\theta) \in [0,1]^C \,\middle|\, \sum_{k=1}^C p(x_j^{(u)};\theta)_k = 1\right\}_{j=1}^n$. Then, we can formulate the consistency loss as follow:

$$\mathcal{L}_c(x;\theta) := \sum_{k=1}^C p(x;\theta)_k \cdot \log(h(\mathtt{pt}(x);\theta)_k). \qquad (2)$$

Here $\mathtt{pt}(x)$ defines the specific augmentation scheme. In VAT (Miyato et al., 2018), this operator is an adversarial perturbation that disregards the input class distribution. FixMatch (Sohn et al., 2020) adopts strong and weak augmentation as perturbations in $\mathtt{pt}(x)$ and $p(x;\theta)$, respectively. Note that other consistency loss like the mean-square error can also be used: $\|p(x;\theta) - h(\mathtt{pt}(x);\theta)\|_2^2$. For simplicity, in the following content, we will only consider consistency loss (2) as the regularization term, i.e., $\Omega = \mathcal{L}_c$.



(a) (b)

Figure 4. Empirical illustration of the performance with smoothing weighting functions under FixMatch (Sohn et al., 2020) on CIFAR10-LT (the imbalance ratio $\gamma = 100$). (a): The performance by using smoothing function (4) and setting different β. (b) The performance by using smoothing function (5) and setting different ν.

3.2. A deeper look at weighting the consistency loss

In this subsection, we take a deeper look at the consistency loss and investigate the feasibility of adding weights to the consistency loss, which differs from traditional cross-entropy loss studied in other imbalanced learning works. Here we define a weighted consistency loss as

$$\mathcal{L}_{cw}(x; w, \theta) := \sum_{k=1}^{C} w_k \cdot p(x;\theta)_k \cdot \log(h(\texttt{pt}(x);\theta)_k), \quad (3)$$

where w_k is the weight of the k-th class. In supervised learning (SL), the assigned weights are associated with the number of samples for each class (Aurelio et al., 2019; Cao et al., 2019; Tan et al., 2020; Khan et al., 2017; Menon et al., 2020). However, in SSL settings, such information for the unlabeled set is typically unavailable. Before we fully address this challenge, we first assume that the unlabeled data share similar distribution as the labeled data and use this as the prior assumption to study the weighting schemes for the consistency loss. Then we will discuss how to synthesize the weights when the distribution is inconsistent and unknown in Section 3.3.

Assumption 3.1. Unlabeled data share similar distribution as the labeled set, i.e., $\gamma_l = \gamma_u$.

The distribution of unlabeled data is assumed accessible under this strong assumption to help us take a deeper analysis on the feasibility of weighting the consistency loss and what weighting schemes should be applied if it is feasible. A long-tailed version of the CIFAR-10 (Krizhevsky, 2009) with a class-imbalanced ratio $\gamma := \gamma_l = \gamma_u = 100$ is studied. We test FixMatch (Sohn et al., 2020) with its consistency loss replaced by $\mathcal{L}_{cw}$. With such assumption, we first apply the weights by strictly inverse the class frequency to the consistency loss. As the imbalance ratio is severe here, we find the gradients can be accumulated to be abnormally large and result in unstable training (An et al., 2020; Cao et al., 2019), which may even hurt the original model (as shown in Figure 4(a) when $\beta = 1$).

Effective re-weighting (Cui et al., 2019) is one of a typical smoothed weighting schemes for supervised learning. We adapt the corresponding weighting function into SSL as one example to further investigate whether weighting schemes are applicable to the consistency loss and what type of weights should be assigned. In (Cui et al., 2019), weights are set to be inversely proportional to the *effective number* of samples for each class, i.e.,

$$w_k \propto 1/E_k, \text{ where } E_k = (1-\beta^{n_k})/(1-\beta). \quad (4)$$

The degree of flattening is determined by the value of β as $\beta = 0$ leads to uniform weights. When β goes to 1, $E_k \to n_k$, which corresponds to the weights by strictly inverse class frequency. The normalized weights with different values of β are shown in Figure 3. The performance is evaluated by balanced accuracy (bACC) (Wang et al., 2017; Huang et al., 2016) and geometric mean scores (GM) (Branco et al., 2016; Kubat, 1997). In Figure 4(a), we observe that selecting weights by either strictly inverse class frequency or uniform weights produce the worst results while smoothed versions of weights show some improvements. This observation indicates the importance of the smoothness for the weights added to consistency loss.

To further verify the necessity of smoothed weights in consistency loss, we study other smoothing functions, such as

$$w_k \propto \frac{n}{n + \nu \cdot n_k}, \text{ for } k = 1, \ldots, C, \quad (5)$$

where ν is the hyper-parameter to describe the smoothing degree. Figure 4(a) shows the results of Equation (4) and Figure 4(b) shows the results of one alternative smoothing

Algorithm 1 The SAW Framework

Data: Labeled data $\{(x_i, y_i)\}_{i=1}^m$, unlabeled data $\{x_j\}_{j=1}^n$, weight in supervised loss $w^{(1)}$, number of classes C, learning rate η, max epoch T
Initialize uniform weights in consistency loss $w^{(u)}$ and model parameter θ
while $t < T$ **do**
  for `1, ...` K **do**
    Sample batches label data $\{x, y\}$
    Sample batches unlabeled data $\{x\}$
    Update $\theta \leftarrow \theta - \eta \cdot \nabla \mathcal{L}(\theta, w^{(u)})$
  end for
  Compute pseudo labels $p(x; \theta)$ and estimate distribution $(n_1, n_2, \ldots, n_C)$
  Adjust the distribution $(\hat{n}_1, \hat{n}_2, \ldots, \hat{n}_C)$
  Update weight $w^{(u)}$ based on smoothed weighting function
end while
Return: θ

function (5): both of them provide some improvements when the weights are smoothed. Therefore, we verify the effectiveness of smoothed weights when the assumption that unlabeled data share similar distribution as the labeled data holds true in this subsection.

3.3. SAW: Smoothed adaptive weighting against unknown distribution data

In real-world scenarios, the distribution of unlabeled data is unknown and **Assumption 3.1** cannot hold true. For example, in medical imaging applications (Lai et al., 2021; Tang et al., 2019), the distribution of gigapixel pathology images is unavailable unless the neuropathologists fully annotate the majority of samples. The weights in consistency loss therefore cannot be directly inferred from labeled data under this setting. In (Kim et al., 2020), the distribution of unlabeled data is estimated from the confusion matrix of a subset of labeled dataset. This estimation is limited as it assumes that confusion matrices of labeled and unlabeled datasets are similar. Besides, when labeled data are scarce, the confusion matrix of a subset of labeled dataset may fail to represent the whole dataset. In this subsection, we propose the learning difficulty to represent the whole dataset and synthesize the weights for the consistency loss, which does not rely on any assumptions about the unlabeled data.

Learning difficulty and adaptive weighting. A recent work (Zhang et al., 2021) indicates that the learning effect of a class can be reflected by the number of predicted samples. When we take a deeper look into Figure 1(b) and (c), the number of pseudo labels for minority classes is significantly underestimated by FixMatch. Hence we follow the finding from (Zhang et al., 2021) to use the number of pseudo labels to reflect the the learning effect of each class during the training process. Here we refer to the minority classes as difficult classes and use the number of pseudo labels for one class as the metric to measure the learning difficulty of the model on that class. Fewer pseudo labels indicate that the corresponding class is a difficult class for the model.

