




A survey of class-imbalanced semi-supervised learning

Qian Gui¹ · Hong Zhou¹ · Na Guo¹ · Baoning Niu¹ 

Received: 25 May 2022 / Revised: 8 March 2023 / Accepted: 21 April 2023 /

Published online: 19 May 2023

© The Author(s), under exclusive licence to Springer Science+Business Media LLC, part of Springer Nature 2023

Abstract

Semi-supervised learning(SSL) can substantially improve the performance of deep neural networks by utilizing unlabeled data when labeled data is scarce. The state-of-the-art(SOTA) semi-supervised algorithms implicitly assume that the class distribution of labeled datasets and unlabeled datasets are balanced, which means the different classes have the same numbers of training samples. However, they can hardly perform well on minority classes when the class distribution of training data is imbalanced. Recent work has found several ways to decrease the degeneration of semi-supervised learning models in class-imbalanced learning. In this article, we comprehensively review class-imbalanced semi-supervised learning (CISSL), starting with an introduction to this field, followed by a realistic evaluation of existing class-imbalanced semi-supervised learning algorithms and a brief summary of them.

Keywords Deep learning · Class-imbalanced supervised learning · Semi-supervised learning · Class-imbalanced semi-supervised learning

1 Introduction

Deep learning, one of the most popular phrases being used in the field of artificial intelligence in recent years, is effective with a range of practical applications such as computer vision, data mining, and nature language processing, and has achieved great commercial success (Goodfellow et al., 2016) due to the fact that a large number of

Editor: Nuno Moniz, Paula Branco, Luís Torgo, Nathalie Japkowicz, Michal Wozniak, Shuo Wang.

✉ Baoning Niu
niubaoning@tyut.edu.cn

Qian Gui
guiqian0420@link.tyut.edu.cn

Hong Zhou
zhouhong4757@link.tyut.edu.cn

Na Guo
guona2287@link.tyut.edu.cn

¹ School of Information and Computer, Taiyuan University of Technology, Taiyuan 030024, Shanxi, People's Republic of China

high-quality labeled training examples are provided. However, there are various real-world applications where the unlabeled data are readily available and easy to acquire, while labeled instances are often hard, expensive, and time-consuming to collect. Thus it is desirable to be able to learn a good model with a few labeled data. Semi-supervised learning(SSL) (Chapelle et al., 2006) is proposed for the purpose.

SSL is a paradigm that can improve learning performance with a few labeled data by using additional unlabeled examples as auxiliaries compared to supervised learning. It provides a way to explore the latent patterns from extra unlabeled examples, alleviating the need for a large number of labels. The SOTA SSL algorithms often construct a model with a common assumption that the class distribution of the training data is balanced, which means the different classes have the same numbers of training samples. Imbalanced data, however, is widely existing in many realistic scenarios, which leads to the poor performance of SSL algorithms. According to recent research (Yang & Xu, 2020), the models trained on imbalanced data are easily biased towards majority classes which have a large number of training examples, and far away from minority classes which have few training examples(see Fig. 1b).

Class-imbalanced supervised learning(CISL) (Cui et al., 2019; Huang et al., 2020; Liu et al., 2019; Cao et al., 2019; Ren et al., 2020; Zhou et al., 2020a) has been widely explored. Most of them (Cao et al., 2019; Cui et al., 2019; Huang et al., 2020; Liu et al., 2019) handle with quantity imbalance, where the distribution of training examples from different classes is imbalanced, such as the long-tailed distribution (Van Horn et al., 2018; Gupta et al., 2019) and step imbalanced distribution (Buda et al., 2018). Few works also handle topology imbalance learning (Deli et al., 2021). The solutions can be categorized as re-sampling (He & Garcia, 2009; Pouyanfar et al., 2018; Xu et al., 2021), re-weighting (Buda et al., 2018; Byrd & Lipton, 2019; Cui et al., 2019; Park et al., 2021; Huang et al., 2020; Cao et al., 2019), synthetic samples (Chou et al., 2020; Chawla et al., 2002), meta learning (Ren et al., 2020; Shu et al., 2019), transfer learning (Liu et al., 2019; Yin et al., 2019; Jamal et al., 2020) and decoupling representation and classifier (Zhou et al., 2020a; Kang et al., 2020; Zhong et al., 2021). These work usually require extra data to rebalance the imbalanced distribution, such as re-sampling and synthetic samples, and are easily overfitting to some certain class. Extra unlabeled data is easy to obtain and has been proved to improve the model generalization(Yang & Xu,

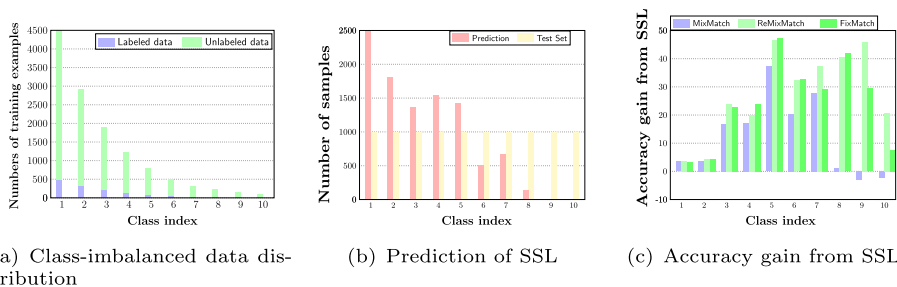


Fig. 1 Experimental results on CIFAR-10-LT under the imbalance ratio $\gamma_l = \gamma_u = 100$. **a** Class distribution of labeled and unlabeled data. **b** Predictions on a class-balanced test set using SSL algorithm MixMatch (Berthelot et al., 2019) **c** Test accuracy gain due to SSL algorithms compares to the vanilla model trained using only labeled data

2020). But these work are designed for supervised learning and do not exploit unlabeled data.

There have been a few studies on class-imbalanced semi-supervised learning(CISSL) (Igal et al., 2015; Salazar et al., 2018; Yang & Xu, 2020; Kim et al., 2020a; Wei et al., 2021a; Lee et al., 2021; Fan et al., 2022; Guo & Li, 2022; Lai et al., 2022). Due to the imbalanced training data(see Fig. 1a), SSL algorithms have to face a great challenge to generalize the minority classes which have few training examples. Pseudo labels for unlabeled data generated by a model trained on labeled data are commonly leveraged in SSL algorithms. Although the large number of unlabeled data can help alleviate the degeneration caused by imbalanced data (Yang & Xu, 2020), the pseudo labels generated by an initial model trained with imbalanced data tend to be biased toward majority classes and deteriorate the model quality(see Fig. 1c). Most SSL methods (Berthelot et al., 2019, 2020; Sohn et al., 2020) have not been evaluated on imbalanced class distribution.

In general, our contribution can be summarized as follows:

- We provide a brief introduction of the deep semi-supervised learning and class-imbalanced supervised learning to better illustrate class-imbalanced semi-supervised learning. We also conduct a comprehensive review of the advanced class-imbalanced semi-supervised learning and summarize them into two categories from the perspective they are used.
- We provide an evaluation of class-imbalanced semi-supervised learning and outline the highlights and limitations of these categories.
- We identify two potential directions for method innovation as well as four new task settings of imbalanced semi-supervised learning for future research.

Due to the reason that existing CISSL are based on the Deep semi-supervised learning and class-imbalanced supervised learning. In order to better illustrate CISSL, we briefly introduce deep semi-supervised learning and class-imbalanced supervised learning before reviewing CISSL. The remainder of this paper is structured as follows. Sections 2 and 3, give a brief introduction to deep semi-supervised learning and class-imbalanced supervised learning, respectively, to facilitate the discussions about CISSL. Section 4 comprehensively reviews existing algorithms for CISSL. Section 5 evaluates and analyzes these CISSL methods when labeled and unlabeled data have the same/different imbalanced class distribution. Section 6 discusses the future research directions for CISSL.

2 Formal definition and taxonomy of deep semi-supervised learning

We firstly give a brief review to the Deep SSL in this section. In the standard deep SSL task, we are provided a set of large training examples, which include n labeled examples $\mathcal{D}_l = \{(x_1, y_1), \dots, (x_n, y_n)\}$ and m unlabeled examples $\mathcal{D}_u = \{x_{n+1}, \dots, x_{n+m}\}$. Generally, $m \gg n, x \in \mathcal{X} \in \mathbb{R}^D, y \in \mathcal{Y} = \{1, \dots, C\}$ where D is the number of input dimensions and C is the number of output classes in training examples. The aim of a deep SSL algorithm is to find an appropriate learning model $f(x; \theta) : \{\mathcal{X}; \Theta\} \rightarrow \mathcal{Y}$ parameterized by $\theta \in \Theta$ from training data, which has higher accuracy than what would have been obtained by only using the labeled data \mathcal{D}_l . In supervised learning, the loss function is always defined as $\min_{\theta \in \Theta} \sum_{x, y \in \mathcal{D}_l} \mathcal{L}_s(f(x; \theta), y)$. Obviously, it is limited to supervised loss and ignores the

useful information of unlabeled data. The deep SSL algorithms usually utilizes unlabeled data by introducing unsupervised loss and regularization. Generally, the loss function optimized by SSL algorithms can be defined using Eq. 1:

$$\min_{\theta \in \Theta} \underbrace{\sum_{x,y \in \mathcal{D}_l} \mathcal{L}_s(f(x;\theta), y)}_{\text{supervised loss}} + \lambda \underbrace{\sum_{x \in \mathcal{D}_u} \mathcal{L}_u(f(x;\theta))}_{\text{unsupervised loss}} + \beta \underbrace{\sum_{x \in \mathcal{D}_l \cup \mathcal{D}_u} \Omega(x;\theta)}_{\text{regularization term}} \tag{1}$$

where \mathcal{L}_s refers the supervised loss, \mathcal{L}_u refers the unsupervised loss, Ω refers the regularization term and $\lambda, \beta \in \mathbb{R} > 0$ denotes the relative weight of the corresponding loss, which balances the loss terms. It is worth mentioning that regularization terms are often regarded as the unsupervised loss in some algorithms, which means that there is not a clear distinction between unsupervised loss and regularization terms. Different choices for the unsupervised loss and regularization terms lead to different deep semi-supervised learning algorithms.

On the one hand, the optimization of regularization terms that are called consistency regularization are designed to make the predictive results to have consistency under various disturbances, which improves the generalization of SSL algorithms by using extra unlabeled data. On the other hand, the optimization of the unsupervised loss, also called entropy minimization, is designed to make the prediction made by training models have high confidence, and prevent the class distribution of predictive results from being too flat and having no tendency. Furthermore, holistic methods combining the entropy minimization and the consistency regularization get larger accuracy gain in SSL algorithms. It is worthy mentioning that although these algorithms are mainly based on deep neural networks, they are also applicable with non-deep neural networks or even with classifiers that do not use neural networks at all. We use "deep" because we want to distinguish it from the traditional SSL algorithms. The overall taxonomy used in SSL algorithms is shown in Fig. 2.

