CoSSL: Co-Learning of Representation and Classifier for Imbalanced Semi-Supervised Learning

Yue Fan Dengxin Dai Bernt Schiele
{yfan, ddai, schiele}@mpi-inf.mpg.de
Max Planck Institute for Informatics, Saarbrücken, Germany
Saarland Informatics Campus

Abstract

Standard semi-supervised learning (SSL) using class-balanced datasets has shown great progress to leverage unlabeled data effectively. However, the more realistic setting of class-imbalanced data – called imbalanced SSL – is largely underexplored and standard SSL tends to underperform. In this paper, we propose a novel co-learning framework (CoSSL) with decoupled representation learning and classifier learning for imbalanced SSL. To handle the data imbalance, we devise Tail-class Feature Enhancement (TFE) for classifier learning. Furthermore, the current evaluation protocol for imbalanced SSL focuses only on balanced test sets, which has limited practicality in real-world scenarios. Therefore, we further conduct a comprehensive evaluation under various shifted test distributions. In experiments, we show that our approach outperforms other methods over a large range of shifted distributions, achieving state-of-the-art performance on benchmark datasets ranging from CIFAR-10, CIFAR-100, ImageNet, to Food-101. Our code will be made publicly available.

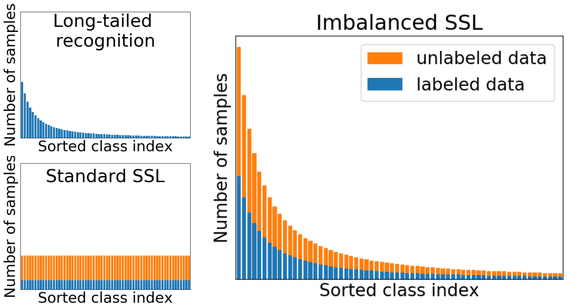

Figure 1. Conventional recognition tasks focus on constrained settings: long-tailed recognition does not involve unlabeled data; semi-supervised learning (SSL) assumes class-balanced distributions for both labeled and unlabeled data. In this work, we aim at *imbalanced SSL*, where the training data is partially annotated, and both labeled and unlabeled data are not manually balanced. This setting is more general and poses great challenges to existing algorithms. A robust learning algorithm should still be able to learn a good classifier under this setting.

1. Introduction

Imbalanced data distributions are ubiquitous, and pose great challenges for standard deep learning methods. Many approaches have been proposed for long-tailed recognition, where the number of (labeled) examples exhibits a long-tailed distribution with heavy class imbalance [17, 18, 33, 36, 38, 51]. While semi-supervised learning (SSL) in the class-balanced setting has shown great promise, in this paper we are interested in the challenging and realistic setting of *imbalanced SSL* where both the labeled and the unlabeled data are class-imbalanced, as shown in Fig. 1.

Despite a few pioneer works [30, 53], existing solutions from long-tailed recognition and SSL do not generalize well to this setting. On the one hand, long-tailed recognition [6, 9, 19, 20, 24] is not designed to utilize unlabeled data despite being good at handling data imbalance. Semi-supervised learning (SSL) [1, 3, 4, 34, 41, 45, 47–49], on the other hand, can effectively leverage unlabeled data but can not address data imbalance. In certain cases, standard SSL methods trained with imbalanced unlabeled datasets can lead to even worse results than a simple re-balancing method without using any unlabeled data [30], which counters the promise of SSL.

In this paper, we address the imbalanced SSL problem by leveraging strong SSL algorithms [3,4,49,54] and recent success of decoupling representation and classifier learning from long-tailed recognition [29]. To this end, we propose CoSSL, a novel framework for imbalanced SSL, which consists of semi-supervised representation learning, classifier learning, and pseudo-label generation as shown in Fig. 2. For the representation learning module, the flexibility of our framework allows us to leverage top-performing SSL algorithms, and we experiment with various SSL methods

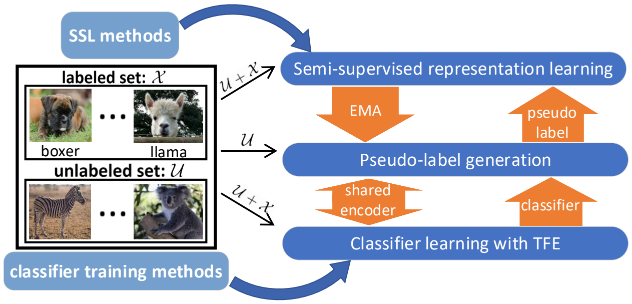
SSL methods
labeled set: $\mathcal{X}$
boxer
llama
unlabeled set: $\mathcal{U}$
classifier training methods
$\mathcal{U}+\mathcal{X}$
$\mathcal{U}$
$\mathcal{U}+\mathcal{X}$
Semi-supervised representation learning
EMA
pseudo label
Pseudo-label generation
shared encoder
classifier
Classifier learning with TFE

Figure 2. We propose a flexible co-learning framework for imbalanced SSL, which separates representation learning and classifier learning while connecting them via EMA and pseudo-labeling. The representation learning module provides a momentum encoder for feature generation of classifier learning, and the improved classifier, in turn, produces pseudo-labels for representation learning. Our method can be used as a drop-in replacement of the pseudo-label generation for any SSL algorithms.

in this paper. For classifier learning, we propose a novel Tail-class Feature Enhancement (TFE) to handle the data imbalance during the classifier training by enhancing the data diversity of the tail classes. These two modules are seamlessly connected via a shared feature encoder and the pseudo-labeling module to improve each other, outperforming previous state-of-the-art methods by a large margin, especially in the case of severe imbalance.

Furthermore, the standard evaluation protocol of long-tailed recognition and SSL normally assumes that the test data are from a uniform class distribution [3, 4, 7, 29, 35, 40, 49, 50, 52]. However, this is insufficient to reflect the diversity of real-world applications, where users may have different needs. It is strongly desired that the trained model can perform well over a large range of varying distributions, including those that are radically different from the training distribution. Therefore, in this paper, we adopt the shifted evaluation from [23], where the test data are from variously shifted class distributions. We further distinguish between unknown shifted evaluation and known shifted evaluation, depending on whether test distribution is known a *priori* during training. This evaluation protocol can be used for long-tailed recognition as well.