Instead of aligning the pseudo labels via estimating the distribution, our aim is to alleviate the confirmation bias of pseudo labels based on the estimated learning difficulty. This can be implemented by adaptively synthesizing the smoothed weight term $w^{(u)}$ for consistency loss, which will encourage the model to focus on difficult classes. In general, we utilize gradient descent methods or its variants, such as Adam (Kingma & Ba, 2015), to update the θ in the training process, i.e., $\theta \leftarrow \theta - \eta \cdot \nabla \mathcal{L}(\theta, w^{(u)})$, where η is the learning rate and $\mathcal{L}$ is the loss function: $\mathcal{L}(\theta, w^{(u)}) = \sum_{i=1}^m \mathcal{L}_{lw}(x_i^{(l)}, y_i^{(l)}; w^{(l)}, \theta) + \sum_{j=1}^n \mathcal{L}_{cw}(x_j^{(u)}; w^{(u)}, \theta)$. The weight $w^{(l)}$ is fixed based on the distribution of the labeled data, while the weight $w^{(u)}$ is initialized to be uniform. After training for one epoch, we compute the one-hot pseudo label $p(x, \theta)$ via the updated θ. Then class distributions can be estimated by $n_k = \sum_{j=1}^n p(x_j, \theta)_k$, for $k = 1, \ldots, C$. Take the effective re-weighting (Cui et al., 2019) as one example here. The weights can be proportional to $(1 - \beta)/(1 - \beta^{n_k})$. But singularity may rise when estimated $n_k = 0$. Here we adjust n_k by $\hat{n}_k = \max(n_k, 1)$. Note that this weighting function (Cui et al., 2019) is **not** a requisite and can be replaced by other smoothed weighting functions described in Section 3.2. During the training process, improved quality of pseudo labels could result in a better estimation of distribution. We call this proposed algorithm as smoothed adaptive weighting (SAW), which is summarized in Algorithm 1 and Figure 2.

4. Experiments

In this section, we evaluate the efficacy of SAW on various class-imbalanced scenarios for SSL in multi-class classification problems, including CIFAR-10, CIFAR-100 (Krizhevsky, 2009), and STL-10 (Coates et al., 2011). The evaluation criteria on these imbalanced classification problems include *balanced accuracy* (bACC) (Huang et al., 2016; Wang et al., 2017) and *geometric mean scores* (GM) (Branco et al., 2016). We run each algorithm for three trials to report the mean value and standard deviation under each setting. All experiments use the same encoder (Wide ResNet-28-2 (Oliver et al., 2018)) and batch size (64) under the same number of training iterations (around 2.5×10^5). Lastly, we evaluate SAW on a gigapixel pathology imaging dataset (Lai et al., 2021) to further verify the applicability of SAW in practical settings with unavailable distribution. Our codes can be found at https://github.com/ZJUJeffLai/SAW_SSL.

Table 1. Comparison of classification performance on CIFAR10-LT under $\gamma = \gamma_l = \gamma_u$ (hold-out test set is balanced). The evaluation criterion is bACC/GM. The best results are in bold.

Algorithm	SSL	RB	$\gamma = 50$	$\gamma = 100$	$\gamma = 150$
Wide ResNet-28-2 (Oliver et al., 2018)	-	-	65.2±0.05 / 61.1±0.09	58.8±0.13 / 51.0±0.11	55.6±0.43 / 44.0±0.98
Re-sampling (Japkowicz, 2000)	-	✓	64.3±0.48 / 60.6±0.67	55.8±0.47 / 45.1±0.30	52.2±0.05 / 38.2±1.49
LDAM-DRW (Cao et al., 2019)	-	✓	68.9±0.07 / 67.0±0.08	62.8±0.17 / 58.9±0.60	57.9±0.20 / 50.4±0.30
cRT (Kang et al., 2020)	-	✓	67.8±0.13 / 66.3±0.15	63.2±0.45 / 59.9±0.40	59.3±0.10 / 54.6±0.72
ReMixMatch (Berthelot et al., 2020)	✓	-	81.5±0.26 / 80.2±0.32	73.8±0.38 / 69.5±0.84	69.9±0.47 / 62.5±0.35
ReMixMatch + cRT (Kang et al., 2019)	✓	✓	86.8±0.50 / 86.5±0.49	81.4±0.41 / 80.7±0.45	78.9±0.84 / 77.8±0.94
ReMixMatch + DARP (Kim et al., 2020)	✓	-	82.1±0.14 / 80.8±0.09	75.8±0.09 / 72.6±0.24	71.0±0.27 / 64.5±0.68
ReMixMatch + DARP + cRT (Kim et al., 2020)	✓	✓	87.3±0.61 / 87.0±0.11	83.5±0.07 / 83.1±0.09	79.7±0.54 / 78.9±0.49
ReMixMatch + CReST (Wei et al., 2021)	✓	✓	85.2±0.19 / 84.9±0.25	76.2±0.31 / 75.1±0.28	71.4±0.23 / 67.5±0.40
ReMixMatch + SAW	✓	-	86.3±0.61 / 86.1±0.64	77.0±0.59 / 76.0±0.42	71.5±0.30 / 68.9±0.26
ReMixMatch + SAW + cRT (Kang et al., 2019)	✓	✓	**87.6**±0.21 / **87.4**±0.26	**85.4**±0.32 / **83.9**±0.21	**79.9**±0.15 / **79.9**±0.12
FixMatch (Sohn et al., 2020)	✓	-	79.2±0.33 / 77.8±0.36	71.5±0.72 / 66.8±1.51	68.4±0.15 / 59.9±0.43
FixMatch + DARP (Kim et al., 2020)	✓	-	81.8±0.24 / 80.9±0.28	75.5±0.05 / 73.0±0.09	70.4±0.25 / 64.9±0.17
FixMatch + CReST (Wei et al., 2021)	✓	✓	83.0±0.39 / 81.5±0.17	75.7±0.38 / 72.7±0.85	70.8±0.25 / 64.5±0.31
FixMatch + CReST + & LA (Wei et al., 2021)	✓	✓	85.6±0.36 / 81.9±0.45	**81.2**±0.70 / 74.5±0.99	71.9±2.24 / 64.4±1.75
FixMatch + SAW	✓	-	84.0±0.10 / 83.6±0.12	77.5±0.63 / 76.3±0.80	71.6±0.35 / 69.7±0.46
FixMatch + SAW + & LA (Menon et al., 2020)	✓	✓	**86.2**±0.15 / **83.9**±0.35	80.7±0.15 / **77.5**±0.21	**73.7**±0.06 / **71.2**±0.17

Table 2. Comparison of classification performance on CIFAR10-LT under $\gamma_l = 100, \gamma_u \neq \gamma_l$ (hold-out test set is balanced). The evaluation criterion is bACC/GM. The best results are in bold.

Algorithm	$\gamma_u = 1$	$\gamma_u = 50$	$\gamma_u = 150$
ReMixMatch (Berthelot et al., 2020)	48.3±0.14 / 19.5±0.85	75.1±0.43 / 71.9±0.77	72.5±0.10 / 68.2±0.32
ReMixMatch* (Berthelot et al., 2020)	85.0±1.35 / 84.3±1.55	77.0±0.12 / 74.7±0.04	72.8±0.10 / 68.8±0.21
ReMixMatch* + DARP (Kim et al., 2020)	**89.7**±0.15 / **89.4**±0.17	77.4±0.22 / 75.0±0.25	73.2±0.11 / 69.2±0.31
ReMixMatch* + CReST (Wei et al., 2021)	45.9±1.27 / 20.1±1.99	70.2±0.45 / 65.8±0.71	65.4±0.34 / 62.9±0.15
ReMixMatch* + SAW	88.3±0.15 / 88.9±0.10	**80.3**±0.36 / **79.6**±0.40	**74.0**±0.94 / **72.4**±0.94
FixMatch (Sohn et al., 2020)	68.9±1.95 / 42.8±8.11	73.9±0.25 / 70.5±0.52	69.6±0.60 / 62.6±1.11
FixMatch + DARP (Kim et al., 2020)	**85.4**±0.55 / **85.0**±0.65	77.3±0.17 / 75.5±0.21	72.9±0.24 / 69.5±0.18
FixMatch + CReST (Wei et al., 2021)	60.2±1.34 / 35.9±2.50	65.8±0.78 / 67.1±0.84	60.1±1.44 / 51.4±1.68
FixMatch + SAW	83.9±0.44 / 83.3±0.47	**81.5**±2.25 / **80.9**±2.30	**76.8**±0.31 / **75.4**±0.37

4.1. CIFAR10-LT

Setup. We evaluate SAW on a "synthetically long-tailed" variant of CIFAR-10 (CIFAR10-LT) introduced in (Kim et al., 2020; Cui et al., 2019; Cao et al., 2019). In the training set, all images are randomly sampled based on the pre-defined imbalance ratios: γ_l is for labeled data while γ_u is for unlabeled data. We set $m_1 = 1500$ and $n_1 = 3000$, then have $m_k = m_1 \cdot \gamma_l^{\epsilon_k}$ and $n_k = n_1 \cdot \gamma_l^{\epsilon_k}$, where $\epsilon_k = -\frac{k-1}{C-1}$. After that, we vary γ_l and γ_u to construct multiple CIFAR10-LT to mimic different imbalanced scenarios. The hold-out test set remains untouched and balanced (Kim et al., 2020).