2.1 Entropy minimization

Entropy minimization (Grandvalet & Bengio, 2005), based on low-density assumption (Zhou, 2017), is a way that encourage deep neural networks to make high confident predictions on unlabeled data regardless of the predicted class. Naturally, entropy minimization discourages

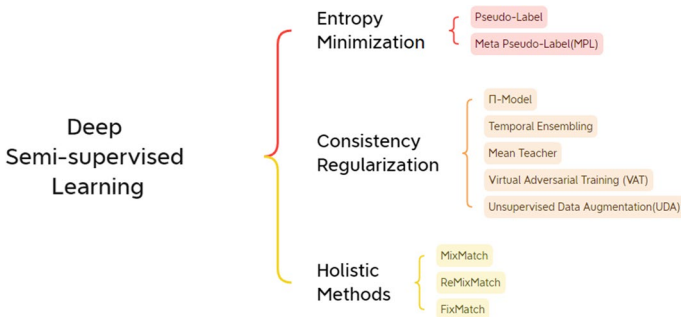


Fig. 2 The taxonomy of representative deep semi-supervised learning methods based on the design of loss function

the decision boundary from passing the nearby area of data points where it would otherwise be forced to produce low-confidence predictions. This is done by adding an unsupervised loss term to minimize the entropy of the prediction function $f(x; \theta)$ on unlabeled data. For a set of training examples with C output classes, the entropy minimization term can be defined as Eq. 2.

$$\mathcal{L}_u = \sum_{k=1}^C -f(x; \theta)_k \log f(x; \theta)_k \quad (2)$$

Inspired by entropy minimization, researchers on SSL algorithms later propose Pseudo-Label (Lee, 2013) and Meta Pseudo-Label (MPL) (Pham et al., 2021). Pseudo-Label produces “pseudo labels” for unlabeled data using the prediction function itself over the course of training, and uses those with a corresponding class probability larger than a pre-defined threshold as targets for a standard supervised loss function applied to. MPL uses the student-teacher setting, where the teacher model and the student model are trained in parallel. The teacher model is responsible for generating better pseudo labels and the student model learns from the pseudo labels generated by the teacher model.

However, models trained with class-imbalanced data can overfit to data points from classes which have a large number of training examples, resulting in a model which is biased towards majority classes and away from minority classes.

2.2 Consistency regularization

Consistency regularization, based on manifold assumption (Zhou, 2017), can be seen as a way of utilizing the unlabeled data to find a smooth manifold on which the dataset lies. It describes a class of methods (Rasmus et al., 2015; Sajjadi et al., 2017; Laine & Aila, 2017; Tarvainen & Valpola, 2017; Miyato et al., 2019; Xie et al., 2020) with following intuitive goal: Giving a perturbations $x + \zeta \rightarrow \hat{x}$ to data points x , its prediction output $f(x; \theta)$ should have consistency. Generally, this involves minimizing $d(f(x; \theta), f(\hat{x}; \theta))$ where d measures a distance between the prediction function’s outputs, e.g. mean squared error (MSE) (Sajjadi et al., 2017), Kullback–Leibler divergence (KL) (Cover & Thomas, 1999) or Jensen-Shannon divergence (JS) (Lin, 1991). For a training example with C possible output classes, and $m = \frac{1}{2}(f(x; \theta) + f(\hat{x}; \theta))$, the measure can be calculated as follows.

$$d_{MSE}(f(x; \theta), f(\hat{x}; \theta)) = \frac{1}{C} \sum_{k=1}^C (f(x; \theta)_k - f(\hat{x}; \theta)_k)^2 \quad (3)$$

$$d_{KL}(f(x; \theta), f(\hat{x}; \theta)) = \frac{1}{C} \sum_{k=1}^C f(x; \theta)_k \log \frac{f(x; \theta)_k}{f(\hat{x}; \theta)_k} \quad (4)$$

$$d_{JS}(f(x; \theta), f(\hat{x}; \theta)) = \frac{1}{2} d_{KL}(f(x; \theta), m) + \frac{1}{2} d_{KL}(f(\hat{x}; \theta), m) \quad (5)$$

This simple principle has produced a series of methods (Rasmus et al., 2015; Sajjadi et al., 2017; Laine & Aila, 2017; Tarvainen & Valpola, 2017; Miyato et al., 2019; Xie et al., 2020) commonly used for SSL. They differ in the use of data perturbation methods and distance calculation methods. The Ladder Network (Rasmus et al., 2015) uses only one perturbation

to produce consistency regularization. Furthermore, Π -Model(Sajjadi et al., 2017) creates two random perturbations of a sample for both labeled and unlabeled data. By making the same unlabeled sample propagates forward twice in each epoch of the training process, it introduced the random perturbations (Hinton et al., 2012; Ciresan et al., 2012). Similar to the Π -Model, Temporal Ensembling (Laine & Aila, 2017) forms a consensus prediction under different regularization and input augmentation conditions. Then, Mean-Teacher (Tarvainen & Valpola, 2017) replaces the output of an ensemble model using an exponential moving average of model weighting parameters. Inspired by adversarial training (Goodfellow et al., 2014), Miyato et al. (2019) propose Virtual Adversarial Training (VAT) to using the adversarial noise as the additive perturbation, which can maximally change the output class distribution. Unsupervised Data Augmentation (Xie et al., 2020) uses advanced data augmentation methods, such as AutoAugment(Cubuk et al., 2018), Rand Augment(Cubuk et al., 2019) and Back Translation(Edunov et al., 2018), as perturbations for consistency training based SSL.

These consistency regularization methods work well when the class distribution is balanced, but they can be overfitting to the data from majority classes due to the reason that consistency loss is mainly determined by the majority of training samples.

2.3 Holistic methods

Entropy minimization and consistency regularization both achieve great success in semi-supervised learning. An emerging line of work (Berthelot et al., 2019, 2020; Sohn et al., 2020) in SSL is a set of holistic approaches that try to unify the current dominant methods in SSL in a single framework, achieving better performances.

2.3.1 MixMatch

MixMatch (Berthelot et al., 2019) is a holistic approach which incorporates ideas of consistency regularization (Sajjadi et al., 2017), pseudo-labeling (Lee, 2013) and MixUp (Zhang et al., 2019), resulting in an algorithm that surpasses the performance of the traditional approaches (Lee, 2013; Rasmus et al., 2015; Laine & Aila, 2017; Tarvainen & Valpola, 2017; Miyato et al., 2019).

Giving a batch \mathcal{X} from the labeled set \mathcal{D}_l containing pairs of inputs and their corresponding one-hot targets and an equal-sized batch \mathcal{U} from the unlabeled set \mathcal{D}_u containing only unlabeled data, MixMatch produces a batch of augmented labeled examples \mathcal{X} and a batch of augmented unlabeled examples \mathcal{U} with their proxy labels \hat{y} , which can be used to compute the losses.

$$\mathcal{X}', \mathcal{U}' = \text{MixMatch}(\mathcal{X}, \mathcal{U}, T, K, \alpha) \quad (6)$$

$$\mathcal{L}_s = \frac{1}{|\mathcal{X}'|} \sum_{x, y \in \mathcal{X}'} H(y, f(x; \theta)) \quad (7)$$

$$\mathcal{L}_u = \frac{1}{|\mathcal{U}'|} \sum_{x, \hat{y} \in \mathcal{U}'} d_{MSE}(\hat{y}, f(x; \theta)) \quad (8)$$

$$\mathcal{L} = \mathcal{L}_s + \lambda \mathcal{L}_u \quad (9)$$

where K is the number of augmented versions for per unlabeled example, T is the sharpen temperature of the categorical distribution (Goodfellow et al., 2016) to reduce the guessed label overlap, H is the function of cross entropy loss, α is a Beta distribution parameter for MixUp and λ is a weight of the unsupervised loss.

MixMatch produces K augmentations $\hat{x}_1, \dots, \hat{x}_K$ for each unlabeled example x and averages the corresponding class probability as the pseudo label $\hat{y} = \frac{1}{K} \sum_{k=1}^K f(\hat{x}_k, \theta)$. The generated pseudo labels \hat{y} in the form of a probability distribution over C classes are sharpened by adjusting the temperature T , computed as follows where $(\hat{y})_i$ refers to the probability of class i out of C classes.

$$(\hat{y})_i = (\hat{y})_i^{\frac{1}{T}} / \sum_{j=1}^C (\hat{y})_j^{\frac{1}{T}} \quad (10)$$

Then, after creating two augmented batches \mathcal{X} and \mathcal{U} using MixUp (Zhang et al., 2019), MixMatch trains the model using the standard SSL losses by computing the cross entropy loss ($H(p, y)$) for the supervised loss \mathcal{L}_s , and the consistency loss for the unsupervised loss \mathcal{L}_u .

2.3.2 ReMixMatch

Berthelot et al. (2020) propose to improve MixMatch (Berthelot et al., 2019) by introducing two new techniques, distribution alignment and augmentation anchoring. Distribution alignment encourages the marginal distribution of predictions on unlabeled data to be close to that of ground-truth labels. Let y be the class distribution in the true labels and \tilde{y} be a running average of model prediction on unlabeled data. The model prediction $q = f(x; \theta)$ on an unlabeled sample x is normalized to be $\tilde{q} = \text{Normalize}(q \times y / \tilde{y})$ to match the true distribution, where $\text{Normalize}(k)_i = k_i / \sum_j k_j$. Then \tilde{q} is used as the pseudo label for x . Augmentation anchoring feeds K strongly augmented versions using CTAugment (Control Theory Augment) (Berthelot et al., 2020) of the input into the model, CTAugment only samples augmentations that keep the model predictions within the network tolerance compared with the prediction for a weakly-augmented version of the same input.

The ReMixMatch loss consists of four terms: a supervised loss with data augmentation and MixUp applied; an unsupervised loss with data augmentation and MixUp applied, using pseudo labels as targets; a cross entropy loss on a single heavily-augmented version of unlabeled image without MixUp; and a rotation loss (Gidaris et al., 2018; Zhai et al., 2019) as in self-supervised learning.

2.3.3 FixMatch

Sohn et al. (2020) combines consistency regularization and pseudo-labeling with a simple framework as well as using weak and strong augmentation for consistency regularization separately. For supervised loss \mathcal{L}_u , FixMatch computes standard cross-entropy loss on a weakly augmented version of labeled examples $A_w(x)$ from the labeled set \mathcal{D}_l . For unsupervised loss, FixMatch first computes the model's predicted class distribution with a weakly augmented unlabeled example from the unlabeled set \mathcal{D}_u . Then, the predicted label

is retained as pseudo label if the highest class probability is greater than the threshold τ . With a pseudo label, strongly augmented unlabeled example $A_s(x)$ is generated to assign the pseudo label obtained with the weakly labeled version. The total loss can be written as follows:

$$\mathcal{L}_s = \frac{1}{|\mathcal{D}_l|} \sum_{x \in \mathcal{D}_l} H(y, f(A_w(x); \theta)) \quad (11)$$

$$\mathcal{L}_u = \frac{1}{|\mathcal{D}_u|} \sum_{x \in \mathcal{D}_u} 1(\max(f(A_w(x); \theta) > \tau)) H(f(A_w(x); \theta), f(A_s(x); \theta)) \quad (12)$$

$$\mathcal{L} = \mathcal{L}_s + \lambda \mathcal{L}_u \quad (13)$$

here H is the function of cross entropy loss and λ is a weight of the unsupervised loss.

According to the ablation studies of FixMatch, Cutout (Devries & Taylor, 2017) and CTAugment (Berthelot et al., 2020) as part of strong augmentations are necessary for good performance. When the weak augmentation for label guessing is replaced with strong augmentation, the model diverges early in training. If discarding weak augmentation completely, the model overfits the guessed labels. Using weak instead of strong augmentation for pseudo label prediction leads to unstable performance.

2.4 Summary of discussions

As discussed above, the hybrid methods integrate the most successful approaches in SSL, such as entropy minimization, consistency regularization and data augmentation, and adapt them in order to achieve SOTA performance. However, these deep semi-supervised algorithms can hardly achieve the same effect in class-imbalanced data distribution as in class-balanced data distribution. But due to its great success in SSL, existing CISSL algorithms use these algorithms as backbone to make sure that they can utilize the high-quality representations learned by the backbone.