Overall, our contributions are: (1) We propose the flexible co-learning framework CoSSL for imbalanced SSL, which consists of a semi-supervised representation learning module, a novel Tail-class Feature Enhancement (TFE) module for better classifier learning, and a carefully designed pseudo-label generation module, which enables the interplay between the representation and the classifier and, thus, leads to improve generalization. (2) We propose new evaluation criteria for imbalanced SSL and conduct a comprehensive evaluation with them. (3) We achieve new state-of-the-art results on multiple imbalanced SSL benchmarks across a wide range of evaluation settings.

2. Related work

Semi-supervised learning. Many efforts have been made in various directions in SSL. For example, many recent powerful methods [1, 45, 47] are based on consistency regularization, where the idea is that the model should output consistent predictions for perturbed versions of the same input. Another spectrum of popular approaches is pseudo-labeling [34, 41, 48] or self-training [46], where the model is trained with artificial labels. Furthermore, there are many excellent works around generative models [15, 32, 42] and graph-based methods [2, 28, 37, 39]. A more comprehensive introduction of SSL methods is available in [8, 57, 58]. However, none of the aforementioned works have studied SSL in the class-imbalanced setting, in which the standard SSL methods fail to generalize well.

Long-tailed recognition. Research on class-imbalanced supervised learning has attracted increasing attention. In particular, many recent efforts have been made to improve the performance under imbalanced data by decoupling the learning of representation and classifier head [29, 35, 40, 50, 52]. In the two-stage framework from [29], an instance-balanced sampling scheme was first used for representation learning. In the second stage, the classifier head is simply retrained by a class-balanced sampling. We found that this scheme is also very competitive for imbalanced SSL in our preliminary experiments. In contrast to this line of works, our co-learning framework focuses on imbalanced SSL and largely simplifies the training pipeline compared to the two-stage framework [29]. The joint training enables interaction between representation learning and classifier learning, which brings additional benefits to the final performance. Evaluation under shifted distributions was also proposed by [23], however, we take a step further and consider settings where the test-time distribution is given or not as prior knowledge during the training.

Imbalanced semi-supervised learning. While SSL has been extensively studied, the setting of class-imbalanced semi-supervised is rather under-explored. Most successful methods from standard SSL do not generalize well to this more realistic scenario without addressing the data imbalance explicitly. Hyun et al. [26] proposed a suppressed consistency loss to suppress the loss on minority classes. Kim et al. [30] proposed Distribution Aligning Refinery (DARP) to refine raw pseudo-labels via convex optimization. Wei et al. [53] found that the raw SSL methods usually have high recall and low precision for head classes while the reverse is true for the tail classes and further proposed a reverse sampling method for unlabeled data based on that. However, previous work cannot fully address the bias of pseudo-label toward the head class. In this aspect, we devise a novel co-learning method that separates representation learning and classifier learning in a joint training framework.

3. CoSSL: Co-learning for imbalanced SSL

In this section, we first present the problem setup of imbalanced semi-supervised learning (SSL). Based on this, we introduce CoSSL, a flexible co-learning framework for imbalanced SSL in Section 3.1.

Problem setup and notations: For a K-class classification problem, there is a labeled set $\mathcal{X} = \{(\mathbf{x}_n, y_n) : n \in (1, ..., N)\}$ and an unlabeled set $\mathcal{U} = \{\mathbf{u}_m : m \in (1, ..., M)\}$, where $\mathbf{x}_n, \mathbf{u}_m \in \mathbb{R}^d$ are training examples and $y_n \in \{1, ..., K\}$ are class labels for labeled examples. N_k and M_k denote the numbers of labeled and unlabeled examples in class 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 the classes are sorted by the number of training samples in descending order, i.e., $N_1 \geq N_2 \geq ... \geq N_k$. The goal of imbalanced SSL is to train a classifier $f : \mathbb{R}^d \to \{1, ..., K\}$ that generalizes well over a large range of varying test data distributions.

3.1. Co-learning representation and classifier

The two-stage framework [29, 35, 40, 50, 52] from long-tailed recognition is quite successful for supervised learning with imbalanced data. It decouples representation and classifier by retraining a classifier after the representation learning. While classifier re-training (cRT) [29] is out-of-the-box a strong starting point, as we will see in the experimental section 4.1, the method has its own limitations when applied to imbalanced SSL: (1) unlabeled data is not utilized during cRT; (2) the two-stage training scheme makes it impossible to refine the pseudo-labels, which in turn limits the quality of feature representation learning.

This motivates us to propose CoSSL, a flexible co-learning framework for imbalanced SSL. As illustrated in Fig. 3, CoSSL consists of three modules: a semi-supervised representation learning module, a classifier learning module, and a pseudo-label generation module. The feature encoder from the representation learning module is shared with the classifier module to learn a better classifier, and the improved classifier is used to generate better pseudo-labels for the representation learning module to further improve the feature encoder. This joint framework largely simplifies the training pipeline compared to the two-stage framework and enables interaction between the representation learning and the classifier learning, which brings additional benefits to the final performance (see Section 4.5 for ablation).

Semi-supervised representation learning: The goal of the semi-supervised representation learning module is to obtain a strong feature encoder by exploring unlabeled data. Thanks to the flexibility of our framework, we can use and evaluate a variety of SSL methods [3, 4, 49]. Given a batch of unlabeled data sampled from the random sampler, we first pass the unlabeled data to the pseudo-label generation module. Then, the unlabeled data loss is computed using the generated pseudo-labels. Meanwhile, a batch of labeled data is sampled by the random sampler, and the labeled data loss is computed. The resulting encoder is accumulated into a momentum encoder and further passed to the classifier module for feature extraction to enhance the classifier training as shown in Fig. 3.

Classifier learning with Tail-class Feature Enhancement: Inspired by the success of cRT, we train a separate classifier in the classifier learning module and aim to further improve it by using unlabeled data. To this end, we propose Tail-class Feature Enhancement (TFE) that exploits unlabeled data by blending unlabeled data features with labeled data features while preserving the label of the labeled sample. Specifically, at each training step, we train the classifier using blended features between labeled and unlabeled data with labels from labeled data. We deploy a class-balanced sampler and a random sampler to sample a labeled example $(\mathbf{x}_i, y_i)$ and an unlabeled example $\mathbf{u}_j$. Then the new fused feature for classifier training is generated by:

$$\tilde{\mathbf{z}} = \lambda \xi(\mathbf{x}_i) + (1 - \lambda)\xi(\mathbf{u}_j) \tag{1}$$

$$\tilde{y} = y_i \tag{2}$$

where ξ is the momentum encoder from the representation learning module and the fusion factor λ is sampled from a uniform distribution over the interval $[\mu, 1]$. In practice we set λ close to 1 to ensure that $\tilde{\mathbf{z}}$ is closer to the labeled feature $\xi(\mathbf{x}_i)$, and therefore we can safely use the label y_i for the synthesized sample.