Main results under $\gamma_l = \gamma_u$. Following (Kim et al., 2020), we evaluate SAW assuming that labeled and unlabeled sets are sampled from almost the same distributions by setting γ at 50, 100, 150, respectively (Kim et al., 2020). To verify the effectiveness of SAW, we apply it to the recent state-of-the-art SSL algorithms: ReMixMatch (Berthelot et al., 2020) and FixMatch (Sohn et al., 2020). To have a comprehensive comparison, we selected several benchmarks: 1) Wide ResNet-28-2 (Oliver et al., 2018) as a supervised learning baseline without using any unlabeled data and re-balancing algorithms; 2) Re-sampling (Japkowicz, 2000); 3) Label distribution-aware margin (LDAM-DRW) (Cao et al., 2019); 4) classifier Re-Training cRT (Kang et al., 2020). We compare SAW with recent imbalanced SSL algorithms, DARP (Kim et al., 2020) and CReST (Wei et al., 2021). The results are summarized in Table 1. SAW consistently improves both FixMatch and ReMixMatch on bACC and GM for all evaluated settings. Specifically, SAW improves the bACC of FixMatch up to 6% absolute bACC gain and achieves up to 9.8% absolute GM gain compared to the original FixMatch. Besides, it also improves up to 4.8%/6.5% at absolute bACC/GM gain compared to ReMixMatch. SAW consistently outperforms the recent imbalanced algorithm (DARP) for both bACC and GM in all settings. In (Kim

Table 3. Comparison of classification performance on CIFAR100-LT under $\gamma = \gamma_l = \gamma_u$, and STL-10 where the distribution of unlabeled data is unknown. Hold-out test set is balanced. The evaluation criterion is bACC/GM. The best results are in bold.

Algorithm	CIFAR100-LT ($\gamma = \gamma_l = \gamma_u$) $\gamma = 10$	CIFAR100-LT ($\gamma = \gamma_l = \gamma_u$) $\gamma = 20$	STL-10 $\gamma_l = 10$	STL-10 $\gamma_l = 20$
ReMixMatch (Berthelot et al., 2020)	59.2±0.03 / 52.1±0.13	53.5±0.03 / 42.3±0.13	67.8±0.45 / 61.1±0.92	60.1±1.18 / 44.9±1.52
ReMixMatch* + DARP (Kim et al., 2020)	59.8±0.20 / 52.9±0.41	54.4±0.07 / 44.2±0.07	79.4±0.07 / 78.2±0.10	70.9±0.44 / 67.0±1.62
ReMixMatch* + CReST (Wei et al., 2021)	59.6±0.32 / 52.5±0.24	53.9±0.15 / 43.9±0.19	65.3±0.23 / 59.3±0.41	55.8±2.05 / 40.2±2.39
ReMixMatch* + SAW	**61.8**±0.06 / **56.9**±0.40	**55.3**±0.26 / **46.3**±0.65	**82.0**±0.55 / **81.0**±0.64	**79.2**±0.44 / **77.9**±0.52
FixMatch (Sohn et al., 2020)	60.1±0.05 / 54.4±0.11	54.0±0.04 / 44.4±0.17	72.9±0.09 / 69.6±0.01	63.4±0.21 / 52.6±0.09
FixMatch + DARP (Kim et al., 2020)	61.1±0.23 / 56.4±0.28	54.9±0.05 / 46.4±0.41	77.8±0.33 / 76.5±0.40	69.9±1.77 / 65.4±3.07
FixMatch + CReST (Wei et al., 2021)	60.8±0.32 / 54.9±0.45	54.5±0.21 / 44.8±0.19	70.5±0.75 / 67.8±0.32	60.5±2.13 / 50.3±2.56
FixMatch + SAW	**62.1**±0.25 / **58.0**±0.20	**55.7**±0.10 / **49.4**±0.40	**78.3**±0.25 / **77.0**±0.19	**71.9**±0.81 / **69.0**±0.81

Table 4. Comparison of classification performance on CIFAR10-LT under $\gamma = \gamma_l = \gamma_u$ (hold-out test set is of reversed distribution). The evaluation criterion is bACC/GM. The best results are in bold.

Algorithm	$\gamma = 50$	$\gamma = 100$	$\gamma = 150$
ReMixMatch (Berthelot et al., 2020)	71.0±0.55 / 83.5±0.29	54.7±0.51 / 74.4±0.47	41.5±1.69 / 66.4±1.22
ReMixMatch + DARP (Kim et al., 2020)	66.9±0.75 / 80.5±0.46	49.7±1.55 / 70.5±0.90	35.8±1.81 / 60.9±2.42
ReMixMatch + CReST (Wei et al., 2021)	64.3±0.25 / 75.7±0.34	51.2±0.92 / 72.1±0.85	39.2±1.46 / 65.8±1.88
ReMixMatch + SAW	**86.3**±0.61 / **86.1**±0.64	**77.0**±0.59 / **76.0**±0.42	**71.5**±0.30 / **68.9**±0.26
FixMatch (Sohn et al., 2020)	70.5±0.26 / 82.2±0.31	51.0±1.65 / 71.5±1.24	38.5±1.15 / 63.4±0.31
FixMatch + DARP (Kim et al., 2020)	72.2±0.62 / 82.8±0.17	57.6±0.36 / 74.8±0.48	46.5±1.26 / 68.1±0.10
FixMatch + CReST (Wei et al., 2021)	69.4±0.35 / 80.1±0.41	52.4±0.32 / 70.3±0.28	42.9±1.45 / 67.4±1.07
FixMatch + SAW	**78.7**±0.77 / **84.2**±0.36	**64.3**±1.96 / **76.4**±0.88	**57.5**±2.83 / **70.5**±1.50

et al., 2020), they showed the compatibility of DARP with cRT (Kang et al., 2019) under ReMixMatch (Berthelot et al., 2020); in (Wei et al., 2021) they showed the compatibility of CReST with LA (Menon et al., 2020). Therefore, to have a fair comparison, we follow their methodologies to design experiments to evaluate the combination of SAW and cRT/LA. We find SAW is compatible with cRT/LA and its performance can be further improved.

Main results under $\gamma_l \neq \gamma_u$. In real-world scenarios, it is natural that unlabeled data have different distributions, i.e., $\gamma_l \neq \gamma_u$. Under this setting, traditional methods (LA (Menon et al., 2020) and cRT (Kang et al., 2019)) designed for SL may not be applicable considering the unknown distribution of unlabeled sets. We acknowledge that CReST (Wei et al., 2021) is originally designed with the assumption that the distribution of the unlabeled data is similar to that of the labeled data, and hence may not work well with the various scenarios we consider. However, we include the baseline results for completion. In Table 2, SAW shows its superiority when unlabeled data have different imbalanced ratio from labeled data, especially when unlabeled data are also severely imbalanced (such as $\gamma_u = 50/150$). In these settings, SAW improves original ReMixMatch up to 5.2% of bACC and 7.7% of GM. Meanwhile, it improves original FixMatch up to 15% of bACC and 40.5% of GM. Even compared to DARP, SAW outperforms up to 4.2% of bACC and 5.9% of GM. Although the performance of SAW on balanced unlabeled data is slightly lower than DARP, it still improves original FixMatch and ReMixMatch significantly. Considering the fact that DARP needs to explicitly assume similar distribution between labeled and unlabeled set (Kim et al., 2020), SAW has the potential to enhance the practicability of SSL in the real-world datasets where the distribution of unlabeled data is unavailable.