3 Formal definition and taxonomy of class-imbalanced supervised learning

Class-imbalanced supervised Learning (CISL) is another field closely related to CISSL. In standard CISL, we are given a labeled dataset $\mathcal{D} = \{(x_1, y_1), \dots, (x_n, y_n)\}$, $x \in \mathcal{X} \in \mathbb{R}^D$, $y \in \mathcal{Y} = \{1, \dots, C\}$ where D is the number of input dimensions and C is the number of output classes in training examples. We denote the number of labeled data points of class c as N_c , i.e., $\sum_{c=1}^C N_c = N$, and assume that the C classes are sorted according to cardinality in descending order, i.e., $N_1 \geq N_2 \geq \dots \geq N_C$. We denote the ratio of the class imbalance as $\gamma = \frac{N_1}{N_C}$. Under class-imbalanced scenarios, especially the long-tailed distribution, $\gamma \gg 1$. The aim of CISL algorithms is to find an appropriate learning model $f(x; \theta) : \{\mathcal{X}; \Theta\} \rightarrow \mathcal{Y}$ parameterized by $\theta \in \Theta$ from training data, which can alleviate the performance degradation caused by using imbalanced training data.

An intuitive solution to class-imbalanced tasks is to make the algorithms have good performance on both majority classes and minority classes by rebalancing the data

distribution. Most SOTA methods use the class-balanced re-sampling (He & Garcia, 2009; Pouyanfar et al., 2018; Xu et al., 2021) or loss re-weighting (Buda et al., 2018; Byrd & Lipton, 2019; Cui et al., 2019; Park et al., 2021; Huang et al., 2020; Cao et al., 2019) to “simulate” a balanced training set. However, they may under-represent the majority class or have gradient issues during optimization. Other learning paradigms, including transfer learning (Liu et al., 2019; Yin et al., 2019; Jamal et al., 2020), synthetic samples (Chawla et al., 2002; Chou et al., 2020) and meta-learning (Ren et al., 2020; Shu et al., 2019), have also been explored. Recent studies (Zhou et al., 2020a; Kang et al., 2020; Zhong et al., 2021) also find that decoupling the representation and classifier can lead to better imbalanced learning results. The overall taxonomy used in CISL algorithms is shown in Fig 3

3.1 Re-sampling

The re-sampling (He & Garcia, 2009; Liu et al., 2009; Shen et al., 2016; Devi et al., 2017; Pouyanfar et al., 2018; Gupta et al., 2019; Kim et al., 2020b; Xu et al., 2021) approach directly balances the training data distributions by re-sampling, e.g., under-sampling (He & Garcia, 2009; Liu et al., 2009; Devi et al., 2017) the majority classes or oversampling (Shen et al., 2016; Pouyanfar et al., 2018; Gupta et al., 2019; Kim et al., 2020b) the minority classes. However, under-sampling of head-classes may lose some valuable information, and is not applicable when the data imbalance between classes is significant as missing a lot of information. Over-sampling is susceptible to overfitting to certain repetitive samples, and often requires a longer training time.

3.2 Re-weighting

Cost-sensitive re-weighting methods assign different weights to samples to adjust their importance. Commonly used methods include re-weighting samples inversely proportional to the number of the class (Wang et al., 2017; Huang et al., 2020) or the square root of class frequency (Mahajan et al., 2018). Instead of heuristically using the number of

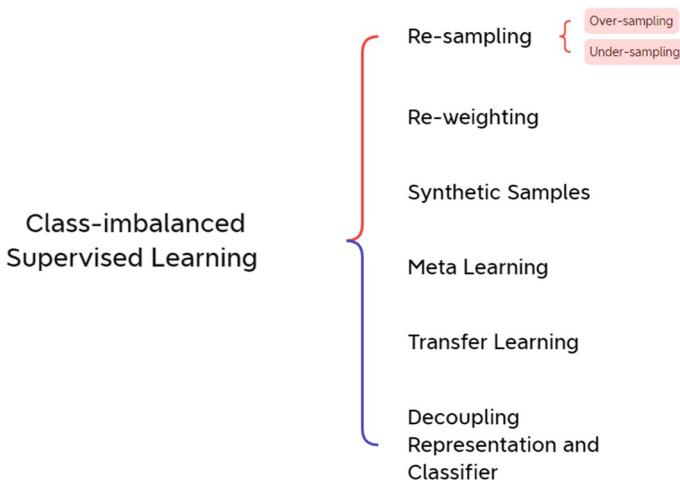


Fig. 3 The taxonomy of representative class-imbalanced supervised learning methods

classes, Cui et al. (2019) proposed using the effective number of samples, and Cao et al. (2019) proposed label-distribution-aware margin loss to solve the overfitting to the minority classes by regularizing the margins. Lin et al. (2017) proposed Focal Loss to improve the Cross-Entropy loss by adding a modulating factor, which distinguishes between simple and hard samples. The weight of the simple samples is reduced, while paying more focus on the hard samples. so Focal Loss can effectively improve the learning of tail classes. While these methods can successfully assign more weights to the minority samples, they assign the same weights to all samples belonging to the same class, regardless of individual importance. It makes deep models with large-scale data difficult to optimize during training and may suffer from heavy over-fitting to tail classes, especially on small datasets.

3.3 Synthetic samples

Synthetic samples (Chawla et al., 2002; He et al., 2008; Chou et al., 2020; Zhong et al., 2021; Dablain et al., 2022) is to generate "new" data similar to the samples belonging to the minority classes. The classic method SMOTE (Chawla et al., 2002), synthesizes a sample by linearly interpolating the K-nearest neighbor of the randomly selected few. It is similar to data augmentation. Inspired by the method of data augmentation called MixUp (Zhang et al., 2019; Chou et al., 2020) proposed the MixUp version of class-imbalanced data distribution. Another classical approach is ADASYN (He et al., 2008), which can adaptively decide how many synthetic samples to generate for each minority class based on the distribution of the samples. First the degree of imbalance as well as the number of new synthetic samples to be generated are calculated, then the distribution of each minority class sample is calculated and the distribution is used to determine the number of synthetic samples for each class. Synthetic samples expand the data in tail classes, alleviating the imbalance of training samples. It is very efficient and economical, especially in some cases where data is hard to obtain. But the new data created by the synthetic approach does not belong to the real dataset. It may easily be influenced by the noise and other undesirable factors that will degenerate the model performance from the original dataset.

3.4 Meta learning

Recently, the meta-learning based approach (Ren et al., 2020; Shu et al., 2019) has emerged to enhance the performance of re-weighting and re-sampling. Shu et al. (2019) proposed a meta-learning process to learn a weighting function. Ren et al. (2020) proposed the meta-sampler and a balanced softmax function, which accommodates the shift of the distributions between the training data and test data. Although these methods can achieve satisfactory performance, they are somewhat difficult to implement in practice. For example, meta-weight-net (Shu et al., 2019) requires additional unbiased data for learning, and the meta-sampler (Ren et al., 2020) is computationally expensive in practice.

3.5 Transfer learning

Due to the rich training resources in the head classes. Some researchers try to leverage the knowledge learned from the head class to guide the learning of the tail class with few training samples. These transfer-based approaches (Liu et al., 2019; Yin et al., 2019; Liu et al., 2020; Jamal et al., 2020) are aimed to share feature knowledge between head and tail

classes. The basic idea of these methods is to model the samples from majority classes and the samples from minority classes separately, and transfer the representation learned from the majority classes to minority classes. For instance, Liu et al. (2019) constructed a feature cloud for each feature, transferring from the head classes to extend the distribution of the tail classes. Yin et al. (2019) trained less biased classifiers by leveraging the knowledge of intra-class variance from head-classes to tail-classes, adapting the feature distribution of tail-classes to mimic that of head-classes. Following that, Liu et al. (2020) transferred the intra-class distribution of head classes to tail classes in the feature space, encouraging the tail classes to achieve similar intra-class angular variability with the head classes. But these methods usually need complex model design for knowledge transfer and may cause the performance degradation of head classes.

3.6 Decoupling representation and classifier

Recent work (Zhou et al., 2020a; Kang et al., 2020; Zhong et al., 2021) find that although class rebalance matters for jointly training representation and classifier, using instance-balanced sampling can provide more general representations. They find that using random data sampling in representation learning and class-balanced sampling in classifier learning can perform better than conventional one-stage methods. Based on this observation, Kang et al. (2020) achieved SOTA results by decoupling representation and classifier learning. In representation learning, the model is trained with instance-balanced sampling. And then, the classifier of the model is fine-tuned with class-balanced sampling to obtain a classifier with balanced decision boundaries, on top of the learned representations. Similarly, Zhou et al. (2020a) integrated MixUp training into the proposed cumulative learning strategy. It bridges the representation learning and classifier rebalancing. The cumulative learning strategy is designed to first learn the universal patterns and then pay attention to the tail data gradually. Furthermore, Zhong et al. (2021) proposes a method designing label-aware smoothing to handle different degrees of overconfidence for classes and reduce dataset bias by shift learning on the batch normalization (Ioffe & Szegedy, 2015) layer in the decoupling framework.

3.7 Summary of discussions

These CISL algorithms are designed for supervised learning and always require label information to balance the class-imbalanced data distribution, thus, not applicable to unlabeled data. The performance degradation of deep learning models is still widespread under extreme class imbalance.

4 Class-imbalanced semi-supervised learning

Most CISSL algorithms are developed from SSL or/and CISL algorithms. With the discussions about algorithms for SSL in Sect. 2 and CISL in Sect. 3, we are ready to discuss CISSL algorithms. Similar to the standard deep SSL task, the training data of the CISSL task consists of n labeled examples $\mathcal{D}_l = \{(x_1, y_1), \dots, (x_n, y_n)\}$ and m unlabeled examples $\mathcal{D}_u = \{x_{n+1}, \dots, x_{n+m}\}$. Generally, $m \gg n$, $x \in \mathcal{X} \in \mathbb{R}^D$, $y \in \mathcal{Y} = \{1, \dots, C\}$ where D is the number of input dimension and C is the number of output class in training examples. We denote the number of data points in class C under \mathcal{D}_l and \mathcal{D}_u as n_c and m_c , respectively, i.e.,

$\sum_{c=1}^C n_c = n$ and $\sum_{c=1}^C m_c = m$. We assume that the C classes are sorted in descending order, i.e., $n_1 \geq n_2 \geq \dots \geq n_C$ and $m_1 \geq m_2 \geq \dots \geq m_C$. The ratio of the class imbalance under \mathcal{D}_l and \mathcal{D}_u are denoted as $\gamma_l = \frac{n_1}{n_c}$ and $\gamma_u = \frac{m_1}{m_c}$. Under class-imbalanced scenarios, $\gamma_l \gg 1, \gamma_u \geq 1$. In general, we assume that \mathcal{D}_l and \mathcal{D}_u have the same distribution, i.e., $\gamma_l = \gamma_u$. But there are some cases where \mathcal{D}_l and \mathcal{D}_u have different distributions, i.e., $\gamma_l \neq \gamma_u$. The aim of CISSL algorithms is to find an appropriate learning model $f(x; \theta) : \{\mathcal{X}; \Theta\} \rightarrow \mathcal{Y}$ parameterized by $\theta \in \Theta$ from imbalanced training data to mitigate the generalization risk.

In general, accurate decision boundaries can be obtained in class-imbalanced settings through self-supervised learning and semi-supervised learning (Yang & Xu, 2020). The representatives are DARP (Kim et al., 2020a), CReST (Wei et al., 2021a), ABC (Lee et al., 2021), DASO (Oh et al., 2021), COSSL (Fan et al., 2022) and Adsh (Guo & Li, 2022). Due to the strategy they used, these CISSL algorithms can be divided into two parts, one is aimed to improve the pseudo-label acquired by the training model from SSL perspective. The other one is to employ classifier adjustment to acquire a balanced classifier from the CILS perspective. The overall taxonomy used in CISSL algorithms is shown in Fig 4.