To enhance the data diversity of tail classes, we train the classifier using different portions of fused examples in a stochastic way. The feature blending is applied with a blend probability that depends on the number of data for each class so that the more labeled data a class has, the less fused data is synthesized for classifier learning. Formally, given a labeled example from class k, we apply feature blending with probability P_k defined as:

$$P_k = \frac{N_1 - N_k}{N_1} \tag{3}$$

where N_k is the number of examples from the k-th class, and N_1 is the number of examples of the first class (with the most labeled data). Such a class-dependent blend probability encourages more augmented data from feature blending for tail classes, thus, improving the data diversity of tail classes. For instance, there is no fused data for the first class, which has the most labeled data, since $P_1 = 0$. For a tail class with only 5% samples of the first class, the blend probability will be as high as 95%. Note, that since fused

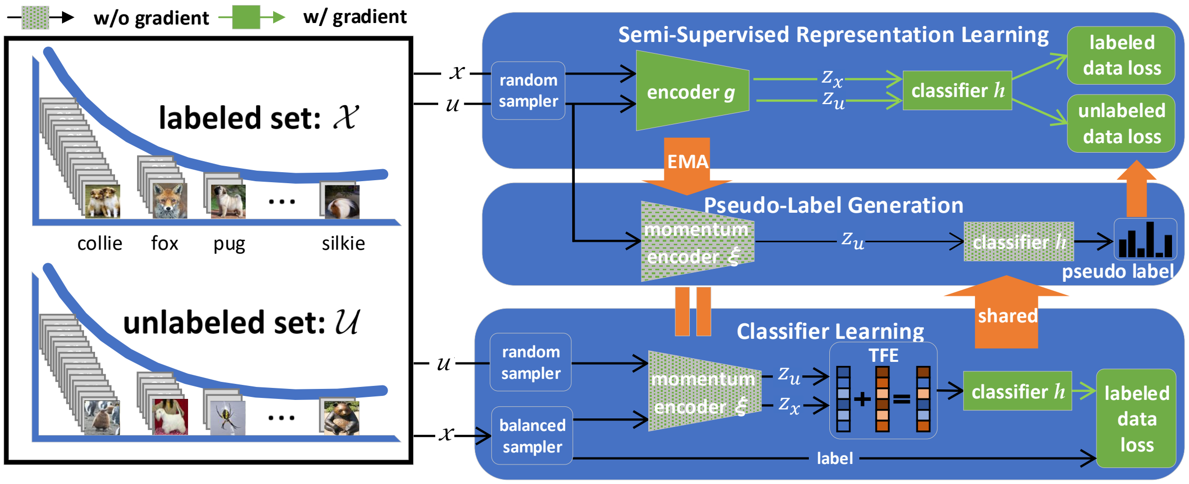

Figure 3. Our co-learning framework CoSSL for imbalanced SSL consists of three modules: a semi-supervised representation learning module, a balanced classifier learning module, and a carefully designed pseudo-label generation module. The representation module provides a momentum encoder for feature extraction in the other two modules, and the classifier module produces a balanced classifier using our novel Tail-class Feature Enhancement (TFE). Then, pseudo-label module generates pseudo-labels for the representation module using the momentum encoder and the balanced classifier. The interplay between these modules enhances each other, leading to both a more powerful representation and a more balanced classifier. Additionally, our framework is flexible as it can accommodate any standard SSL methods and classifier learning methods.

Algorithm 1 Tail-class Feature Enhancement

```
Input: Labeled set $\mathcal{X}$, unlabeled set $\mathcal{U}$, feature encoder $\xi$, parameter $\mu$, and batch size $B$
for $b = 1$ to $B$ do
    // Sample labeled and unlabeled examples
    $\mathbf{x}_i, y_i \sim$ Class-balanced sampler($\mathcal{X}$)
    $\mathbf{u}_j \sim$ Random sampler($\mathcal{U}$)
    $P_{y_i} = \frac{N_1 - N_{y_i}}{N_1}$ // Compute the blend probability
    if Uniform$(0, 1) \leq P_{y_i}$ then
        // Generate features by feature blending
        $\lambda \sim$ Uniform$(\mu, 1)$
        $\tilde{\mathbf{z}}_b = \lambda\xi(\mathbf{x}_i) + (1 - \lambda)\xi(\mathbf{u}_j)$
        $\tilde{y}_b = y_i$
    else
        // Use features of labeled data directly
        $\tilde{\mathbf{z}}_b = \xi(\mathbf{x}_i)$
        $\tilde{y}_b = y_i$
    end if
end for
return $\{\tilde{\mathbf{z}}, \tilde{y}\}$ // Features for classifier training
```

data share the same label with the labeled data, the class distribution is uniform during cRT as the labeled set is sampled using a class-balanced sampler. Pseudo-code for processing a batch of labeled and unlabeled examples can be found in Alg. 1.

Pseudo-label generation: As standard SSL methods suffer from biased pseudo-labels under data imbalance [30, 53], we devise a pseudo-label generation module to generate high-quality pseudo-labels by combining the strengths of the representation learning module and the classifier learning module. Given a batch of unlabeled data, it first uses the momentum encoder ξ from the representation learning to extract features since the representations learned from instance-balanced sampling from SSL is the most generalizable [29]. Then the pseudo-labels are predicted using the classifier trained from TFE leveraging its robustness against data imbalance. Our pseudo-label generation module replaces the original pseudo-labeling part of the SSL algorithm in the representation learning module and enables the trained classifier to enhance representation learning. Note, that no gradient updates happen at this step.