4.2. CIFAR100-LT and STL-10

Setup. Besides CIFAR10-LT, we also evaluate our framework on CIFAR-100 (Krizhevsky, 2009) and STL-10 (Coates et al., 2011). For CIFAR-100, we follow the above process to set γ and prepare a long-tailed training set as CIFAR100-LT, where we set $m_1 = 150$ and $n_1 = 300$. For STL-10, we set m_1 as 450 and construct a similar "synthetically long-tailed" version of labeled data; all unlabeled data are used since their class distribution is unknown. Hence the imbalanced ratio of labeled data can be different from that of unlabeled data ($\gamma_l \neq \gamma_u$). **Main results.** The results on CIFAR100-LT and STL-10 are summarized in Table 3. Note that the gain of SAW is significant under $\gamma_l = 20$ in STL-10: 19.1%/30.0% of improvements at bACC/GM for ReMixMatch, 18.5%/16.4% of improve-

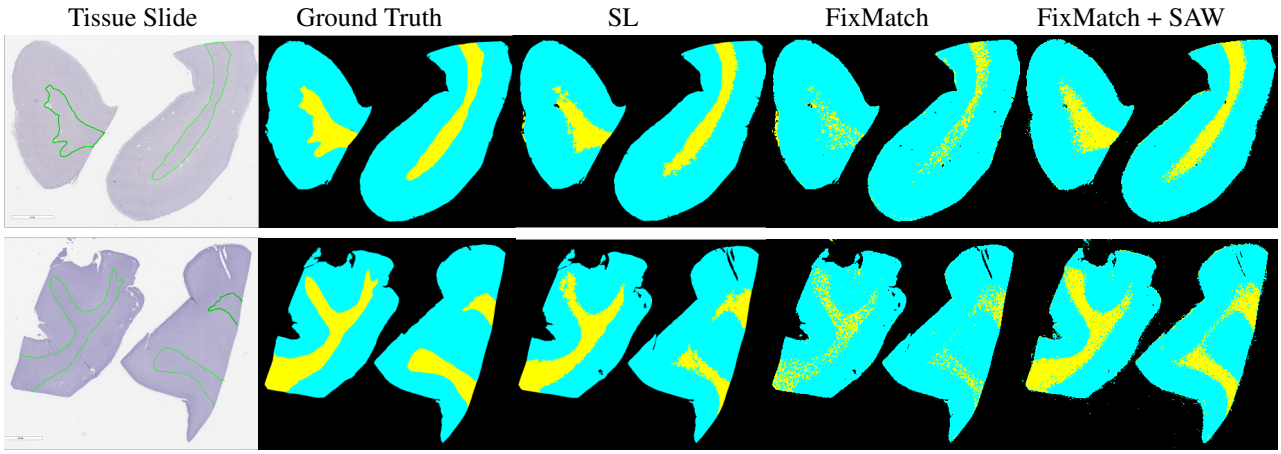


Figure 5. Segmentation masks visualization: GM, WM, and background are indicated by cyan, yellow, and black, respectively. SL refers to train supervised learning (U-Net (Oskal et al., 2019)) on all 20 pathology images with their annotations. The SSL results are only using 0.1% area of 2 pathology images selected from the training set.

ments at bACC/GM for FixMatch. When we compare it with DARP, it still outperforms DARP up to 8.3%/10.9% on ReMixMatch. This is due to the vast mismatch between the distribution of labeled data and that of unlabeled data. Therefore, the results here prove the superiority of SAW for both matching and non-matching distributions between the labeled and unlabeled data.

4.3. Empirical analysis on SAW

Stress-test. The previous sections report the results on a balanced test set. In reality, if the training set is severely imbalanced, the hold-out test set can also be imbalanced, and even follow a totally different distribution compared to the training set. We first evaluate all algorithms on a test set that is of the same distribution as training set, but we find all algorithms perform very well as expected. Thus, to further stress-test these algorithms, we construct a hold-out test set with a flipped distribution compared to the training data. Table 4 summarizes the results on the situations where labeled and unlabeled data are sampled from the same distribution. Under this setting, DARP (Kim et al., 2020) even hurt the performance of original ReMixMatch. SAW improves both original SSL algorithms and outperforms DARP (Kim et al., 2020). SAW improves ReMixMatch up to 30% at bACC and 2.6% at GM. Similarly, SAW also improves original FixMatch up to 19% at bACC and 7.1% at GM. We find that DARP may degrade the performance of original ReMixMatch while SAW consistently outperforms both DARP and baselines. For the settings where unlabeled data have **different** distributions (as shown in Appendix B.2), SAW achieves more surprising improvements, with up to 15% of absolute performance gain compared to DARP.

Computational complexity. In SAW, as we neither have any resampling actions nor alignment optimization, the additional running time constitutes only 5% of the original SSL models as we simply add the weights. However, the additional running time incurred by DARP due to the pseudo-label alignment can be up to 20% of that of the original SSL algorithms. For CReST, as it iteratively samples the training data and re-initializes the classifier's network, the additional running time can be an order of magnitude longer than the original SSL algorithms.

4.4. Additional evaluation on a real-world application

Besides above standard datasets, we further evaluate SAW on a gigapixel pathology imaging dataset (Lai et al., 2021) as a real-world application. In this dataset, each slide is at gigapixel level and the goal is to segment white matter (WM) from grey matter (GM). We follow the settings of a recent work (Lai et al., 2021): 0.1% of regions from 2 slides are labeled and the hold-out test set includes 10 slides. Figure 5 shows the segmentation masks from the SL (U-Net (Oskal et al., 2019)) trained with all pathology images annotated at pixel level, original FixMatch (Sohn et al., 2020), and SAW without any prior information from the unlabeled data by using only 0.1% regions from 2 slides as the labeled set. Figure 5 clearly shows the performance gain of SAW on WM (the minority class): original SSL algorithm has difficulty classifying WM while SAW improves significantly. Statistically, for the minority class - WM, SAW improves the IoU score (Rahman & Wang, 2016) by over 22%; for the majority class - GM, it also achieves nearly 4% of improvement. For the DICE coefficient (Zou et al., 2004), another metric, SAW brings up to 15.9% of improvement on the minority class and 1.6% on the majority class. More numerical results are in the Appendix B.8. Considering the fact that we truly regard the dataset as unlabeled and does not make any assumption on the unlabeled data, we

have verified the applicability of SAW on a medical case. We do not include the comparison with DARP (Kim et al., 2020) and CReST (Wei et al., 2021) here as the distribution of the unlabeled set in this case is unavailable without neuropathologists' assessments.

5. Discussion

In this work, we have presented the effectiveness of smoothed adaptive weighting scheme (SAW) on imbalanced semi-supervised learning via various imbalanced scenarios using several public datasets. We also evaluate SAW on a pathology imaging dataset to show its reliability against unknown distribution data. We verify the feasibility of adding weights to the consistency loss and the necessity of smoothed weights. Based on this, we estimate learning difficulty of each class and synthesize the smoothed weights based on this estimation, which can overcome the challenge of unknown distribution in unlabeled data. When combined with two leading SSL methods FixMatch and ReMixMatch, SAW shows consistent improvements over the baselines and those augmented with DARP in almost all experiment settings. The two main advantages of SAW are: first, it does not require any knowledge of the class distribution of the unlabeled data and work well even in the case when the labeled data, unlabeled data and testing data all assume vastly different distributions; second, it can be easily integrated with any SSL algorithms that use consistency loss.