4.1 Pseudo labeling

Existing SSL studies (Lee, 2013) are mostly based on generating pseudo labels for unlabeled data from the prediction of the training model. But the pseudo labels could be even more imbalanced compared with the true labels of labeled and unlabeled data due to the biased prediction of the model, caused by imbalanced data distribution (Kim et al., 2020a). The low-quality pseudo labels degenerate the model performance. Several methods (Kim et al., 2020a; Wei et al., 2021a; Oh et al., 2021; Guo & Li, 2022) have been proposed to acquire high quality pseudo labels.

4.1.1 DARP

Although higher accuracy could be achieved by using extra unlabeled data in class-imbalanced learning, the model accuracy is mainly improved for majority classes and even declined for minority classes. Recent studies (Kim et al., 2020a; Wei et al., 2021a) also find that pseudo labels generated by initial model trained on imbalanced data are biased toward majority classes. Subsequent training with such biased pseudo labels intensifies the bias and deteriorates the model quality.

Kim et al. (2020a) refine the original biased pseudo labels to match the true distribution of unlabeled data by formulating a Lagrangian dual optimization problem, which can minimize

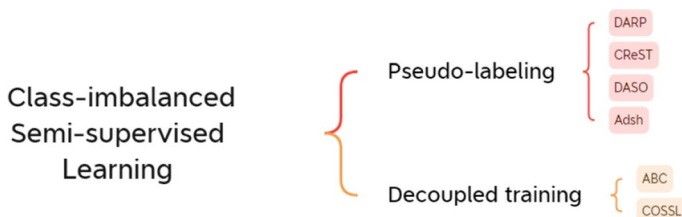


Fig. 4 The taxonomy of representative class-imbalanced semi-supervised learning methods

the distortion from the original pseudo labels and match the true distribution of the training data. In order to get higher quality of the refined pseudo labels, they remove some small and noisy entries of the original pseudo labels in DARP. DARP is flexible to cope with different scenarios and can be combined with some CISL algorithms (Kang et al., 2020).

4.1.2 CReST

Wei et al. (2021a) find that the biased model trained on class-imbalanced data indeed performs favorably on majority classes in terms of recall, but favors minority classes in terms of precision, which indicates that many samples from minority classes are predicted as one of the majority classes. Based on this finding, they propose a self-training technique called CReST to balance biased models.

To accommodate the class imbalance, CReST uses two modifications to the self-training strategy. First, instead of solely training on the labeled data, CReST uses SSL algorithms to exploit both labeled and unlabeled data to get a better initial model in the first step. In the second step, rather than including every sample that has pseudo labels with high confidence into the labeled set, CReST instead expands the labeled set with a re-sampling strategy (Xu et al., 2021). CReST chooses the pseudo labels following a class-rebalancing rule: the less frequent a class c is, the more unlabeled samples that are predicted as class c are included into the labeled set. Due to the reason that the minority classes maintain high precision in the biased models, CReST can get more pseudo labels that are close to the true labels from the minority classes, which alleviate the data imbalance. By introducing progressive distribution alignment (Berthelot et al., 2020), it also improves the quality of pseudo labels, which is distinguished as CReST+.

4.1.3 DASO

Oh et al. (2021) observed that semantic pseudo-labels (Han et al., 2020) obtained from a similarity-based classifier (Snell et al., 2017) $p = g(x; \theta)$ are biased towards minority classes as opposed to linear classifier-based pseudo-labels (Lee, 2013; Sohn et al., 2020) $q = f(x; \theta)$ being biased towards majority classes. To obtain more balanced pseudo-label, Oh et al. (2021) proposed to blend the linear and semantic pseudo-labels for each class in different proportions, which is based on the distribution of the unlabeled training data. Inspired by the success of consistency regularization (Xie et al., 2020; Sohn et al., 2020), Oh et al. (2021) also introduced semantic alignment loss \mathcal{L}_{align} as extra regularization term. It use a weakly augmented version $A_w(x)$ and a strongly augmented version $A_s(x)$ of labeled examples x from the labeled set \mathcal{D}_u to compute the semantic alignment loss. The total loss can be written as follow:

$$\mathcal{L}_{align} = H(g(A_w(x); \theta), g(A_s(x); \theta)) \quad (14)$$

$$\mathcal{L}_{total} = \mathcal{L}_{back} + \mathcal{L}_{align} \quad (15)$$

where \mathcal{L}_{back} refers to the loss of the backbone algorithm (Sohn et al., 2020). It is worth mentioning that the original linear pseudo-label of the backbone algorithms is replaced by the proposed blended pseudo-label.

4.1.4 Adsh

FixMatch uses a fixed threshold for all classes to select pseudo-label, in order to select correct pseudo-labels and discard noise ones. But it is not available when training data is class-imbalanced. In CISSL, the training models are easily biased toward majority classes. Samples predicted as minority class tend to be eliminated while samples predicted as majority classes tend to be selected. And the discarded samples that are predicted as minority classes still hold high precision, while the retained samples that are predicted as majority classes have low precision and many of them may have wrong pseudo-labels, which leads to performance degradation.

In order to obtain more correct pseudo-label, Guo and Li (2022) proposed the adaptive thresholding for different classes to minimize empirical risk. Similar to curriculum learning (Zou et al., 2019), it uses the percentage of selected pseudo-labels for the most majority class to measure the learning effect. The threshold of each class depends on the number of pseudo-labels to be selected for the class and the confidence of selected pseudo-label of each class, making sure that the same percentage of pseudo-labels is selected for each class. This ensures pseudo-labels with the same confidence level within class can be selected for every class.

4.2 Balanced classifier learning

Unlike the algorithms trying to obtain pseudo-labels with higher quality, the balanced classifier learning method aims to employ class-balanced sampling or post-hoc classifier adjustment from the CISL perspective. It alleviates the need for a large number of pseudo-labels with high quality to rebalance the distribution.

4.2.1 ABC

To alleviate the bias caused by the class-imbalanced loss, Lee et al. (2021) provides an auxiliary balanced classifier (ABC) to rebalance the biased model by introducing extra regularization terms.

Inspired by the success of decoupling representation and classifier (Zhou et al., 2020a; Kang et al., 2020; Zhong et al., 2021) in CISL, ABC is attached to a representation layer immediately preceding the classification layer of the backbone, based on the argument that a classification algorithm can learn high-quality representations even if its classifier is biased toward the majority classes. ABC is trained to be balanced across all classes by using a mask that rebalances the class distribution, which is similar to re-sampling in CISL studies. The mask stochastically regenerates a class-balanced subset of a minibatch on which the ABC is trained. So ABC can overcome the limitations of the previous re-sampling techniques, including overfitting on minority-class data and loss of information on majority-class data. To increase the margin between the decision boundary and the data points using unlabeled data, beside the classification loss \mathcal{L}_{cls} of the auxiliary balanced classifier, ABC also conducts a consistency regularization loss \mathcal{L}_{con} for the auxiliary balanced classifier, similar to the way in FixMatch. The total loss function \mathcal{L}_{total} can be calculated as:

$$\mathcal{L}_{total} = \mathcal{L}_{cls} + \mathcal{L}_{con} + \mathcal{L}_{back}. \quad (16)$$

where \mathcal{L}_{back} refers to the loss of the backbone algorithm (Sohn et al., 2020; Berthelot et al., 2020). By using extra regularization term, ABC can mitigate the bias of the learning model caused by class imbalance.

4.2.2 COSSL

The COSSL (Fan et al., 2022) decouples the training of representation and classifier while coupling them in a non-gradient manner. The learning is decoupled into two parts: the semi-supervised representation learning and balanced classifier learning. COSSL connects them by sharing the pseudo-label generated by the balanced classifier and the shared feature learned by representation learning.

During the semi-supervised representation learning, the training samples are selected randomly. COSSL uses the backbone algorithms to obtain a good representation and share it with a momentum encoder for feature extraction. During the classifier learning, the labeled data is selected by a balanced sampler and unlabeled data is selected by a random sampler. Then, it produces a balanced classifier by using Tail-class Feature Enhancement, which is similar to re-sampling and MixUp (Zhang et al., 2019). Different from MixUp, it blends the labeled data and unlabeled data with the probability depending on the class distribution so that the more labeled data a class has, the less fused data is synthesized for classifier learning. It generated pseudo-labels for representation learning by using the momentum encoder from representation learning and the balanced classifier from balanced classifier learning, enhancing both representation learning and classifier learning.

5 Evaluation

In this section, we evaluate various algorithms including SSL, CISL and CISSL under various scenarios for class-imbalanced classification problems. We first provide description of our experimental setups in Sect. 5.1, and give empirical evaluations on existing CISSL algorithms and other baseline algorithms under various scenarios.

5.1 Experimental setup

We choose CIFAR-10/100 (Krizhevsky, 2009) and SHVN (Netzer et al., 2011) as the basic datasets to create various class-imbalanced datasets with the class-imbalance ratio of labeled data γ_l and the class-imbalance ratio of unlabeled data γ_u . There are two types of class imbalance, the long-tailed imbalance where the number of data points exponential decline from the largest class to the smallest class, i.e. $n_k = n_1 * \gamma^{\frac{1-k}{L-1}}$, and the step imbalance (Buda et al., 2018) where the majority classes have same number of data points and the minority classes also have the same number of data points. We choose $n_1 = 500, m_1 = 4500$ for CIFAR-10-LT, $n_1 = 150, m_1 = 300$ for CIFAR-100-LT and $n_1 = 1000, m_1 = 4000$ for CIFAR-10-Step and SHVN-Step. Two types of class imbalance for the considered datasets are illustrated in Fig. 5. In Fig. 5a, we set $\gamma_l = \gamma_u = 50, n_1 = 500, m_1 = 4500$. In Fig. 5b, we set $\gamma_l = \gamma_u = 100, n_1 = 1000, m_1 = 4000$. We can also see that each minority class of step-imbalance setting has a very small amount of data in Fig. 5b. Existing SSL algorithms

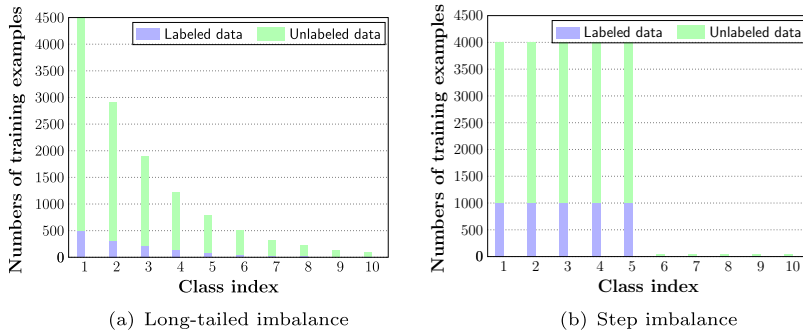


Fig. 5 Long-tailed imbalance and step imbalance

can hardly perform well on minority class under step imbalanced settings due to the scarce data in minority class.