Overall co-learning framework: The three aforementioned modules interact with each other via a shared feature encoder and pseudo-labeling. CoSSL can then bootstrap itself by exchanging information between them: the representation learning module provides a momentum encoder for better feature extraction for training classifiers and pseudo-labeling. And the improved classifier, in turn, generates high-quality pseudo-labels to further enhance representation learning. Specifically, denote the overall network by f, which consists of a feature extractor network $g(\cdot)$ and a classifier head $h(\cdot)$. At training iteration t, the three modules operate successively as shown in Fig. 3. (1) For the classifier module, a batch of labeled data and unlabeled data from $\mathcal{X}$ and $\mathcal{U}$ are sampled using a class-balanced sampler and a random sampler, respectively. Then, the features are extracted by a momentum encoder $\xi(\cdot)$ of $g(\cdot)$, which is provided by the representation learning module. We update ξ by $\xi_t = m\xi_{t-1} + (1-m)g_t$ where $\xi_0 = g_0$ and $m \in [0,1)$ is a momentum coefficient. Then, the classifier h is trained using our TFE with standard cross-entropy loss. (2) For the pseudo-label generation module, it encodes a new batch of unlabeled data with the same momentum encoder ξ_t and predicts the pseudo-labels using the classifier h from the classifier module. (3) The generated pseudo-labels are then fed into the representation module to compute the unlabeled data loss. Meanwhile, a new batch of labeled data is used in the representation module.

CoSSL fits particularly well for imbalanced SSL as the representation module and the classifier module, despite being decoupled, can enhance each other via pseudo-labeling and the momentum encoder, leading to both a more powerful representation and a more balanced classifier. Moreover, our co-learning framework is very flexible as it can accommodate any standard SSL algorithm and classifier learning method, which makes it possible to benefit from the most advanced approaches. We present the complete algorithm of our co-learning framework in the Appendix.

	CIFAR-10-LT		
	γ=50	γ=100	γ=150
vanilla	$65.2^{*}_{\pm0.05}$	$58.8^{*}_{\pm0.13}$	$55.6^{*}_{\pm0.43}$
Long-tailed recognition methods			
w/ Re-sampling [27]	$64.3^{*}_{\pm0.48}$	$55.8^{*}_{\pm0.47}$	$52.2^{*}_{\pm0.05}$
w/ LDAM-DRW [7]	$68.9^{*}_{\pm0.07}$	$62.8^{*}_{\pm0.17}$	$57.9^{*}_{\pm0.20}$
w/ cRT [29]	$67.8^{*}_{\pm0.13}$	$63.2^{*}_{\pm0.45}$	$59.3^{*}_{\pm0.10}$
SSL methods			
MixMatch [4]	$73.2^{*}_{\pm0.56}$	$64.8^{*}_{\pm0.28}$	$62.5^{*}_{\pm0.31}$
w/ DARP [30]	$75.2^{*}_{\pm0.47}$	$67.9^{*}_{\pm0.14}$	$65.8^{*}_{\pm0.52}$
w/ CReST+ [53]	$79.0^{*}_{\pm0.26}$	$71.9^{*}_{\pm0.33}$	$68.3^{*}_{\pm0.57}$
w/ CoSSL	**80.31**$_{\pm0.31}$	**76.4**$_{\pm1.14}$	**73.5**$_{\pm1.25}$
ReMixMatch [3]	$81.5^{*}_{\pm0.26}$	$73.8^{*}_{\pm0.38}$	$69.9^{*}_{\pm0.47}$
w/ DARP [30]	$82.1^{*}_{\pm0.14}$	$75.8^{*}_{\pm0.09}$	$71.0^{*}_{\pm0.27}$
w/ CReST+ [53]	$83.7_{\pm0.15}$	$78.8_{\pm0.54}$	$75.2_{\pm0.30}$
w/ CoSSL	**87.7**$_{\pm0.21}$	**84.1**$_{\pm0.56}$	**81.3**$_{\pm0.83}$
FixMatch [49]	$79.2^{*}_{\pm0.33}$	$71.5^{*}_{\pm0.72}$	$68.4^{*}_{\pm0.15}$
w/ DARP [30]	$81.8^{*}_{\pm0.24}$	$75.5^{*}_{\pm0.04}$	$70.4^{*}_{\pm0.25}$
w/ CReST+ [53]	$83.9^{*}_{\pm0.14}$	$77.4^{*}_{\pm0.36}$	$72.8^{*}_{\pm0.58}$
w/ CoSSL	**86.8**$_{\pm0.30}$	**83.2**$_{\pm0.49}$	**80.3**$_{\pm0.55}$

Table 1. Classification accuracy (%) on CIFAR-10-LT using a Wide ResNet-28-2 under the uniform test distribution of three different class-imbalance ratios γ. The numbers are averaged over 5 different folds. We use the same code base as [30] for fair comparison following [44]. Numbers with * are taken from the original papers. The best number is in bold.

4. Experimental evaluation

In this section, we conduct extensive experiments to evaluate the efficacy of our framework. In Section 4.1, 4.2, and 4.3, we compare our method with existing works and show that we achieve state-of-the-art performance for the commonly used uniform test evaluation. Section 4.4 evaluates different methods over a large range of imbalance settings, and we distinguish between two cases: the distributions are unknown or known a priori during training. A detailed analysis of our framework can be found in Section 4.5.

4.1. Main results on CIFAR-10 and CIFAR-100

Datasets. Following common practice [7, 13], we employ CIFAR10-LT and CIFAR100-LT for imbalanced SSL by randomly selecting some training images for each class determined by a pre-defined imbalance ratio γ as the labeled and the unlabeled set. Specifically, we set $N_k = N_1 \cdot \gamma^{-\frac{k-1}{K-1}}$ for labeled data and $M_k = M_1 \cdot \gamma^{-\frac{k-1}{K-1}}$ for unlabeled data. We use $N_1 = 1500$; $M_1 = 3000$ for CIFAR-10 and $N_1 = 150$; $M_1 = 300$ for CIFAR-100, respectively. Following [30, 53], we report results with imbalance ratio $\gamma = 50$, 100 and 150 for CIFAR10-LT and $\gamma = 20$, 50 and 100 for CIFAR100-LT. Therefore, the number of labeled samples for the least class is 10 and 1 for CIFAR-10 with $\gamma = 150$ and CIFAR-100 with $\gamma = 100$, respectively.

Setup. Following [11, 30], we evaluate our method with MixMatch [4], ReMixMatch [3], and FixMatch [49] under the same implementation (as recommended by [44]) using Wide ResNet-28-2 [55] as the backbone. The hyper-parameter μ in Alg. 1 is set to 0.6 based on the ablation study in Section 4.5. We apply TFE module in the last 20% of iterations for faster training and better accuracy (see Appendix for more details). As our implementation is based on the public codebase from [30], we use the same hyper-parameters as theirs. For example, all experiments are trained with batch size 64 using Adam optimizer [31] with a constant learning rate of 0.002 without any decay. We train all models for 500 epochs, each of which has 500 steps, resulting in a total number of 2.5×10^5 training iterations. For all experiments, we report the average test accuracy of the last 20 epochs following [44]. For CReST+, we use the official TensorFlow implementation. As for data augmentation for TFE, we use the strong augmentation from [49], which consists of RandAugment [12] and CutOut [16].