Acknowledgements

This work was supported by the NSF HDR:TRIPODS grant CCF-1934568.

References

An, J., Ying, L., and Zhu, Y. Why resampling outperforms reweighting for correcting sampling bias with stochastic gradients. In *International Conference on Learning Representations*, 2020.

Arazo, E., Ortego, D., Albert, P., O'Connor, N. E., and McGuinness, K. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2020.

Aurelio, Y. S., de Almeida, G. M., de Castro, C. L., and Braga, A. P. Learning from imbalanced data sets with weighted cross-entropy function. *Neural Processing Letters*, 50(2):1937–1949, 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 *International Conference on Learning Representations (ICLR)*, 2020.

Branco, P., Torgo, L., and Ribeiro, R. P. A survey of predictive modeling on imbalanced domains. *ACM Computing Surveys (CSUR)*, 49(2):31, 2016.

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.

Busa-Fekete, R., Szörényi, B., Dembczynski, K., and Hüllermeier, E. Online F-measure optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 28, 2015.

Calderon-Ramirez, S., Moemeni, A., Elizondo, D., Colreavy-Donnelly, S., Chavarria-Estrada, L. F., Molina-Cabello, M. A., et al. Correcting data imbalance for semi-supervised covid-19 detection using x-ray chest images. *arXiv preprint arXiv:2008.08496*, 2020.

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

Chan, L., Hosseini, M. S., Rowsell, C., Plataniotis, K. N., and Damaskinos, S. Histosegnet: Semantic segmentation of histological tissue type in whole slide images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR)*, pp. 10662–10671, 2019.

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

Chen, W., Jiang, Z., Wang, Z., Cui, K., and Qian, X. Collaborative global-local networks for memory-efficient segmentation of ultra-high resolution images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8924–8933, 2019.

Coates, A., Ng, A., and Lee, H. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics (AISTATS)*, 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/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Dong, Q., Gong, S., and Zhu, X. Imbalanced deep learning by minority class incremental rectification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

Huang, C., Li, Y., Change Loy, C., and Tang, X. Learning deep representation for imbalanced classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Hyun, M., Jeong, J., and Kwak, N. Class-imbalanced semi-supervised learning. *arXiv preprint arXiv:2002.06815*, 2020.

Jamal, M. A., Brown, M., Yang, M.-H., Wang, L., and Gong, B. Rethinking class-balanced methods for long-tailed visual recognition from a domain adaptation perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7610–7619, 2020.

Japkowicz, N. The class imbalance problem: Significance and strategies. In *Proceedings of the International Conference on Artificial Intelligence (ICAI)*, 2000.

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

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 (NeurIPS)*, volume 33, pp. 14567–14579, 2020.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*, 2015.

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

Kubat, M. Addressing the curse of imbalanced training sets: One-sided selection. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 179–186, 1997.

Lai, Z., Wang, C., Oliveira, L. C., Dugger, B. N., Cheung, S.-C., and Chuah, C.-N. Joint semi-supervised and active learning for segmentation of gigapixel pathology images with cost-effective labeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 591–600, 2021.

Lee, H., Shin, S., and Kim, H. Abc: Auxiliary balanced classifier for class-imbalanced semi-supervised learning. *Advances in Neural Information Processing Systems*, 34, 2021.

Liu, J., Sun, Y., Han, C., Dou, Z., and Li, W. Deep representation learning on long-tailed data: A learnable embedding augmentation perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2970–2979, 2020.

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/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2537–2546, 2019.

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

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*, 2018.

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 (NeurIPS)*, pp. 3235–3246, 2018.

Oskal, K., Risdal, M., Janssen, E., Undersrud, E., and Gulsrud, T. A U-net based approach to epidermal tissue segmentation in whole slide histopathological images. *SN Appl. Sci.*, 1:672, 06 2019. doi: 10.1007/s42452-019-0694-y.

Pouyanfar, S., Tao, Y., Mohan, A., Tian, H., Kaseb, A. S., Gauen, K., Dailey, R., Aghajanzadeh, S., Lu, Y.-H., Chen, S.-C., et al. Dynamic sampling in convolutional neural networks for imbalanced data classification. In *2018 IEEE conference on multimedia information processing and retrieval (MIPR)*, pp. 112–117. IEEE, 2018.

Pulido, J. V., Guleria, S., Ehsan, L., Fasullo, M., Lippman, R., Mutha, P., Shah, T., Syed, S., and Brown, D. E. Semi-supervised classification of noisy, gigapixel histology images. In *2020 IEEE 20th International Conference on Bioinformatics and Bioengineering (BIBE)*, pp. 563–568. IEEE, 2020.

Rahman, M. A. and Wang, Y. Optimizing intersection-over-union in deep neural networks for image segmentation. In *International symposium on visual computing*, pp. 234–244. Springer, 2016.

Sohn, K., Berthelot, D., Carlini, N., Zhang, Z., Zhang, H., Raffel, C. A., Cubuk, E. D., Kurakin, A., and Li, C.-L. FixMatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in Neural Information Processing Systems (NeurIPS)*, 33, 2020.

Tai, K. S., Bailis, P., and Valiant, G. Sinkhorn label allocation: Semi-supervised classification via annealed self-training. In *International Conference on Machine Learning (ICML)*, 2021.

Tan, J., Wang, C., Li, B., Li, Q., Ouyang, W., Yin, C., and Yan, J. Equalization loss for long-tailed object recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11662–11671, 2020.

Tang, Z., Chuang, K. V., DeCarli, C., Jin, L.-W., Beckett, L., Keiser, M. J., and Dugger, B. N. Interpretable classification of alzheimer's disease pathologies with a convolutional neural network pipeline. *Nature communications*, 10(1):1–14, 2019.

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 (NeurIPS)*, 2017.

Van Horn, G., Mac Aodha, O., Song, Y., Cui, Y., Sun, C., Shepard, A., Adam, H., Perona, P., and Belongie, S. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Wang, Y.-X., Ramanan, D., and Hebert, M. Learning to model the tail. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Wei, C., Sohn, K., Mellina, C., Yuille, A., and Yang, F. CReST: A class-rebalancing self-training framework for imbalanced semi-supervised learning. *arXiv preprint arXiv:2102.09559*, 2021.

Yang, Y. and Xu, Z. Rethinking the value of labels for improving class-imbalanced learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 19290–19301, 2020.

Zhang, B., Wang, Y., Hou, W., Wu, H., Wang, J., Okumura, M., and Shinozaki, T. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. *Advances in Neural Information Processing Systems*, 34, 2021.

Zou, K. H., Warfield, S. K., Bharatha, A., Tempany, C. M., Kaus, M. R., Haker, S. J., Wells III, W. M., Jolesz, F. A., and Kikinis, R. Statistical validation of image segmentation quality based on a spatial overlap index1: scientific reports. *Academic radiology*, 11(2):178–189, 2004.

Appendix

A. Training details

A.1. Data source and compute resources

The three standard datasets we use are public:

- CIFAR-10 (Krizhevsky, 2009) can be downloaded at https://www.cs.toronto.edu/ kriz/cifar.html;
- CIFAR-100 (Krizhevsky, 2009) can be downloaded at https://www.cs.toronto.edu/ kriz/cifar.html;
- STL-10 (Coates et al., 2011) can be downloaded at https://cs.stanford.edu/ acoates/stl10/.

We run our experiments on GPUs including: Nvidia Titan XP, 2080Ti, Tesla T4, Titan RTX. We conduct all experiments under an environment with Ubuntu 18.04 (x86) as the OS, CUDA 10.0.130 and cuDNN 8 as the GPU drivers and libraries. Our mainly used framework includes PyTorch 1.1.0 and Torchvision 0.3.0.