We evaluated the performance of seven algorithms, including:

- WRN-28-2 (Zagoruyko & Komodakis, 2016) (Vanilla algorithm): The basic Deep CNN is trained on only labeled data with the simple cross-entropy loss.
- MiSLAS (Zhong et al., 2021) (CISL algorithm): The SOTA CISL algorithm uses MixUp (Zhang et al., 2019) and label-aware smoothing to handle different degrees of overconfidence for classes and reduce dataset bias by shift learning on the batch normalization layer in the decoupling framework, without using extra unlabeled data.
- MixMatch (Berthelot et al., 2019), ReMixMatch (Berthelot et al., 2020), FixMatch (Sohn et al., 2020) (SSL algorithms): The SOTA SSL algorithms combined consist-

Table 1 Overall accuracy/tail-class(the three classes with least training samples)accuracy with the long tailed imbalanced setting

Algorithm	SSL	CISL	CIFAR-10-LT($\gamma_l = \gamma_u$)		
			$\gamma_l = 50$	$\gamma_l = 100$	$\gamma_l = 150$
Vanilla	–	✓	49.3 ± 1.68 / 23.0 ± 3.44	44.2 ± 0.37 / 10.3 ± 2.22	40.4 ± 1.10 / 5.1 ± 1.83
MiSLAS	–	–	60.0 ± 0.38 / 45.1 ± 2.79	53.0 ± 0.11 / 28.4 ± 1.39	48.6 ± 0.86 / 20.5 ± 2.15
MixMatch	✓	–	62.5 ± 1.46 / 22.1 ± 3.80	56.7 ± 1.05 / 7.6 ± 2.56	52.2 ± 2.07 / 7.9 ± 3.21
FixMatch	✓	–	76.0 ± 0.96 / 52.5 ± 3.67	68.7 ± 0.70 / 35.3 ± 2.58	63.2 ± 0.32 / 20.5 ± 0.20
w/ CReST+	✓	–	81.0 ± 0.51 / 73.4 ± 1.47	74.5 ± 0.61 / 56.1 ± 1.56	72.3 ± 0.70 / 46.3 ± 2.52
w/ DARP	✓	–	79.9 ± 0.12 / 65.2 ± 0.59	73.9 ± 0.96 / 51.0 ± 1.98	68.4 ± 0.23 / 36.5 ± 1.13
w/ ABC	✓	–	82.4 ± 0.51 / 73.4 ± 2.05	77.2 ± 0.54 / 63.4 ± 1.87	73.4 ± 0.81 / 51.6 ± 2.65
w/ Adsh	✓	–	77.8 ± 0.41 / 61.0 ± 0.54	68.9 ± 0.74 / 36.1 ± 1.89	65.1 ± 1.10 / 28.8 ± 1.54
ReMixMatch	✓	–	78.1 ± 0.43 / 59.7 ± 1.78	72.2 ± 0.40 / 45.1 ± 1.99	68.1 ± 0.52 / 35.8 ± 1.63
w/ CReST+	✓	–	80.7 ± 0.61 / 71.2 ± 1.18	74.0 ± 0.64 / 54.9 ± 2.69	69.4 ± 2.54 / 37.5 ± 3.71
w/ DARP	✓	–	78.6 ± 0.30 / 61.3 ± 0.95	72.9 ± 0.62 / 47.1 ± 1.10	68.7 ± 0.81 / 37.0 ± 2.59
w/ ABC	✓	–	84.2 ± 0.20 / 77.9 ± 0.34	79.0 ± 0.29 / 70.7 ± 1.44	77.9 ± 1.02 / 68.5 ± 2.21

SSL denotes semi-supervised learning and CISL denotes class-imbalanced supervised learning

Bold values indicate the best number

ency regularization and pseudo labels have achieved great success in SSL, without taking class imbalance into account.

- DARP (Kim et al., 2020a) (CISSL algorithm): The algorithm uses DARP to refine the pseudo label obtained by SSL algorithms, e.g. FixMatch and ReMixMatch.
- CREST (Wei et al., 2021a) (CISSL algorithm): The algorithm alleviates the class imbalance by selecting pseudo-labeled unlabeled instances classified as minority classes with a higher confidence than those classified as majority classes.
- ABC (Lee et al., 2021) (CISSL algorithm): The algorithm provides auxiliary balanced classifier to rebalance the biased model by introducing extra regularization terms.
- Adsh (Guo & Li, 2022) (CISSL algorithm): The algorithm based on FixMatch uses adaptive class-dependent pseudo-label thresholding to get high quality pseudo-labels.

All experiments are trained with batch size 64 for 250, 000 iterations. We used the Adam optimizer (Kingma & Ba, 2015) with a learning rate of 0.002, and used Cutout (Devries & Taylor, 2017) and RandomAugment (Cubuk et al., 2019) for strong data augmentation, following the approach provided in Lee et al. (2021). As suggested by Berthelot et al. (2019), we evaluated the performance of these algorithms using an exponential moving average of

Table 2 Overall accuracy/tail-class(the three classes with smallest training samples) accuracy under the long tailed setting($\gamma_l \neq \gamma_u$)

Algorithm	SSL	CISL	CIFAR-10-LT($\gamma_l = 100$)		
			$\gamma_u = 1$	$\gamma_u = 50$	$\gamma = 150$
Vanilla	–	–	44.2 ± 0.37 / 10.3 ± 2.22	44.2 ± 0.37 / 10.3 ± 2.22	44.2 ± 0.37 / 10.3 ± 2.22
MiSLAS	–	✓	53.0 ± 0.11 / 28.4 ± 1.39	53.0 ± 0.11 / 28.4 ± 1.39	53.0 ± 0.11 / 28.4 ± 1.39
MixMatch	✓	–	36.7 ± 0.56 / 1.0 ± 0.55	56.6 ± 0.52 / 13.1 ± 2.92	56.2 ± 1.35 / 11.8 ± 3.70
FixMatch	✓	–	65.7 ± 0.52 / 23.1 ± 0.24	71.8 ± 1.12 / 41.2 ± 3.42	67.7 ± 0.77 / 33.3 ± 2.62
w/ CREST+	✓	–	76.1 ± 1.62 / 62.1 ± 3.01	79.4 ± 1.48 / 68.6 ± 0.95	72.1 ± 2.36 / 46.2 ± 4.37
w/ DARP	✓	–	76.7 ± 0.13 / 65.5 ± 0.41	74.3 ± 0.29 / 63.4 ± 0.39	71.1 ± 0.13 / 48.1 ± 0.39
w/ ABC	✓	–	74.7 ± 0.75 / 54.5 ± 2.52	79.2 ± 0.46 / 65.3 ± 1.92	74.7 ± 0.27 / 65.1 ± 1.77
w/ Adsh	✓	–	60.3 ± 0.61 / 20.5 ± 2.49	73.5 ± 0.47 / 50.7 ± 2.23	67.0 ± 0.78 / 34.2 ± 1.71
ReMixMatch	✓	–	45.2 ± 0.85 / 3.4 ± 0.49	73.9 ± 0.40 / 49.4 ± 1.02	68.4 ± 0.98 / 41.3 ± 2.49
w/ CREST+	✓	–	68.0 ± 1.35 / 28.3 ± 4.18	80.2 ± 1.01 / 70.9 ± 1.83	70.4 ± 2.10 / 46.6 ± 3.94
w/ DARP	✓	–	79.3 ± 0.24 / 86.1 ± 0.13	74.9 ± 0.19 / 66.6 ± 0.40	68.9 ± 0.16 / 46.9 ± 0.41
w/ ABC	✓	–	52.0 ± 1.80 / 49.4 ± 10.15	82.6 ± 1.81 / 72.3 ± 0.87	78.5 ± 1.06 / 67.6 ± 4.13

SSL denotes semi-supervised learning and CISL denotes class-imbalanced supervised learning

Bold values indicate the best number

Table 3 Overall accuracy on CIFAR-100-LT

Algorithm	CIFAR-100-LT($\gamma_l = \gamma_u$)				
	FixMatch	w/ CREST+	w/ DARP	w/ ABC	w/ Adsh
$\gamma_l = 10$	55.1 ± 0.19	57.4 ± 0.18	56.3 ± 1.98	58.2 ± 0.49	57.1 ± 0.49
$\gamma_l = 20$	49.5 ± 1.34	52.1 ± 0.21	50.2 ± 0.19	53.1 ± 0.17	50.3 ± 0.35

Bold values indicate the best number

Table 4 Overall accuracy/tail-class(the three classes with smallest training samples) accuracy on CIFAR-10 and SVHN under step imbalanced setting

Algorithm	SSL	CISL	CIFAR-10-Step $\gamma_l = \gamma_u = 100$	SVHN-Step $\gamma_l = \gamma_u = 100$
FixMatch	✓	–	54.0 ± 0.84 / 11.8 ± 1.71	79.8 ± 1.34 / 61.5 ± 2.76
w/ CReST+	✓	–	71.1 ± 0.78 / 48.2 ± 2.26	86.6 ± 0.19 / 76.3 ± 0.23
w/ DARP	✓	–	67.9 ± 1.98 / 43.0 ± 2.12	85.3 ± 0.19 / 67.9 ± 0.40
w/ DARP+cRT	✓	✓	69.8 ± 1.51 / 45.1 ± 2.70	85.9 ± 0.28 / 74.3 ± 0.37
w/ ABC	✓	–	75.9±0.49 / 57.0±1.07	91.2±0.15 / 85.6 ± 0.35
ReMixMatch	✓	–	60.8 ± 0.10 / 25.1 ± 1.28	82.7 ± 0.42 / 67.4 ± 0.81
w/ CReST+	✓	–	64.6 ± 0.97 / 33.5 ± 2.05	85.9 ± 0.13 / 73.9 ± 0.16
w/ DARP	✓	–	71.4 ± 1.97 / 48.8 ± 2.30	89.6 ± 1.08 / 77.4 ± 0.32
w/ DARP+cRT	✓	✓	72.3 ± 1.77 / 50.6 ± 3.53	90.5 ± 1.13 / 84.3 ± 1.86
w/ ABC	✓	–	76.4±1.70 / 65.7±1.30	91.3±1.61 / 89.8±0.95

Bold values indicate the best number

the parameters over iterations with a decay rate of 0.999, instead of scheduling the learning rate. There are many metrics to be used for comparison in an imbalanced setting, such as overall accuracy, F1-score and geometric mean(GM). Most of advanced work uses overall accuracy to measure the model performance due to the balanced test datasets. It can be seen from the Fig. 1b and c that the semi-supervised learning can hardly perform well in tail classes. We also use tail class accuracy to measure the model performance in another perspective in Tables 1, 2, 3 and 4. Each experiment is repeated five times with the long-tailed imbalance setting and three times with the step-imbalance setting. We report the average and standard deviation of the performance measures.

5.2 Results

5.2.1 CIFAR-10-LT under $\gamma_l = \gamma_u$

We first evaluate the algorithms with $\gamma_l = \gamma_u$, which is the most common scenarios that labeled and unlabeled data are sampled from the same distribution. In order to produce convincing results, we compare the existing CISSL algorithms with SSL and CISL algorithms on CIFAR-10-LT with various imbalance ratios. The result is shown in Table 1.

As shown in Table 1, we can observe that CISSL algorithms achieve better overall accuracy than CISL and SSL algorithms, with improved accuracy of the tail classes(the three classes with smallest training samples). MiSLAS, the SOTA CISL algorithm achieves better performance than vanilla algorithm, and worse performance than the SSL algorithms. Although MiSLAS alleviates class imbalance and produces higher accuracy for tail classes, it produces poor overall accuracy, as it doesn't use extra unlabeled data. We can also observe that the SSL algorithm MixMatch has little improvement or even decreased the performance for the tail classes compared with other SSL algorithms. This may be because MixMatch only uses weak augmentation and can not learn a good high-quality representations when training data is too little to learn. Other SSL algorithms, FixMatch and ReMixMatch show better overall accuracy and tail-class accuracy than MiSLAS, even though they don't consider class

imbalance. This illustrates the importance of using extra unlabeled data for training. Interestingly, ReMixMatch achieves better accuracy than FixMatch in different imbalanced ratio γ_l , which is opposite of the result of training with class-balanced data. This may be because only ReMixMatch uses distribution alignment(DA) (Berthelot et al., 2020), which encourages the model predictions to have the same class distribution as the labeled set. The effect of DA can be reflected in the comparison of results between FixMatch+CRcST+ and ReMixMatch+CRcST+. Compared with DARP, Adsh and CRcST+, ABC achieves better performance in both the whole and tail classes. This may be due to the reason that DARP, Adsh, CRcST+ are designed to get more pseudo labels with higher confidence when unlabeled data is imbalanced and the unlabeled data points with pseudo label is too few to rebalance the class-imbalanced distribution.