Results. Table 1 and Table 2 compare our method with various SSL algorithms and long-tailed recognition algorithms on CIFAR-10-LT and CIFAR-100-LT with various imbalance ratios γ. Our method achieves the best performance across all settings with significant margins over the previous state-of-the-art. Noticeably, our method is particularly

	CIFAR-100-LT		
	γ=20	γ=50	γ=100
ReMixMatch [3]	$51.6_{\pm 0.43}$	$44.2_{\pm 0.59}$	$39.3_{\pm 0.43}$
w/ DARP [30]	$51.9_{\pm 0.35}$	$44.7_{\pm 0.66}$	$39.8_{\pm 0.53}$
w/ CReST+ [53]	$51.3_{\pm 0.34}$	$45.5_{\pm 0.76}$	$41.0_{\pm 0.78}$
w/ CoSSL	$\mathbf{55.8}_{\pm 0.62}$	$\mathbf{48.9}_{\pm 0.61}$	$\mathbf{44.1}_{\pm 0.59}$
FixMatch [49]	$49.6_{\pm 0.78}$	$42.1_{\pm 0.33}$	$37.6_{\pm 0.48}$
w/ DARP [30]	$50.8_{\pm 0.77}$	$43.1_{\pm 0.54}$	$38.3_{\pm 0.47}$
w/ CReST+ [53]	$51.8_{\pm 0.12}$	$44.9_{\pm 0.50}$	$40.1_{\pm 0.65}$
w/ CoSSL	$\mathbf{53.9}_{\pm 0.78}$	$\mathbf{47.6}_{\pm 0.57}$	$\mathbf{43.0}_{\pm 0.61}$

Table 2. Classification accuracy (%) on CIFAR-100-LT under the uniform test distribution of three different class-imbalance ratios γ. The numbers are averaged over 5 different folds. We reproduce all numbers using the same codebase from [30] for a fair comparison[1]. The best number is in bold.

good at larger imbalance ratios. For example, we outperform the second-best by an absolute accuracy of 7.5% on CIFAR-10-LT at imbalance ratio $\gamma = 150$ with FixMatch, which underlines the superiority of our method. Replacing MixMatch with ReMixMatch or FixMatch as the representation learning module can increase test accuracy on CIFAR-10-LT at imbalance ratio $\gamma = 150$ by 7.8% and 6.8%, respectively. On CIFAR-100-LT, we evaluate our method on top of ReMixMatch and FixMatch as they give the best performance on CIFAR-10-LT. Besides the best performance across settings, our method also improves performance for small imbalance ratios as well (4.5% higher than the second-best at imbalance ratio $\gamma = 20$ with ReMixMatch).

Comparison with concurrent works. There are other two concurrent works [21, 43]. Since their official code is not available, here we only compare the results under the common settings on CIFAR datasets. As is shown in Table 3, our method shows better generalization performance than these concurrent works across different settings. We use the numbers reported in their papers directly so there is no standard deviation. Nevertheless, the results can still show that our method clearly outperforms theirs.

	CIFAR-10-LT		CIFAR-100-LT
	γ=100	γ=150	γ=20
FixMatch [49]	$71.5^{*}_{\pm 0.72}$	$68.4^{*}_{\pm 0.15}$	$49.6_{\pm 0.78}$
w/ BiS [21]	81.0^{*}	76.9^{*}	50.6^{*}
w/ DASO [43]	79.1^{*}	75.1^{*}	52.9^{*}
w/ CoSSL	$\mathbf{83.2}_{\pm 0.49}$	$\mathbf{80.3}_{\pm 0.55}$	$\mathbf{53.9}_{\pm 0.78}$

Table 3. Classification accuracy (%) on CIFAR-LT under the uniform test distribution. Since the official code for BiS and DASO is not available, here we only compare our method with theirs under the same settings where the results are available. Numbers with * are taken from the original papers. The best number is in bold.

4.2. Main results on Small-ImageNet-127

Dataset. ImageNet127 is originally introduced in [25] and used by [53] for imbalanced SSL. It is a naturally imbalanced dataset with imbalance ratio $\gamma \approx 286$ by grouping the 1000 classes of ImageNet [14] into 127 classes based on the WordNet hierarchy. Due to limited resources, we are not able to conduct experiments on ImageNet127 with the full resolution[2]. Instead, we propose a down-sampled version of ImageNet127 to test the effectiveness of our method on a large-scale dataset. Inspired by [10], we down-sample the original images from ImageNet127 to smaller images of 32×32 or 64×64 pixels using the box method from Pillow library (different down-sampling techniques yield very similar performance as pointed out by [10]). Following [53], we randomly select 10% training samples as the labeled set. The test set is unchanged, and averaged class recall is used to achieve a balanced metric.

Setup. We evaluate our method using FixMatch [49] with ResNet-50 [22] due to its good performance on CIFAR. For all experiments, we train for a total number of 500 epochs. For CReST+, we train for 5 generations with 100 epoch per generation. The rest of hyper-parameters are the same as used in CIFAR-LT.

Results. Table 4 summarizes the results on Small-ImageNet-127. Our method improves over the baseline FixMatch by 7.9% and 6.1% for image size 32 and 64, respectively. Due to the large size of this dataset, we only report numbers for a single data split. Nevertheless, the large and consistent improvement over other methods on this dataset confirms the efficacy of our method.

	Small-ImageNet-127		Food-101-LT	
	32×32	64×64	$\gamma = 50$	$\gamma = 100$
FixMatch	29.7	42.3	42.6	35.3
w/ DARP	30.5	42.5	42.0	34.2
w/ CReST+	32.5	44.7	43.8	31.2
w/ CoSSL	**37.6**	**48.4**	**49.0**	**40.4**

Table 4. Averaged class recall (%) on Small-ImageNet-127 and Food-101. We test image size 32×32 and 64×64 for Small-ImageNet-127 and imbalance ratio $\gamma = 50$ and $\gamma = 100$ for Food-101.