A.2. Training settings

General settings. We adopt a Wide ResNet-28-2 (Oliver et al., 2018) as the encoder for all experiments in this work. We use Adam (Kingma & Ba, 2015) as our optimizer with the learning rate at 0.002. We also utilize an Exponential Moving Average (EMA) with the weight decay of 0.999 during the training process. Our baseline SSL models include the most recent state-of-the-art algorithms: FixMatch (Sohn et al., 2020), ReMixMatch (Berthelot et al., 2020). We train each setting with 500 epochs and have three trials with different random seeds.

FixMatch. We set the hyper-parameters of FixMatch by following (Sohn et al., 2020; Kim et al., 2020). The details are summarized in Table 5. The GitHub we refer to is available at https://github.com/google-research/fixmatch.

Table 5. FixMatch settings.

Hyper-parameter	Value
Confidence Threshold τ	0.95
Unlabeled loss weight λ_u	1
Ratio of unlabeled data in each mini-batch μ	2

ReMixMatch. We set the hyper-parameters of ReMixMatch by following (Berthelot et al., 2020; Kim et al., 2020). The details are summarized in Table 6. The GitHub we refer to is available at https://github.com/google-research/remixmatch.

Table 6. ReMixMatch settings.

Hyperparameter	Value
MixUp Beta Distribution α	0.75
Sharpening Temperature (T)	0.5
Weight on Unsupervised Loss λ_u	1.5
Weight on Rotation Loss λ_r	0.5
Weight on Un-Augmented Example λ_u	0.5

DARP. One of our baseline is the most recent imbalanced algorithm, DARP (Kim et al., 2020). We follow their settings strictly: α is set as 2.0, the number of warm-up epochs is set as 200, the number of iterations for DARP convergence is set as 10. The GitHub we refer to is available at https://github.com/bbuing9/DARP.

Our codes. To protect the confidentiality of double blind review, we will post our GitHub URL for all of our codes of SAW after the review process.

B. Additional Experimental Results

B.1. Distribution estimation

We also plot the class frequency during the training process to observe the role SAW plays in SSL algorithms. We set the imbalance ratio of CIFAR10-LT as $\gamma_l = \gamma_u = 50$ and use FixMatch as an example. The results are as shown in Figure 6: for the majority class, SAW is able to drive it back from biased prediction within a couple of epochs. For the minority class, SAW shows its focus and encourages the model to predict samples as miniorty class.

B.2. Stress-test

We also conduct stress-test experiments under the settings of $\gamma_l \neq \gamma_u$ on the test set of reversed distributions. The results are summarized in Table 7. SAW achieves significant improvements in all settings where the distribution of unlabeled data is different from that of labeled data. Specifically, SAW improves original FixMatch by up to 15.4% of bACC and 35.1% of GM; SAW also improves original ReMixMatch by up to 14.0% of bACC and 35.4% of GM. SAW also shows its superiority compared to DARP, especially considering SAW does not require any prior information from the unlabeled data.

On the other hand, we study the scenarios where the hold-out test set is imbalanced with the same distribution as the labeled dataset in the training set. The results are summarized in Table 9. We can see that all methods achieve good performance since the test set has the same distribution as the training set. However, CReST slightly hurts the original performance of SSL.

After that, we also study more severe scenarios: when the

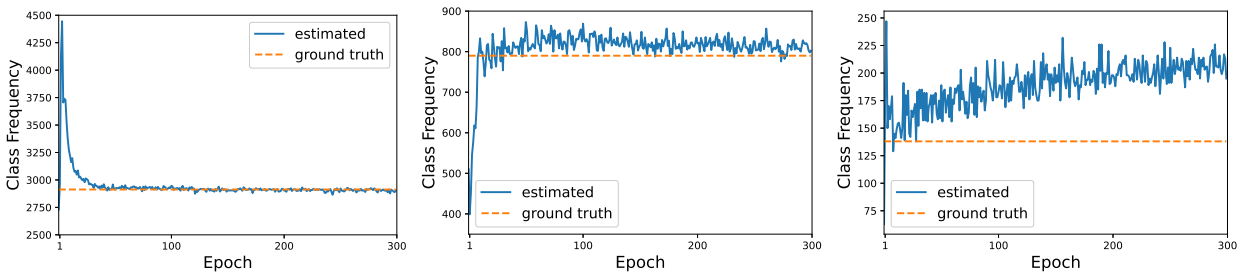


Figure 6. Estimation of distribution: class frequency of class 2 (a), class 5 (b), and class 9 (c).

Table 7. Comparison of classification performance on CIFAR10-LT under $\gamma_l = 100, \gamma_u \neq \gamma_l$ (hold-out test set is of reversed distributions). The evaluation criterion is bACC/GM. The best results are in bold.

	CIFAR-10 ($\gamma_l = 100$)		
Algorithm	$\gamma_u = 1$	$\gamma_u = 50$	$\gamma_u = 150$
ReMixMatch (Berthelot et al., 2020)	70.0$_{\pm 1.39}$ / 48.3$_{\pm 0.53}$	61.4$_{\pm 0.21}$ / 77.3$_{\pm 0.41}$	48.6$_{\pm 1.62}$ / 71.3$_{\pm 0.67}$
ReMixMatch* + DARP (Kim et al., 2020)	67.3$_{\pm 1.63}$ / 49.5$_{\pm 2.01}$	54.5$_{\pm 1.30}$ / 72.6$_{\pm 0.58}$	44.6$_{\pm 1.69}$ / 67.7$_{\pm 1.16}$
ReMixMatch* + SAW	**84.0**$_{\pm 0.05}$ / **83.7**$_{\pm 0.42}$	**72.7**$_{\pm 0.56}$ / **79.6**$_{\pm 1.11}$	**59.8**$_{\pm 1.03}$ / **73.6**$_{\pm 1.19}$
FixMatch (Sohn et al., 2020)	70.0$_{\pm 1.39}$ / 48.3$_{\pm 0.53}$	61.0$_{\pm 1.61}$ / 76.4$_{\pm 0.52}$	48.0$_{\pm 2.26}$ / 70.4$_{\pm 0.88}$
FixMatch + DARP (Kim et al., 2020)	68.1$_{\pm 0.37}$ / 48.5$_{\pm 1.46}$	59.5$_{\pm 1.41}$ / 75.2$_{\pm 1.08}$	55.1$_{\pm 1.31}$ / 73.8$_{\pm 0.43}$
FixMatch + SAW	**84.0**$_{\pm 0.39}$ / **83.4**$_{\pm 0.42}$	**76.4**$_{\pm 6.71}$ / **81.5**$_{\pm 1.97}$	**63.3**$_{\pm 0.37}$ / **75.4**$_{\pm 1.23}$

training labeled data is perfectly balanced, but the unlabeled data and hold-out data are imbalanced and flipped, we summarize the results in Table 8: we find DARP even hurts the original FixMatch while SAW still achieves performance gain consistently.

Table 8. Comparison of bACC on CIFAR10-LT under $\gamma_l = 1$ (labaled set is **balanced**) but $\gamma_u \neq \gamma_l$. The hold-out test set is of the **flipped** distribution of the training unlabeled set.