5.2.2 CIFAR-10-LT under $\gamma_l \neq \gamma_u$

We then evaluate the algorithms with $\gamma_l \neq \gamma_u$, which is not unusual in realistic scenarios where labeled and unlabeled data are sampled from a different distribution. In this case, it is also hard to know the real distribution of unlabeled data. So, for the training model, the imbalance ratio γ_u of unlabeled data is an unknown parameter. The result is shown in Table 2.

Generally, the performance is related to the class-imbalanced ratio of all training data. The accuracy increases as the γ_u decreases, which means the overall distribution of training data becomes more balanced. Surprisingly, in Table 2, when $\gamma_l = 100$ and $\gamma_u = 1$, we can observe that SSL algorithms, MixMatch and ReMixMatch, have little improvement on or even decreased the performance compared with vanilla algorithm and the performance of FixMatch is also decreased compared with the situation $\gamma_l = \gamma_u = 100$. This may be because a significant number of unlabeled data from tail classes are identified as the data from head classes, which leads to the lower confidence of the pseudo labels and degenerates the model performance. This demonstrates that using extra unlabeled data is not always helpful in SSL. Interestingly, unlike the situation in Table 1, the best accuracy in different ratios of γ_u is achieved by different CISSL algorithms. In the case of $\gamma_l = 100$ and $\gamma_u = 1$, where the labeled training data is imbalanced and the unlabeled data is balanced, ReMixMatch+DARP achieves better overall accuracy and tail-class accuracy than ReMixMatch+ABC and ReMixMatch+CRcST+. The way DARP used is designed to cope with the situation $\gamma_l \neq \gamma_u$ and $\gamma_l = \gamma_u$, even ABC and CRcST+ only consider the situation $\gamma_l = \gamma_u$. When unlabeled data is balanced and labeled data is imbalanced, DARP refines the pseudo labels with low confidence according to the real distribution. This may be explained by the fact that DARP can achieve better accuracy with the situation $\gamma_l = 100$ and $\gamma_u = 1$. We can also find that FixMatch+Adsh, the algorithm using adaptive pseudo-label thresholding, has worse result than the baseline due to its adaptive thresholding strategy. The adaptive thresholding may be not available when the distribution of labeled and unlabeled training data is quite different. In the case of $\gamma_l = 100$ and $\gamma_u = 50$, where the labeled training data is imbalanced and the unlabeled data is a little more balanced, ReMixMatch+ABC achieve best overall accuracy and tail-class accuracy. It is interesting to find that FixMatch+CRcST+ achieves better overall accuracy and tail-class accuracy than FixMatch+ABC. This may be because getting pseudo labels with higher confidence can effectively reduce the class imbalance when unlabeled data is a little more balanced than unlabeled data. In the case of $\gamma_l = 100$ and $\gamma_u = 150$, where the labeled training data is imbalanced and the unlabeled data is more imbalanced,

ReMixMatch+ABC achieves better accuracy than other algorithms. When unlabeled data is more imbalanced than labeled data, even if we can get perfect pseudo labels for unlabeled data, it is hard to mitigate the imbalance of true distribution. This may be the reason why DARP, CReST+ and Adsh, the pseudo label based CISSL algorithms, can't achieve the same improvement as ABC.

5.2.3 CIFAR-100-LT under $\gamma_l \neq \gamma_u$

To make a more comprehensive comparison, we evaluate the algorithms on CIFAR-100-LT. Compared with CIFAR-10, CIFAR-100 is more complex and it is hard for the same algorithms to achieve the same results as it in CIFAR-10. We only use overall accuracy to evaluate the performance in this dataset. As the trend in Table 1, the experiment results of CIFAR-100-LT in Table 3 also show that ABC still outperforms other algorithms. Due to the more complex classification task, the improvement of CISSL algorithms is also limited.

5.2.4 CIFAR-10-Step and SVHN-Step under $\gamma_l = \gamma_u$

We also evaluate the algorithms with the step imbalance setting, where half of the classes have few training data. The experiment result of CIFAR-10-Step and SHVN-Step are presented in Table 4.

As shown in Table 4, we observe that ReMixMatch+ABC also achieves best overall accuracy and tail-class accuracy in CIFAR-10-Step and SVHN-Step. Due to the reason that the number of training samples from tail classes in step imbalance is less than that in long-tailed imbalance, the improvement of accuracy caused by CISSL algorithms is more significant.

5.2.5 More quantitative comparison

In order to present more distinct results, we provide the confusion matrices of the selected pseudo-labels on the unlabeled data and confusion matrices of the prediction on the

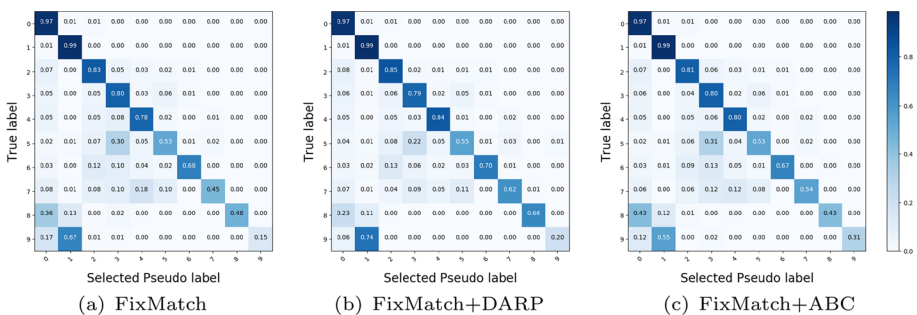


Fig. 6 Confusion matrices of the selected pseudo-labels on the unlabeled data of CIFAR-10-LT under imbalance ratio $\gamma_l = \gamma_u = 100$

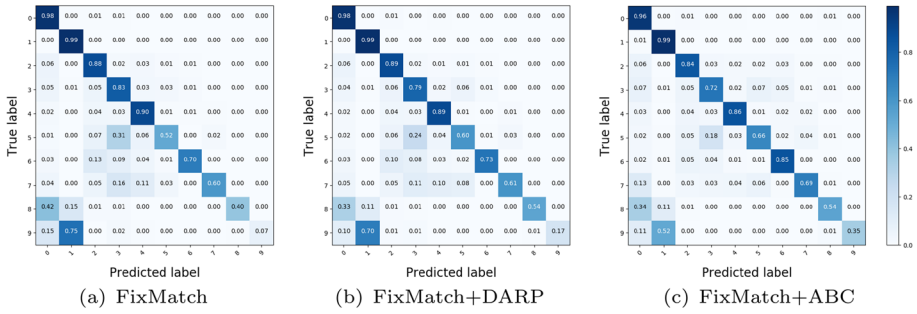


Fig. 7 Confusion matrices of the prediction on the test set of CIFAR-10-LT under imbalance ratio $\gamma_l = \gamma_u = 100$

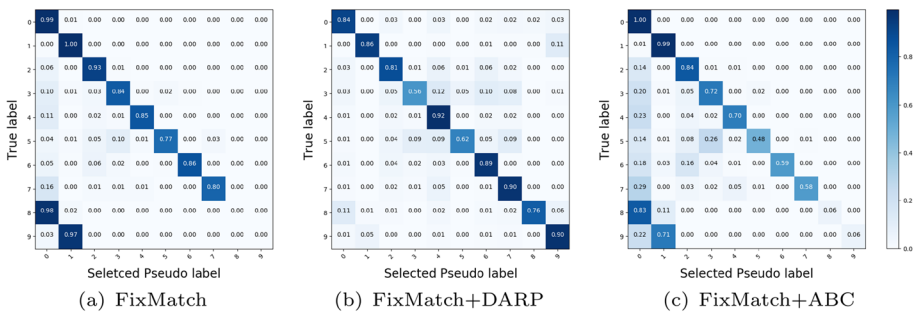


Fig. 8 Confusion matrices of the selected pseudo-labels on the unlabeled data of CIFAR-10-LT under imbalance ratio $\gamma_l = 100, \gamma_u = 1$

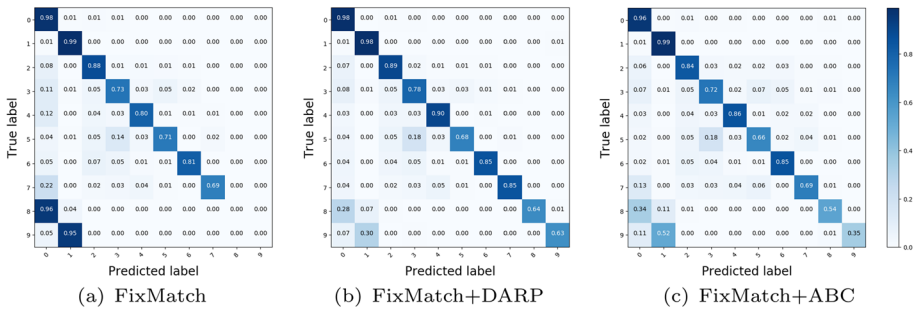


Fig. 9 Confusion matrices of the prediction on the test set of CIFAR-10-LT under imbalance ratio $\gamma_l = 100, \gamma_u = 1$

test set. We consider FixMatch(Sohn et al., 2020), FixMatch+DARP(Kim et al., 2020a) and FixMatch+ABC(Lee et al., 2021) trained on CIFAR-10-LT with $\gamma_l = \gamma_u = 100$ and $\gamma_l = 100, \gamma_u = 1$. The Figs. 6 and 7 show the results of on CIFAR-10-LT with $\gamma_l = \gamma_u = 100$, where the labeled data and unlabeled data are sampled from the same distribution. FixMatch+ABC achieves similar pseudo-labels as FixMatch and FixMatch+DARP achieves more balanced pseudo-labels than FixMatch+ABC and FixMatch. But due to

limited unlabeled data, the balanced classifier learning based FixMatch+ABC can easily produce more balanced results than FixMatch+DARP. The Figs. 8 and 9 show the results of on CIFAR-10-LT with $\gamma_l = \gamma_u = 1$, where the labeled data and unlabeled data are sampled from the different distribution. When unlabeled data is much more balanced than labeled data. FixMatch+DARP achieves more balanced pseudo labels than FixMatch and FixMatch+ABC, leading its better accuracy on test set. As shown in Fig. 9, the pseudo-label generated by FixMatch and FixMatch+ABC are biased towards head classes. The pseudo-label with low quality results that FixMatch and FixMatch+ABC often misclassify test data points in the tail-classes as the data point in the head-classes. In contrast, FixMatch+DARP can achieve more unbiased confusion matrix on selected pseudo-label. It can produce a significantly more balanced class-distribution than FixMatch and FixMatch+ABC with high quality pseudo-label. This result indicates that the improvement of the quality of pseudo-label can effectively improve the performance of model generalization.

5.3 Summary of discussions

According to the results above, some of our finding include:

- SSL methods can effectively improve the model performance by using extra unlabeled data, even in class-imbalanced distribution.
- SSL may degenerate the model performance compared with the vanilla algorithms in some settings, as the reversed bias towards the tail class occurs.
- Different approaches exhibit substantially different levels of sensitivity to the imbalanced ratio of labeled and unlabeled data.
- When the unlabeled data is much more balanced than labeled data, pseudo label based CISSL algorithms achieve better performance as getting a lot of pseudo labels with high confidence can mitigate the imbalance.
- Pseudo label based algorithms can be improved by combining decoupling representation and classifier learning (Kim et al., 2020a; Kang et al., 2020) in class-imbalanced distribution.
- Although CISSL algorithms achieve better performance than SSL algorithms and CISL algorithms, it's still hard to find a general CISSL algorithm that can achieve good results in all scenarios.

6 Challenges and future directions

In this section, we discuss several future research directions for CISSL from perspectives of method innovation and task innovation.