[1] Note that the results from [30] with $\gamma = 20$ are not used here because they were produced by $N_1 = 300, M_1 = 150$: https://github.com/bbuing9/DARP/blob/master/run.sh

[2] One run of vanilla FixMatch on ImageNet127 on a single NVIDIA Tesla V100 takes 10676.5 hours which is about 444 days.

Test imbalance ratio	512	256	**150**	128	64	32	16	8	4	2	1	-2	-4	-8	-16	-32	-64	-128	-256	-512	Mean
	Unknown test-time imbalance ratio																				
Fix	94.83	93.95	93.13	92.87	91.24	89.11	86.62	82.90	78.92	73.58	67.83	61.83	55.41	49.50	44.46	40.37	36.88	33.89	30.95	29.04	66.36
Fix + PC	94.63	93.95	93.30	92.95	91.54	89.89	87.87	84.89	82.05	77.97	73.49	68.86	63.88	59.45	55.70	52.76	50.24	47.90	45.77	44.23	72.57
Fix + vanilla cRT	94.78	93.90	93.17	92.83	91.24	89.24	86.87	83.75	80.29	75.54	70.40	65.10	59.47	54.36	49.86	46.35	43.39	40.81	38.34	36.61	69.31
Fix + DARP	**95.14**	**94.46**	**93.73**	**93.50**	**92.18**	**90.12**	87.70	84.39	81.03	76.26	71.15	66.12	60.99	56.10	52.28	48.84	45.75	43.25	40.79	39.17	70.65
Fix + CReST+	94.18	93.39	92.74	92.45	91.05	89.04	86.70	83.52	80.20	76.05	71.75	67.28	62.76	58.73	55.68	52.89	50.47	48.49	46.61	45.54	71.98
Fix + CoSSL	91.73	91.13	90.90	90.60	89.85	89.07	**87.95**	**86.24**	**84.60**	**82.61**	**80.40**	**78.39**	**76.03**	**74.19**	**73.21**	**72.49**	**71.43**	**70.64**	**70.02**	**69.71**	**81.06**
	Known test-time imbalance ratio																				
Fix + PC	94.98	94.00	93.13	92.83	91.16	89.24	87.03	84.00	81.03	77.31	73.49	70.10	66.79	64.21	62.69	61.89	62.41	63.26	64.80	66.50	77.04
Fix + vanilla cRT	95.14	94.32	93.39	93.25	91.35	89.24	86.73	83.45	79.85	75.04	70.40	65.76	60.65	56.67	53.81	52.04	51.07	51.09	49.98	51.60	72.24
Fix + DARP + PC	**95.19**	**94.46**	**93.73**	**93.54**	**92.32**	**90.32**	**88.17**	**85.53**	83.00	79.96	76.82	74.33	72.05	70.88	70.37	70.53	70.98	71.39	72.19	73.07	80.94
Fix + CReST+ + PC	94.48	93.44	92.74	92.49	91.09	89.17	87.20	84.75	82.60	79.86	77.74	76.09	74.41	74.03	74.40	75.40	76.38	77.22	78.66	80.29	82.62
Fix + CoSSL + PC	92.83	91.59	90.90	90.31	89.22	87.93	86.42	85.01	**84.00**	**82.57**	**82.00**	**81.70**	**81.72**	**81.66**	**82.94**	**84.66**	**85.77**	**86.83**	**87.58**	**88.31**	**86.20**

Table 5. Classification accuracy (%) on CIFAR-10-LT with imbalance ratio $\gamma = 150$. We test different methods on top of FixMatch [49] for known and unknown shifted distributions. Post-compensation (PC) [23] is deployed to utilize the information of the known test distribution.

4.3. Main results on Food-101

Dataset. To evaluate the effectiveness of our method on high-resolution images, we use the fine-grained image classification dataset Food-101 [5]. The original dataset consists of 101 food categories, with 101,000 images. For each class, 250 manually reviewed test images are provided as well as 750 training images. All images were rescaled to have a maximum side length of 512 pixels. We construct Food-101-LT for imbalanced SSL using the same way as CIFAR-10-LT with imbalance ratio $\gamma = 50$ and 100.

Setup. We consider FixMatch [49] as the SSL algorithm due to its good performance. We train a ResNet-50 [22] for 1,000 epochs of unlabeled dataset using a SGD optimizer with momentum 0.9. The learning rate is set to 0.04 without decay, with a linear warm-up for the first 5 epochs. We set the labeled batch size as 256 and the unlabeled batch size as 512. The EMA decay rate is 0.999.

Results. Table 4 shows the results on Food-101-LT. Compared to other methods, which give marginal improvements or, in some cases, even worse performance over the baseline, our method consistently improves the accuracy. We improve the FixMatch baseline by 6.4% and 5.1% at imbalance ratio $\gamma = 50$ and 100, respectively.

4.4. Evaluation at unknown and known shifted test distributions

As mentioned above, the standard evaluation under uniform test distribution is often limited in reflecting real-world scenarios. To this end, we conduct a more realistic evaluation by assessing different methods at shifted test distributions. Moreover, we argue that the test distribution can be given as prior knowledge in real-world applications in some cases. Thus, we distinguish two types of shifted evaluation: known test distributions in which the test distribution is given during training, and unknown test distributions in which this information is unknown. When the test distribution is known, the imbalanced SSL method should be able to accommodate the information for further improvement.

Inspired by [23], we construct shifted test sets with a wide range of imbalance ratios. When $\gamma > 0$, the number of test examples of class k is defined as $N_k = N_1 \cdot \gamma^{-\frac{k-1}{K-1}}$, where class 1 has the most test data. Similarly, $N_k = N_1 \cdot |\gamma|^{\frac{k-1}{K-1}}$ when $\gamma < 0$, where class 1 has the least test data, and, thus, test set is weighted in favor of tail classes. For unknown distributions, we train different methods and evaluate them directly over a family of shifted distributions. The mean accuracy is also reported. When the distribution is known during training, we deploy post-compensation [23] as a post-processing method to utilize this information for all methods. For all experiments, we use FixMatch and train on CIFAR-10-LT with imbalance ratio $\gamma = 150$. Then, we evaluate different methods at unknown and known shifted test distributions varying from imbalance ratio $\gamma = 512$ to -512. All experiments are run with the same data split and the training protocol from Section 4.1. Results of other training settings can be found in the Appendix.