	CIFAR10-LT		
γ_u	50	100	150
FixMatch (Sohn et al., 2020)	77.8$_{\pm 0.36}$	66.8$_{\pm 1.51}$	59.9$_{\pm 0.43}$
FixMatch + DARP (Kim et al., 2020)	68.4$_{\pm 1.36}$	55.6$_{\pm 3.22}$	52.1$_{\pm 2.07}$
FixMatch + SAW	**82.4**$_{\pm 0.49}$	**75.2**$_{\pm 1.46}$	**70.1**$_{\pm 0.94}$

B.3. Per-class performance

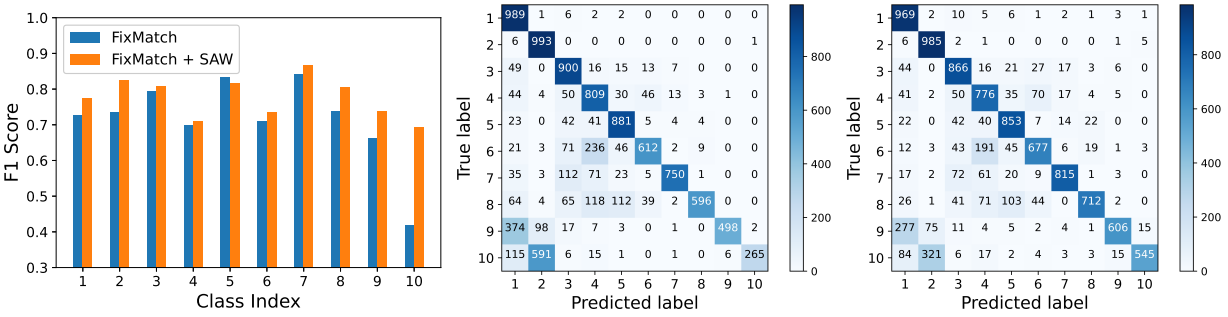
To show the source of our absolute performance gain, we present per-class F1 score (Busa-Fekete et al., 2015) on the balanced test set of CIFAR10-LT with $\gamma_l = \gamma_u = 100$ as shown in Figure 7(a). We find SAW improves the F1 score for almost every class, especially for minority classes (index as 8,9,10). Specifically, SAW provides gain on the precision of majority classes and the recall of minority classes without much penalty on other metrics. This explains again why

Table 9. Comparison of bACC on CIFAR10-LT under $\gamma = \gamma_l = \gamma_u$. The hold-out test set is of the **same** distribution of the training set (imbalanced).

	CIFAR10-LT		
Algorithm	$\gamma = 50$	$\gamma = 100$	$\gamma = 150$
FixMatch (Sohn et al., 2020)	89.2$_{\pm 0.59}$	85.4$_{\pm 0.59}$	83.1$_{\pm 0.33}$
FixMatch + DARP (Kim et al., 2020)	93.1$_{\pm 1.95}$	90.1$_{\pm 0.98}$	88.5$_{\pm 0.89}$
FixMatch + CReST (Wei et al., 2021)	86.7$_{\pm 1.98}$	84.3$_{\pm 1.20}$	82.4$_{\pm 1.46}$
FixMatch + SAW	**94.7**$_{\pm 1.23}$	**92.4**$_{\pm 0.35}$	**90.5**$_{\pm 0.47}$

SAW achieves significant gain in Table 4 where the distribution of hold-out test set is flipped based on unlabeled data' distribution (the minority class in the training set becomes the majority class in the test set). Figure 7 also shows the comparison of the confusion matrix of original FixMatch (Figure 7(b)) and SAW (Figure 7(c)) on the hold-out test set. Similar to the estimated distributions shown in Figure 1(b) and (c), we find that SAW provides robust re-weighting on both the minority and majority classes due to the smoothed weighting scheme, which is easily transferred to better generalization on a test set of totally different distributions.

We also compare the per-class performance during the training process between original FixMatch (Sohn et al., 2020)



(a) (b) (c)

Figure 7. Per-class results on the balanced test set of CIFAR-10 under the imbalanced ratio $\gamma = 100$. (a) F1 score for each class. (b) Confusion matrix of original FixMatch. (c) Confusion matrix of FixMatch with SAW.

and SAW. As shown in Figure 8, we plot the confusion matrix after training 100, 200, and 300 epochs, respectively. SAW improves the performance of minority classes quickly, even 100 epochs make a difference compared to the original FixMatch.

B.4. Alternative for smoothed weighting function

Besides the two mentioned in the main paper, we also studied

$$w_k \propto (n/n_k)^{\alpha}, \text{ for } k = 1, \dots, C, \tag{6}$$

as another smoothing weighting function. We use SAW_2 to denote using Equation 6 as the smoothing function. We use SAW_1 to denote using Equation 5. As shown in Table 10, SAW_2 achieves 83.2%, 77.3%, and 71.5% of bACC when $\gamma = 50, 100$, and 150, respectively in CIFAR 10-LT. These results are comparable to the results of the other two functions in Table 1. Hence we verify the necessity of smoothing weights to SSL.

Table 10. Comparison of classification performance on CIFAR10-LT under $\gamma = \gamma_l = \gamma_u$ (hold-out test set is balanced). The evaluation criterion is bACC.

	CIFAR10-LT		
Algorithm	$\gamma = 50$	$\gamma = 100$	$\gamma = 150$
FixMatch (Sohn et al., 2020)	79.2±0.33	71.5±0.72	68.4±0.15
FixMatch + DARP (Kim et al., 2020)	81.8±0.24	75.5±0.05	70.4±0.25
FixMatch + CReST (Wei et al., 2021)	83.0±0.39	75.7±0.38	70.8±0.25
FixMatch + SAW	84.0±0.10	77.5±0.63	71.6±0.35
FixMatch + SAW_1	83.7±0.18	78.2±0.21	72.5±0.67
FixMatch + SAW_2	83.2±0.48	77.3±0.17	71.5±0.53

B.5. Effect of adaptive weighting

To investigate the effect of adaptive updating, we conduct a sanity test: provide the true class distribution to the model by replacing the estimated distribution with the true distribution. Here we test the CIFAR10-LT where the labeled and unlabeled data share the same distribution. The true distribution of unlabeled data can be directly estimated from labeled data. To distinguish from SAW, we call this smoothed weighting without the adaptive scheme as SW. The results are summarized in Table 11. With the true distribution for estimating β, our SW improves FixMatch up to 4.7% on bACC when $\gamma = 100$. With the adaptive weighting scheme, our algorithm SAW further improves the performance, which proves the advantage of the adaptive weighting. SAW even achieves better performance compared to CReST+ (Wei et al., 2021) considering they have a strong pre-assumption that unlabeled data are sample from the same distributions as labeled data while SAW does not require any prior information from unlabeled data. In SW, where the weights are calculated based on the true distribution, the weight for each class is fixed and not adjustable during the training process. The performance will be affected by the degree of smoothing, i.e., the value of β. In SAW, the weight is based on the pseudo-label distribution, which can make the model adjustable and self-reinforcing. Hence SAW is less sensitive to the value of β and may perform better than SW when using non-optimal value of β. A similar observation can be seen in Table 12. With an optimal value of beta and a known distribution of unlabeled data, SW performs slightly better than SAW.

Table 11. Comparison of classification performance on CIFAR10-LT under $\gamma = \gamma_l = \gamma_u$ (hold-out test set is balanced). The evaluation criterion is bACC. The best results are in bold.

	CIFAR10-LT		
Algorithm	$\gamma = 50$	$\gamma = 100$	$\gamma = 150$
FixMatch (Sohn et al., 2020)	79.2$_{\pm 0.33}$	71.5$_{\pm 0.72}$	68.4$_{\pm 0.15}$
FixMatch + DARP (Kim et al., 2020)	81.8$_{\pm 0.24}$	75.5$_{\pm 0.05}$	70.4$_{\pm 0.25}$
FixMatch + CReST (Wei et al., 2021)	83.0$_{\pm 0.39}$	75.7$_{\pm 0.38}$	70.8$_{\pm 0.25}$
FixMatch + CReST+ (Wei et al., 2021)	83.9$_{\pm 0.14}$	77.4$_{\pm 0.36}$	**72.8**$_{\pm 0.58}$
FixMatch + SW	82.7$_{\pm 0.58}$	76.2$_{\pm 0.06}$	70.8$_{\pm 0.26}$
FixMatch + SAW	**84.0**$_{\pm 0.10}$	**77.5**$_{\pm 0.63}$	71.6$_{\pm 0.35}$