6.1 New methods

6.1.1 Holistic methods for CISSL

The success of holistic methods in SSL (Berthelot et al., 2019, 2020; Sohn et al., 2020) demonstrate the feasibility to unify the current dominant methods in CISSL. Data

augmentation, pseudo-labeling and consistency regularization can be integrated into a single framework (Berthelot et al., 2019). Similarly, the refining of pseudo label and decoupling representation and classifier learning can be combined to get better generalization of learning algorithms, which can be proved by the fact that DARP+cRT (Kim et al., 2020a) can achieve better performance than any single one of them. Hence, how to better use unlabeled data for CISSL is worth further exploring.

6.1.2 Self-supervised learning

Self-supervised learning (Devlin et al., 2018; Brown et al., 2020; Grill et al., 2020; Chen & He, 2021; Tian et al., 2021; He et al., 2021) has attracted great attention for learning useful feature knowledge by only using unlabeled data in recent years. It has also been proved that self-supervised pre-training can benefit both class-imbalanced learning (Yang & Xu, 2020) and semi-supervised learning (Chen et al., 2020). Considering representation learning is fundamental for all deep learning tasks, it is valuable to design better self-supervised learning methods that can resolve multiple CISSL tasks.

6.2 New tasks

6.2.1 Safe class-imbalanced semi-supervised learning

In SSL, it is generally accepted that the learning performance can benefit from training with unlabeled data, especially when labeled data is scarce. However, they are based on a basic assumption that labeled data and unlabeled data come from the same distribution. For example, under the situation where the imbalanced ratio γ_l of labeled data is much larger than the imbalanced ratio γ_u of unlabeled data, the use of unlabeled data can lead to the degeneration of learning performance (Kim et al., 2020a). And, the distribution of imbalanced data is often accompanied by low quality data, such as the data with data noise (Wu et al., 2021; Cao et al., 2021) or label noise (Wei et al., 2021b; Karthik et al., 2021) and unlabeled data contains classes that are not seen in the labeled data (Guo et al., 2020). Most existing algorithms (Kim et al., 2020a; Wei et al., 2021a; Lee et al., 2021), are trained with the assumption that all training data and labels are clean, leading to degeneration of model performance in practical applications. Thus, safe CISSL approaches have practical significance.

6.2.2 Semi-supervised out-of-distribution detection

Class-imbalanced learning can be regarded as a subdomain of out-of-distribution (OOD) generalization (Wald et al., 2021; Wang et al., 2021; Zhang et al., 2021), where the distribution of training data and testing data are different. OOD generalization is closely related to OOD detection (Yang et al., 2021; Fang et al., 2022). The former requires the model being robust to distribution shift while the latter requires the model being aware of the semantic shift. In class-imbalanced semi-supervised learning, the distributions of labeled training data and unlabeled training data can also be different. It requires the training model to be robust to the unlabeled data from different distribution. Several methods about semi-supervised out-of-distribution detection (Guo et al., 2020; Saito et al., 2021; Zhou et al., 2021; Huang et al., 2021; Rizve et al., 2022; He et al.,

2022) have been proposed to detect the data from unknown distribution. But there are great research opportunities on how OOD detection and OOD generalization can better enable each other, in terms of both algorithmic design and comprehensive performance evaluation.

6.2.3 Class-imbalanced weakly supervised learning

SSL is closely related to weakly supervised learning (WSL) (Zhou, 2017). Similar to SSL, WSL is designed to overcome the need for large hand-labeled and expensive training datasets. WSL refers to learning from a large amount of weak supervision data. This includes: incomplete supervision (Chapelle et al., 2006) (e.g., semi-supervised learning), inexact supervision (Dietterich et al., 1997; Foulds & Frank, 2010; Carbonneau et al., 2018) (e.g., multi-instance learning) and inaccurate supervision (Frénay & Verleysen, 2014; Gao et al., 2016) (e.g., label noise learning). Although we have discussed several algorithms for CISSL, but how to achieve good performance for imbalanced WSL is still under study.

6.2.4 Topology-imbalanced semi-supervised learning

Most of imbalanced learning algorithms (Cao et al., 2019; Cui et al., 2019; Huang et al., 2020; Liu et al., 2019) handle quantity-imbalanced learning, where the distribution of training examples from different classes is imbalanced. But graph-structured data (Zhou et al., 2020b) suffers from another aspect of the imbalance problem, the imbalance caused by the asymmetric and uneven topology of labeled nodes (Deli et al., 2021), i.e., labeled nodes are not equal in terms of their structural role in the graph (topology imbalance). The methods for quantity imbalance can be hardly applied to topology imbalance because quantity imbalanced learning usually treats the labeled nodes of the same class as a whole. Thus how to cope with the topology-imbalanced semi-supervised learning remains an open problem.

7 Conclusion

SSL has achieved great successes recently. Existing SSL algorithms often construct a model with a common assumption that the class distribution of the training data is balanced. However, it is well-known that real-world datasets are often imbalanced, which leads to the degeneration of SSL models in realistic tasks. It's worthwhile to make SSL algorithms to cope with the class-imbalanced data distribution. In this survey, we first give a short introduction of SSL and CSSL algorithms. Then, we have comprehensively reviewed several deep CSSL learning methods proposed before 2022. We analyze the SOTA SSL, CSSL and CISSL methods by evaluating them in a unified framework of reimplementation. Following that, we discussed the potential innovation directions for methods and task settings. In the end, we hope that this survey can help push these successes towards the real world.

Author contributions QG: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing-original draft, Writing-review and editing, Visualization. HZ: Validation, Formal analysis. NG: Validation, Formal analysis. BN: Resources, Writing-review and editing, Supervision, Funding acquisition.

Funding This work is supported by National Natural Science Foundation of China (62072326), and the Key Research and Development Plan of Shanxi Province No. 201903D421007 and No. 202102010101004.

Data availability The datasets are the benchmark datasets available online.

Code availability The codes will be available from the corresponding author upon request.

Declarations

Conflict of interest The authors declare that there is no conflict of interest.

Ethics approval Not Applicable.

Consent to participate Not Applicable.

Consent for publication Not Applicable.

References

- Berthelot, D., Carlini, N., & Goodfellow, I. J., et al. (2019). Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019*, NeurIPS, Vancouver, BC, Canada.
- Berthelot, D., Carlini, N., & Cubuk, E. D., et al. (2020). Remixmatch: Semi-supervised learning with distribution matching and augmentation anchoring. In *8th International Conference on Learning Representations, ICLR 2020*, Addis Ababa, Ethiopia.
- Brown, T. B., Mann, B., & Ryder, N., et al. (2020). Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020*, NeurIPS 2020.
- Buda, M., Maki, A., & Mazurowski, M. A. (2018). A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, *106*, 249–259.
- Byrd, J., & Lipton, Z. C. (2019). What is the effect of importance weighting in deep learning? In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019*, 9–15 June 2019, Long Beach, California, USA, Proceedings of Machine Learning Research, vol 97. PMLR, pp 872–881.
- Cao, K., Wei, C., & Gaidon, A., et al. (2019). Learning imbalanced datasets with label-distribution-aware margin loss. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019*, NeurIPS 2019, Vancouver, BC, Canada, pp 1565–1576.
- Cao, K., Chen, Y., Lu, J., et al. (2021). Heteroskedastic and imbalanced deep learning with adaptive regularization. In *9th International Conference on Learning Representations, ICLR 2021*.
- Carboneau, M., Cheplygina, V., Granger, E., et al. (2018). Multiple instance learning: A survey of problem characteristics and applications. *Pattern Recognition*, *77*, 329–353.
- Chapelle, O., Schölkopf, B., & Zien, A. (2006). Introduction to semi-supervised learning. In O. Chapelle, B. Schölkopf, & A. Zien (Eds.), *Semi-Supervised Learning* (pp. 1–12). The MIT Press.
- Chawla, N. V., Bowyer, K. W., Hall, L. O., et al. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, *16*, 321–357.
- Chen, T., Kornblith, S., Swersky, K., et al. (2020). Big self-supervised models are strong semi-supervised learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020*, NeurIPS 2020.
- Chen, X., & He, K. (2021). Exploring simple siamese representation learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021*. Computer Vision Foundation / IEEE, pp. 15750–15758.

- Chou, H., Chang, S., Pan, J., et al. (2020). Remix: Rebalanced mixup. In *Computer Vision - ECCV 2020 Workshops - Glasgow, UK, August 23–28, 2020, Proceedings, Part VI*.
- Ciresan, D. C., Meier, U., & Schmidhuber, J. (2012). Multi-column deep neural networks for image classification. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, Providence, RI, USA. IEEE Computer Society, pp 3642–3649.
- Cover, T. M., & Thomas, J. A. (1999). *Elements of information theory*. Wiley.
- Cubuk, E. D., Zoph, B., & Mané, D., et al. (2018). Autoaugment: Learning augmentation policies from data. CoRR abs/1805.09501.
- Cubuk, E. D., Zoph, B., & Shlens, J., et al. (2019). Randaugment: Practical data augmentation with no separate search. CoRR abs/1909.13719.
- Cui, Y., Jia, M., & Lin, T., et al. (2019). Class-balanced loss based on effective number of samples. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019*. Computer Vision Foundation/IEEE, pp. 9268–9277.
- Dablain, D., Krawczyk, B., & Chawla, N. V. (2022). Deepsmote: Fusing deep learning and smote for imbalanced data. In *EE Transactions on Neural Networks and Learning Systems*, pp. 1–15.
- Deli, C., Yankai, L., & Guangxiang, Z., et al. (2021). Topology-imbalance learning for semi-supervised node classification. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021*.
- Devi, D., Biswas, S. K., & Purkayastha, B. (2017). Redundancy-driven modified tomek-link based under-sampling: A solution to class imbalance. *Pattern Recognition Letters*, 93, 3–12.
- Devlin, J., Chang, M., & Lee, K., et al. (2018). BERT: pre-training of deep bidirectional transformers for language understanding. CoRR abs/1810.04805.
- Devries, T., & Taylor, G. W. (2017). Improved regularization of convolutional neural networks with cutout. CoRR abs/1708.04552.
- Dietterich, T. G., Lathrop, R. H., & Lozano-Pérez, T. (1997). Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1–2), 31–71.
- Edunov, S., Ott, M., & Auli, M., et al. (2018). Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium. Association for Computational Linguistics, pp. 489–500.
- Fan, Y., Dai, D., & Kukleva, A., et al. (2022). Coss1: Co-learning of representation and classifier for imbalanced semi-supervised learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022*, New Orleans, LA, USA. IEEE, pp. 14554–14564.
- Fang, Z., Li, Y., & Lu, J., et al. (2022). Is out-of-distribution detection learnable? CoRR abs/2210.14707.
- Foulds, J. R., & Frank, E. (2010). A review of multi-instance learning assumptions. *The Knowledge Engineering Review*, 25(1), 1–25.
- Frénay, B., & Verleysen, M. (2014). Classification in the presence of label noise: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 25(5), 845–869.
- Gao, W., Wang, L., & Li, Y., et al. (2016). Risk minimization in the presence of label noise. In Schuurmans, D., Wellman, M. P. (Eds). *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, Phoenix, Arizona, USA. AAAI Press, pp 1575–1581.
- Gidaris, S., Singh, P., & Komodakis, N. (2018). Unsupervised representation learning by predicting image rotations. CoRR abs/1803.07728.
- Goodfellow, I. J., Pouget-Abadie, J., & Mirza, M., et al. (2014). Generative adversarial nets. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014*, Montreal, Quebec, Canada, pp 2672–2680.
- Goodfellow, I. J., Bengio, Y., & Courville, A. C. (2016). *Deep Learning*. Adaptive computation and machine learning, MIT Press.
- Grandvalet, Y., & Bengio, Y. (2005). Semi-supervised learning by entropy minimization. In *Actes de CAP 05, Conférence francophone sur l'apprentissage automatique - 2005*, Nice, France.
- Grill, J., Strub, F., & Althé, F., et al. (2020). Bootstrap your own latent - A new approach to self-supervised learning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020*.
- Guo, L., & Li, Y. (2022). Class-imbalanced semi-supervised learning with adaptive thresholding. In *International Conference on Machine Learning, ICML 2022*, Baltimore, Maryland, USA, Proceedings of Machine Learning Research, vol 162. PMLR, pp. 8082–8094.
- Guo, L., Zhang, Z., & Jiang, Y., et al. (2020). Safe deep semi-supervised learning for unseen-class unlabeled data. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020*, Virtual Event, Proceedings of Machine Learning Research, vol 119. PMLR, pp. 3897–3906.
- Gupta, A., Dollar, P., & Girshick, R. (2019). LVIS: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