Table 5 summarizes the results. Compared to other methods, our approach has higher mean accuracy for both known and unknown distributions, which is mainly due to the good performance at the negative test imbalance ratios. For example, while being lower at positive ratios, our method is 24.17% and 8.02% better than the second-best at imbalance ratio $\gamma = -512$ in known and unknown cases, respectively. Our method also shows good robustness against the change of test imbalance ratios. For known test distribution, as the information of test distributions is utilized during the training in our method, we achieve a more balanced performance under various imbalance ratios. For example, the performance gap between $\gamma = 512$ and $\gamma = -512$ is 4.52% for our method compared to 14.19% for CReST+ and 22.12% for DARP. Despite the improved performance from our method, the relatively lower results at the negative ratios also indicate that none of the existing methods, including ours, can achieve a real balanced performance. Note that our protocol can be applied for imbalanced supervised learning as well.

ID	Method	Trainig Framework		Unlabeled Data	Enhancement Level		Enhancement Method		Blend Probability	Test Acc.
		Joint	Two-Stage		Image	Feature	TFE	mixUp		
1	CoSSL	✓		✓		✓	✓		✓	80.24
2	remove unlabeled data from TFE	✓				✓	✓		✓	79.26
3	remove blend probability	✓		✓		✓	✓			79.74
4	image-level enhancement	✓		✓	✓		✓		✓	78.92
5	classifier learning with cRT [29]	✓								77.38
6	replace TFE with mixUp [56]	✓		✓		✓		✓	✓	77.31
7	FixMatch + cRT [29]		✓							69.90
8	FixMatch + cRT with mixUp		✓					✓		77.50
9	FixMatch + TFE		✓	✓		✓	✓		✓	77.93
10	FixMatch [49]									68.30

Table 6. Ablation studies of our method. Experiments 1 - 6 study different design choices in TFE. Experiments 7 - 9 present results of two-stage methods to demonstrate the effectiveness of our co-learning framework. The same split on CIFAR-10-LT with imbalance ratio 150 is used for all experiments.

4.5. Ablation study

In this section, we analyze different design choices for our method to provide additional insights into how it helps generalization. We focus on a single split with an imbalance ratio of 150 on CIFAR-10-LT and report results for a Wide ResNet-28-2 [55]. For fair comparison, the same data split is used for all experiments in this section.

Design choices in TFE. The upper half of Table 6 summarizes the results of replacing different components of TFE. As mentioned in Section 3, we apply feature blending with a class-wise probability to utilize unlabeled data and enhance the data diversity of tail classes. When unlabeled data is removed, our TFE reduces to vanilla cRT [29] with a 2.86% test accuracy decrease. And using TFE with labeled data only in the co-learning framework gives 79.26% test accuracy, which indicates that the performance decrease from the unavailability of unlabeled data cannot be compensated by feature fusion alone. Keeping the unlabeled data, we can also replace TFE with mixUp [56] by using pseudo-labels for unlabeled data. However, due to the bias in pseudo-labels, this reduces the test accuracy to 77.31%, which suggests the superiority of TFE for imbalanced SSL. Furthermore, when we remove the class-wise blend probability or apply the enhancement by blending labeled and unlabeled data at pixel level, the accuracy drops by 0.5% and 1.32%, respectively. The ablation studies of the lower bound μ on fusion factor λ from Alg. 1 and the number of warm-up epochs can be found in the Appendix.

Benefits of the co-learning framework. Here we investigate the effect of our co-learning framework by comparing with two-stage approaches. For all three two-stage approaches, we first train a complete FixMatch for representation learning. Then, keeping the feature encoder fixed, the classification layer is reinitialized and trained for 20 epochs. As is shown in the lower half of Table 6, all of the two-stage approaches outperform the FixMatch baseline. However, none of these methods reaches the accuracy of our co-learning method (80.24%), which confirms the benefits of the joint learning framework.

5. Conclusion and limitations

In this work, we study imbalanced SSL, which is a more general setting as both labeled and unlabeled data from imbalanced distributions. We propose CoSSL, a flexible co-learning framework for imbalanced SSL, which decouples the representation learning and classifier learning while connecting them by sharing learned features and generated pseudo-labels. We also design Tail-class Feature Enhancement for learning the classifier with unlabeled data and enhancing the performance at tail classes. Integrating TFE and strong SSL methods into our CoSSL framework, we achieve new state-of-the-art results across a variety of imbalanced SSL benchmarks, especially when the imbalance ratio is large. At the evaluation, we address the limitation of the conventional uniform protocol by evaluating methods at shifted distributions and considering known and unknown test distribution during training. Such a comprehensive evaluation provides more insights into the existing methods and uncovers limitations.

This work, however, is also subject to several limitations. First, this paper focuses on the object recognition problem under class-imbalanced distribution. Therefore, caution must be taken when generalizing to other vision tasks. Second, our method only considers in-class unlabeled data whose potential class labels are covered by the labeled set. However, there are often a large number of out-of-class unlabeled data available in real-world applications. And they are often mixed with in-class unlabeled data, which can be detrimental if not properly handled. Our method, at the current stage, is not able to handle such a case and effectively leverage out-of-class unlabeled data, which we leave for future work. Thirdly, as we have seen from Section 4.4, all of the existing methods, including ours, can not achieve a real balanced performance across test distributions. The performance at distributions that are radically different from the training distribution is relatively lower.

References

[1] Philip Bachman, Ouais Alsharif, and Doina Precup. Learning with pseudo-ensembles. In Advances in neural information processing systems, 2014. 1, 2

[2] Yoshua Bengio, Olivier Delalleau, and Nicolas Le Roux. 11 label propagation and quadratic criterion, 2006. 2

[3] David Berthelot, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution matching and augmentation anchoring. In 8th International Conference on Learning Representations, ICLR, 2020. 1, 2, 3, 5, 6

[4] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. In Advances in Neural Information Processing Systems, 2019. 1, 2, 3, 5

[5] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In European conference on computer vision, 2014. 7

[6] Mateusz Buda, Atsuto Maki, and Maciej A Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. Neural Networks, 2018. 1

[7] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In In Advances in Neural Information Processing Systems, 2019. 2, 5

[8] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. IEEE Transactions on Neural Networks, 20(3), 2009. 2

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

[10] Patryk Chrabaszcz, Ilya Loshchilov, and Frank Hutter. A downsampled variant of imagenet as an alternative to the cifar datasets. arXiv preprint arXiv:1707.08819, 2017. 6

[11] Dan Claudiu Cireşan, Ueli Meier, Luca Maria Gambardella, and Jürgen Schmidhuber. Deep, big, simple neural nets for handwritten digit recognition. Neural computation, 2010. 5