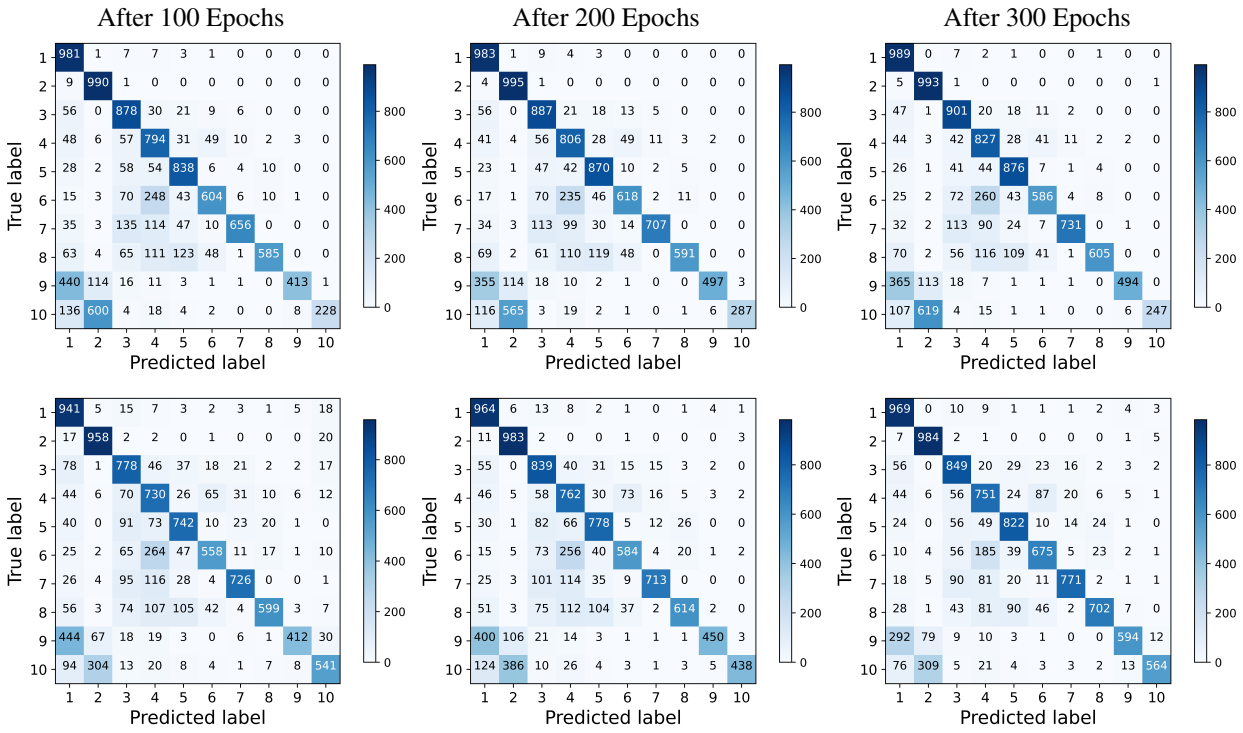


Figure 8. The confusion matrix comparison between original FixMatch (Sohn et al., 2020) and FixMatch + SAW during the training progress. The top row is from original FixMatch while the bottom row denotes FixMatch + SAW.

Table 12. Comparison of classification performance on CIFAR10-LT under $\gamma = 100$ (hold-out test set is of balanced) with a fixed $\beta = 0.99, 0.999, 0.9999$. The evaluation criterion is bACC/GM. The best results are in bold.

Algorithm	$\beta = 0.99$	$\beta = 0.999$	$\beta = 0.9999$
FixMatch + SW	75.0$_{\pm 0.67}$ / 73.0$_{\pm 0.58}$	**77.5**$_{\pm 0.98}$ / **76.3**$_{\pm 1.21}$	72.4$_{\pm 1.69}$ / 70.7$_{\pm 2.16}$
FixMatch + SAW	77.3$_{\pm 0.67}$ / 75.4$_{\pm 0.88}$	**77.4**$_{\pm 0.30}$ / **76.2**$_{\pm 0.40}$	73.7$_{\pm 0.63}$ / 72.3$_{\pm 0.93}$

Table 13. Comparison of classification performance on CIFAR10-LT under $\gamma = \gamma_l = \gamma_u$ (hold-out test set is balanced). The evaluation criterion is bACC. SL_W denotes only applying smoothed weights to the supervised loss on labeled data, CL_W denotes only applying smoothed weights to the consistency loss.

	CIFAR10-LT		
Algorithm	$\gamma = 50$	$\gamma = 100$	$\gamma = 150$
FixMatch (Sohn et al., 2020)	79.2$_{\pm 0.33}$	71.5$_{\pm 0.72}$	68.4$_{\pm 0.15}$
FixMatch + SL_W	79.9$_{\pm 0.05}$	72.2$_{\pm 0.89}$	68.9$_{\pm 0.33}$
FixMatch + CL_W	82.5$_{\pm 0.30}$	76.3$_{\pm 0.21}$	70.4$_{\pm 0.19}$
FixMatch + SAW	84.0$_{\pm 0.10}$	77.5$_{\pm 0.63}$	71.6$_{\pm 0.35}$

B.6. Effect of weights on supervised loss

The motivation on adding the weight in consistency loss is to alleviate the bias in pseudo labels during the class-imbalanced training. We have studied the function of weighting in both supervised loss and consistency loss. We conclude that applying smoothed weights to both loss items can result in the best performance based on our experiments. Table 13 summarizes the results: the weights on consistency loss are important, which is expected considering the fact that the majority samples are unlabeled in SSL settings. We also find that applying weights to the supervised loss can further improve the performance. Thus we decide to apply smoothed weights to both loss items, which is SAW.

B.7. Effect of the parameter β

As discussed in Methodology, we heuristically select $\beta = (N-1)/N$, where N is updated after each epoch via $N = \sum_{k=1}^{C}(m_k + \hat{n}_k)/C$. Here a sensitivity analysis on the parameter β is provided. Similarly, we consider CIFAR10-LT under $\gamma = 100$, and test both SW and SAW with a fixed $\beta \in \{0.99, 0.999, 0.9999\}$ in the training process. Table 12 shows that the trend is similar to the weighting function of ν, indicating the effectiveness of smoothed weighting in SSL. Moreover, SAW is less sensitive to the value of β. Therefore, there is no need to specifically tune the parameter β for each setting. The heuristic selection in SAW already brings the

optimal result among these trials. Hence $\beta = (N-1)/N$ is a parameter determining the degree of smoothing. In reality, this parameter needs to be tuned as the data volume N is unknown. However, the value of β is only sensitive to the scale of N. We only provided a reference value of N, which is the average number of samples per class. This can be regarded as an approximate value to the optimal one. When applying the proposed method to a new dataset, researchers can fine-tune the value of beta from the initial reference one.

B.8. Additional evaluation on a pathology application

Besides three standard datasets (CIFAR-10, CIFAR-100, and STL-10), we also select a gigapixel pathology dataset (Lai et al., 2021) as an additional real-world case to further evaluate the reliability of SAW in realistic applications. This main goal of this dataset is to segment white matter (WM) from grey matter (GM). We follow the way in (Lai et al., 2021) to use only 0.1% of regions as the labeled data and treat the remaining areas as unlabeled. This is a common case in medical imaging applications where the distribution of unlabeled data is unavailable as the gigapixel resolution requires massive annotation efforts.

Table 14. IoU score comparison on the 10 hold-out slides.

Algorithm	GM	WM
FixMatch	80.7	43.0
FixMatch + SAW	**84.6**	**65.4**

Table 15. DICE coefficient comparison on the 10 hold-out slides.

Algorithm	GM	WM
FixMatch	89.4	61.3
FixMatch + SAW	**91.0**	**77.2**

Table 14 and Table 15 summarizes the results of two measuring metrics on the hold-out test set: IoU score (Rahman & Wang, 2016) and DICE coefficient (Zou et al., 2004). For the minority class - WM, SAW improves the IoU score (Rahman & Wang, 2016) by over 22%; for the majority class - GM, it also achieves nearly 4% of improvement. For the DICE coefficient, another metric, SAW brings up to 15.9% of improvement on the minority class and 1.6% on the majority class. From this case, we verify the reliability of SAW on a real-world application when the distribution information is unknown.