- Han, T., Gao, J., & Yuan, Y., et al. (2020). Unsupervised semantic aggregation and deformable template matching for semi-supervised learning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020*, NeurIPS 2020.
- He, H., & Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9), 1263–1284.
- He, H., Bai, Y., & Garcia, E. A., et al. (2008). Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*, IEEE, pp. 1322–1328.
- He, K., Chen, X., & Xie, S., et al. (2021). Masked autoencoders are scalable vision learners. CoRR abs/2111.06377.
- He, R., Han, Z., & Lu, X., et al. (2022). Safe-student for safe deep semi-supervised learning with unseen-class unlabeled data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14585–14594.
- Hinton, G. E., Srivastava, N., & Krizhevsky, A., et al. (2012). Improving neural networks by preventing co-adaptation of feature detectors. CoRR abs/1207.0580.
- Huang, C., Li, Y., Loy, C. C., et al. (2020). Deep imbalanced learning for face recognition and attribute prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(11), 2781–2794.
- Huang, Z., Xue, C., & Han, B., et al. (2021). Universal semi-supervised learning. In *Advances in Neural Information Processing Systems*, vol 34. Curran Associates, Inc., pp. 26714–26725.
- Igual, J., Salazar, A., & Safont, G., et al. (2015). Semi-supervised bayesian classification of materials with impact-echo signals. *Sensors* 15(5):11,528–11,550.
- Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Bach, F. R., Blei, D. M. (eds) *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, JMLR Workshop and Conference Proceedings*, vol 37. JMLR.org, pp 448–456.
- Jamal, M. A., Brown, M., & Yang, M., et al. (2020). Rethinking class-balanced methods for long-tailed visual recognition from a domain adaptation perspective. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020*, Seattle, WA.
- Kang, B., Xie, S., & Rohrbach, M., et al. (2020). Decoupling representation and classifier for long-tailed recognition. In *8th International Conference on Learning Representations, ICLR 2020*.
- Karthik, S., Revaud, J., & Boris, C. (2021). Learning from long-tailed data with noisy labels. CoRR abs/2108.11096.
- Kim, J., Hur, Y., & Park, S., et al. (2020a). Distribution aligning refinery of pseudo-label for imbalanced semi-supervised learning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020*, NeurIPS 2020.
- Kim, J., Jeong, J., & Shin, J. (2020b). M2m: Imbalanced classification via major-to-minor translation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020*. Computer Vision Foundation/IEEE, pp. 13893–13902.
- Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015*, Conference Track Proceedings.
- Krizhevsky, A. (2009). *Learning multiple layers of features from tiny images*. Department of Computer Science, University of Tech. rep.
- Lai, Z., Wang, C., & Gunawan, H., et al. (2022). Smoothed adaptive weighting for imbalanced semi-supervised learning: Improve reliability against unknown distribution data. In *Proceedings of the 39th International Conference on Machine Learning, Proceedings of Machine Learning Research*, vol 162. PMLR, pp 11828–11843.
- Laine, S., & Aila, T. (2017). Temporal ensembling for semi-supervised learning. In *5th International Conference on Learning Representations, ICLR 2017*, Conference Track Proceedings.
- Lee, D. H. (2013). Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *In ICML Workshop on Challenges in Representation Learning*.
- Lee, H., Shin, S., & Kim, H. (2021). ABC: auxiliary balanced classifier for class-imbalanced semi-supervised learning. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021*, NeurIPS 2021.
- Lin, J. (1991). Divergence measures based on the shannon entropy. *IEEE Transactions on Information Theory*, 37(1), 145–151.
- Lin, T., Goyal, P., & Girshick, R. B., et al. (2017). Focal loss for dense object detection. In *IEEE International Conference on Computer Vision, ICCV 2017*. IEEE Computer Society, pp. 2999–3007.
- Liu, J., Sun, Y., & Han, C., et al. (2020). 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)*.

- Liu, X., Wu, J., & Zhou, Z. (2009). Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(2), 539–550.
- Liu, Z., Miao, Z., & Zhan, X., et al. (2019). Large-scale long-tailed recognition in an open world. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019*. Computer Vision Foundation / IEEE, pp. 2537–2546.
- Mahajan, D., Girshick, R. B., & Ramanathan, V., et al. (2018). Exploring the limits of weakly supervised pretraining. In *Computer Vision - ECCV 2018 - 15th European Conference, Proceedings, Part II, Lecture Notes in Computer Science*, vol. 11206. Springer, pp. 185–201.
- Miyato, T., Maeda, S., Koyama, M., et al. (2019). Virtual adversarial training: A regularization method for supervised and semi-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8), 1979–1993.
- Netzer, Y., Wang, T., & Coates, A., et al. (2011). Reading digits in natural images with unsupervised feature learning. In *Deep Learning and Unsupervised Feature Learning Workshop, Advances in Neural Information Processing Systems 2011, NeurIPS 2011*.
- Oh, Y., Kim, D. J., & Kweon, I. S. (2021). Distribution-aware semantics-oriented pseudo-label for imbalanced semi-supervised learning. CoRR abs/2016.05682.
- Park, S., Lim, J., & Jeon, Y., et al. (2021). Influence-balanced loss for imbalanced visual classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 735–744.
- Pham, H., Dai, Z., & Xie, Q., et al. (2021). Meta pseudo labels. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021*. Computer Vision Foundation / IEEE, pp. 11557–11568.
- Pouyanfar, S., Tao, Y., Mohan, A., et al. (2018). Dynamic sampling in convolutional neural networks for imbalanced data classification. In *IEEE 1st Conference on Multimedia Information Processing and Retrieval, MIPR 2018*. IEEE, pp. 112–117.
- Rasmus, A., Berglund, M., & Honkela, M., et al. (2015). Semi-supervised learning with ladder networks. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015*, pp. 3546–3554.
- Ren, J., Yu, C., & Sheng, S., et al. (2020). Balanced meta-softmax for long-tailed visual recognition. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020*.
- Rizve, M. N., Kardan, N., & Shah, M., et al. (2022). Towards realistic semi-supervised learning. In S. Avidan, G. Brostow, & M. Cissé (Eds.), *Computer Vision - ECCV 2022* (pp. 437–455). Springer.
- Saito, K., Kim, D., & Saenko, K. (2021). Openmatch: Open-set semi-supervised learning with open-set consistency regularization. In *Advances in Neural Information Processing Systems*, vol 34. Curran Associates, Inc., pp. 25956–25967.
- Sajjadi, M., Javanmardi, M., & Tasdizen, T. (2017). Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016*.
- Salazar, A., Safont, G., & Vergara, L. (2018). Semi-supervised learning for imbalanced classification of credit card transaction. In *2018 International Joint Conference on Neural Networks, IJCNN 2018*. IEEE, pp. 1–7.
- Shen, L., Lin, Z., & Huang, Q. (2016). Relay backpropagation for effective learning of deep convolutional neural networks. In *Computer Vision - ECCV 2016 - 14th European Conference, Proceedings, Part VII, Lecture Notes in Computer Science*, vol. 9911. Springer, pp. 467–482.
- Shu, J., Xie, Q., & Yi, L., et al. (2019). Meta-weight-net: Learning an explicit mapping for sample weighting. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019*, pp. 1917–1928.
- Snell, J., Swersky, K., & Zemel, R. S. (2017). Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017* pp. 4077–4087.
- Sohn, K., Berthelot, D., & Li, C. L., et al. (2020). Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020*.
- Tarvainen, A., & Valpola, H. (2017). Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*.
- Tian, Y., Chen, X., & Ganguli, S. (2021). Understanding self-supervised learning dynamics without contrastive pairs. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 10268–10278.

- Van Horn, G., Mac Aodha, O., & Song, Y., et al. (2018). The inaturalist species classification and detection dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp 8769–8778.
- Wald, Y., Feder, A., & Greenfeld, D., et al. (2021). On calibration and out-of-domain generalization. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021*, pp. 2215–2227.
- Wang, J., Lan, C., & Liu, C., et al. (2021). Generalizing to unseen domains: A survey on domain generalization. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021*. ijcai.org, pp. 4627–4635.
- Wang, Y., Ramanan, D., & Hebert, M. (2017). Learning to model the tail. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, pp. 7029–7039.
- Wei, C., Sohn, K., & Mellina, C., et al. (2021a). Crest: A class-rebalancing self-training framework for imbalanced semi-supervised learning. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2021*.
- Wei, T., Shi, J., & Tu, W., et al. (2021b). Robust long-tailed learning under label noise. CoRR abs/2108.11569.
- Wu, T., Liu, Z., & Huang, Q., et al. (2021). Adversarial robustness under long-tailed distribution. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021*. Computer Vision Foundation/IEEE, pp. 8659–8668.
- Xie, Q., Dai, Z., & Hovy, E. H., et al. (2020). Unsupervised data augmentation for consistency training. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020*.
- Xu, Z., Chai, Z., & Yuan, C. (2021). Towards calibrated model for long-tailed visual recognition from prior perspective. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021*.
- Yang, J., Zhou, K., & Li, Y., et al. (2021). Generalized out-of-distribution detection: A survey. CoRR abs/2110.11334.
- Yang, Y., & Xu, Z. (2020). Rethinking the value of labels for improving class-imbalanced learning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020*.
- Yin, X., Yu, X., & Sohn, K., et al. (2019). Feature transfer learning for face recognition with under-represented data. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019*, Computer Vision Foundation / IEEE, pp. 5704–5713.
- Zagoruyko, S., & Komodakis, N. (2016). Wide residual networks. In *Proceedings of the British Machine Vision Conference 2016, BMVC 2016*.
- Zhai, X., Oliver, A., & Kolesnikov, A., et al. (2019). s4l: Self-supervised semisupervised learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019*. Computer Vision Foundation/IEEE, pp. 2537–2546.
- Zhang, D., Ahuja, K., & Xu, Y., et al. (2021). Can subnetwork structure be the key to out-of-distribution generalization? In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021*, Proceedings of Machine Learning Research, vol. 139. PMLR, pp. 12356–12367.
- Zhang, H., Cissé, M., & Dauphin, Y. N., et al. (2019). mixup: Beyond empirical risk minimization. In *6th International Conference on Learning Representations, ICLR 2018*, Conference Track Proceedings.
- Zhong, Z., Cui, J., & Liu, S., et al. (2021). Improving calibration for long-tailed recognition. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2021*.
- Zhou, B., Cui, Q., & Wei, X., et al. (2020a). BBN: bilateral-branch network with cumulative learning for long-tailed visual recognition. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020*.
- Zhou, J., Cui, G., Hu, S., et al. (2020). Graph neural networks: A review of methods and applications. *AI Open*, 1, 57–81.
- Zhou, Z., Guo, L. Z., & Cheng, Z., et al. (2021). Step: Out-of-distribution detection in the presence of limited in-distribution labeled data. In *Advances in Neural Information Processing Systems*, vol 34. Curran Associates, Inc., pp. 29168–29180.
- Zhou, Z. H. (2017). A brief introduction to weakly supervised learning. *National Science Review*, 5(1), 44–53.
- Zou, Y., Yu, Z., & Liu, X., et al. (2019). Confidence regularized self-training. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South)*. IEEE, pp. 5981–5990.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.