[12] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In Proceedings of the IEEE conference on computer vision and pattern recognition Workshops, 2020. 5

[13] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019. 5

[14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In IEEE conference on computer vision and pattern recognition, 2009. 6

[15] Emily Denton, Sam Gross, and Rob Fergus. Semi-supervised learning with context-conditional generative adversarial networks. arXiv preprint arXiv:1611.06430, 2016. 2

[16] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. arXiv preprint arXiv:1708.04552, 2017. 5

[17] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. International journal of computer vision, 2010. 1

[18] Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In European conference on computer vision, 2016. 1

[19] Haibo He and Edwardo A Garcia. Learning from imbalanced data. IEEE Transactions on knowledge and data engineering, 2009. 1

[20] Haibo He and Yunqian Ma. Imbalanced learning: foundations, algorithms, and applications. Wiley-IEEE Press, 2013. 1

[21] Ju He, Adam Kortylewski, Shaokang Yang, Shuai Liu, Cheng Yang, Changhu Wang, and Alan Yuille. Rethinking re-sampling in imbalanced semi-supervised learning. arXiv preprint arXiv:2106.00209, 2021. 6

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, 2016. 6, 7

[23] Youngkyu Hong, Seungju Han, Kwanghee Choi, Seokjun Seo, Beomsu Kim, and Buru Chang. Disentangling label distribution for long-tailed visual recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021. 2, 7

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

[25] Minyoung Huh, Pulkit Agrawal, and Alexei A Efros. What makes imagenet good for transfer learning? arXiv preprint arXiv:1608.08614, 2016. 6

[26] Minsung Hyun, Jisoo Jeong, and Nojun Kwak. Class-imbalanced semi-supervised learning. arXiv preprint arXiv:2002.06815, 2020. 2

[27] Nathalie Japkowicz. The class imbalance problem: Significance and strategies. In In Proceedings of the International Conference on Artificial Intelligence, 2000. 5

[28] Thorsten Joachims. Transductive learning via spectral graph partitioning. In Proceedings of the 20th International Conference on Machine Learning (ICML), 2003. 2

[29] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In International conference on learning representations, 2020. 1, 2, 3, 4, 5, 8

[30] Jaehyung Kim, Youngbum Hur, Sejun Park, Eunho Yang, SungJu Hwang, and Jinwoo Shin. Distribution aligning refinery of pseudo-label for imbalanced semi-supervised learning. In Advances in neural information processing systems, 2020. 1, 2, 4, 5, 6

[31] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In International conference on learning representations, 2015. 5

[32] Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In Advances in neural information processing systems, 2014. 2

[33] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. International journal of computer vision, 2017. 1

[34] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In Workshop on challenges in representation learning, ICML, 2013. 1, 2

[35] Yu Li, Tao Wang, Bingyi Kang, Sheng Tang, Chunfeng Wang, Jintao Li, and Jiashi Feng. Overcoming classifier imbalance for long-tail object detection with balanced group softmax. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020. 2, 3

[36] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In European conference on computer vision, 2014. 1

[37] Bin Liu, Zhirong Wu, Han Hu, and Stephen Lin. Deep metric transfer for label propagation with limited annotated data. In Proceedings of the IEEE International Conference on Computer Vision Workshops, 2019. 2

[38] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019. 1

[39] Yucen Luo, Jun Zhu, Mengxi Li, Yong Ren, and Bo Zhang. Smooth neighbors on teacher graphs for semi-supervised learning. In Proceedings of the IEEE conference on computer vision and pattern recognition, 2018. 2

[40] Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. In International conference on learning representations, 2021. 2, 3

[41] Yu Nesterov. A method of solving a convex programming problem with convergence rate $o(k^2)$. Doklady Akademii Nauk, 1983. 1, 2

[42] Augustus Odena. Semi-supervised learning with generative adversarial networks. arXiv preprint arXiv:1606.01583, 2016. 2

[43] Youngtaek Oh, Dong-Jin Kim, and In So Kweon. Distribution-aware semantics-oriented pseudo-label for imbalanced semi-supervised learning. arXiv preprint arXiv:2106.05682, 2021. 6

[44] Avital Oliver, Augustus Odena, Colin A Raffel, Ekin Dogus Cubuk, and Ian Goodfellow. Realistic evaluation of deep semi-supervised learning algorithms. In Advances in neural information processing systems, 2018. 5

[45] Antti Rasmus, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko. Semi-supervised learning with ladder networks. In Advances in neural information processing systems, 2015. 1, 2

[46] Chuck Rosenberg, Martial Hebert, and Henry Schneiderman. Semi-supervised self-training of object detection models. Carnegie Mellon University, 2005. 2

[47] Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In Advances in neural information processing systems, 2016. 1, 2

[48] H Scudder. Probability of error of some adaptive pattern-recognition machines. IEEE Transactions on Information Theory, 1965. 1, 2

[49] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In Advances in Neural Information Processing Systems, 2020. 1, 2, 3, 5, 6, 7, 8

[50] Kaihua Tang, Jianqiang Huang, and Hanwang Zhang. Long-tailed classification by keeping the good and removing the bad momentum causal effect. In In Advances in Neural Information Processing Systems, 2020. 2, 3

[51] Grant Van Horn and Pietro Perona. The devil is in the tails: Fine-grained classification in the wild. arXiv preprint arXiv:1709.01450, 2017. 1

[52] Tao Wang, Yu Li, Bingyi Kang, Junnan Li, Junhao Liew, Sheng Tang, Steven Hoi, and Jiashi Feng. The devil is in classification: A simple framework for long-tail instance segmentation. In European Conference on computer vision, 2020. 2, 3

[53] Chen Wei, Kihyuk Sohn, Clayton Mellina, Alan Yuille, and Fan Yang. Crest: A class-rebalancing self-training framework for imbalanced semi-supervised learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021. 1, 2, 4, 5, 6

[54] Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. Unsupervised data augmentation for consistency training. arXiv preprint arXiv:1904.12848, 2019. 1

[55] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In Proceedings of the British Machine Vision Conference (BMVC), 2016. 5, 8

[56] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In 6th International Conference on Learning Representations, ICLR, 2018. 8

[57] Xiaojin Zhu. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005. 2

[58] Xiaojin Zhu and Andrew B Goldberg. Introduction to semi-supervised learning. Synthesis lectures on artificial intelligence and machine learning, 3(1), 2009. 2