Class-Imbalanced Semi-Supervised Learning with Inverse Auxiliary Classifier

Tiansong Jiang[1,2,3]
jiangtiansong@njust.edu.cn
Sheng Wan[1,2,3 *]
wansheng315@hotmail.com
Chen Gong[1,2,3 *]
chen.gong@njust.edu.cn

[1] School of Computer Science and Engineering, Nanjing University of Science and Technology, China
[2] Key Laboratory of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education, China
[3] Jiangsu Key Laboratory of Image and Video Understanding for Social Security, China

Abstract

In this paper, we focus on the challenge posed by Class-Imbalanced Semi-Supervised Learning (CISSL). Existing pseudo-labeling-based Semi-Supervised Learning (SSL) algorithms often exhibit poor performance in minority classes, thereby resulting in degradation of the feature learning process. This issue becomes more pronounced when labeled and unlabeled data exhibit different class distributions. To mitigate the effect of imbalanced labeled data on feature learning, we introduce a simple yet effective plug-in module, *i.e.*, Inverse Auxiliary Classifier (IAC). The module utilizes a down-sampling strategy by using a mask that inverts the class distribution of labeled data. Additionally, we propose an Inverse Distribution Alignment (IDA) loss to encourage IAC to focus on the underrepresented minority classes in labeled data. The proposed method can be seamlessly integrated into multiple existing CISSL algorithms without any difficulty. Extensive experiments conducted in this paper demonstrate that incorporating the proposed IAC can improve the performance of different CISSL models, especially when there is a significant disparity between the class distributions of labeled and unlabeled data.

1 Introduction

Semi-Supervised Learning (SSL) has garnered significant attention in recent years, as it has demonstrated effective performance [2, 3, 9, 23, 35, 38, 39, 42, 45, 47] by leveraging both labeled and unlabeled data. Many SSL methods employ pseudo-labeling techniques [29] and consistency regularization [44] to utilize unlabeled data. Nevertheless, state-of-the-art SSL algorithms [38, 42, 47] typically assume a balanced class distribution in training data, which may not hold in real-world application [1, 20, 21]. Models trained on imbalanced data are very likely to be biased toward the majority classes with more examples while neglecting

*Corresponding authors.

the minority classes with fewer examples, resulting in poor performance [5, 22, 43]. Though there are some mature methods for Class-Imbalanced Learning (CIL) [4, 7, 8, 12, 14, 17, 19, 37, 49, 50], they are heavily dependent on accurate labels. This is not always feasible in SSL, as accurate annotations can be quite scarce [15, 22].

Recently, various strategies have been proposed for tackling the challenge of Class-Imbalanced SSL (CISSL), including refining the distribution of pseudo-labels [22], expanding the labeled set with pseudo-labeled examples [43], or utilizing representations learned from existing SSL methods, to learn a balanced classifier [15, 30, 34]. Broadly, these methods aim to train a balanced classifier on pseudo-labeled examples derived from conventional SSL methods, under the assumption of consistent class distributions between labeled and unlabeled data. Figure 1(a) portrays a scenario where the class distribution of labeled data is consistent with that of unlabeled data. Here, the aforementioned CISSL methods usually exhibit good performance. These methods predominantly focus on mitigating the bias towards majority classes by refining the pseudo-labels. However, they demand prior knowledge of the class distribution of unlabeled data, which are often unavailable in practical use [6, 24, 25]. Unlabeled data belonging to the minority classes of the imbalanced labeled data are prone to being erroneously assigned with pseudo-labels of majority classes. This misclassification of pseudo-labels can result in the confusion of features from the corresponding classes, hampering the performance of feature learning, especially when there exists significant divergence in class distributions between labeled and unlabeled data as depicted in Figure 1(d).

To mitigate this issue, we introduce a simple and efficient plug-in module, *i.e.*, Inverse Auxiliary Classifier (IAC), to enhance the performance of CISSL algorithms. IAC employs a down-sampling strategy that can be performed independently of the prior knowledge of class distribution, aiming to emphasize the importance of minority classes during the feature learning process. Additionally, we present an Inverse Distribution Alignment (IDA) loss to extract sufficient supervision from the down-sampled labeled data. During model training, the proposed IAC serves as a regularization term, assisting the feature extractor in discriminating the minority and majority classes in labeled data. This regularization can help refine the pseudo-labels produced by FixMatch [38], a widely used SSL algorithm serving as the foundation for many CISSL algorithms, as shown in Figure 1(b) and Figure 1(e) respectively. It is noteworthy that the proposed IAC can be adopted to enhance the performance of most existing CISSL algorithms, e.g., Adsh [16], as shown in Figure 1(c) and Figure 1(f). The improvement can be especially notable when there is a large distribution gap between labeled and unlabeled data. The contributions of this paper are as follows:

- We introduce IAC, which is a novel and effective plug-in module for existing CISSL algorithms.

- The proposed IDA loss is utilized to enrich the limited supervision, facilitating IAC in effectively learning from the minority-class data.

- Through comprehensive experiments, we demonstrate that the proposed IAC improves the performance of various existing CISSL models, especially when there is a significant discrepancy between the class distributions of labeled and unlabeled data.

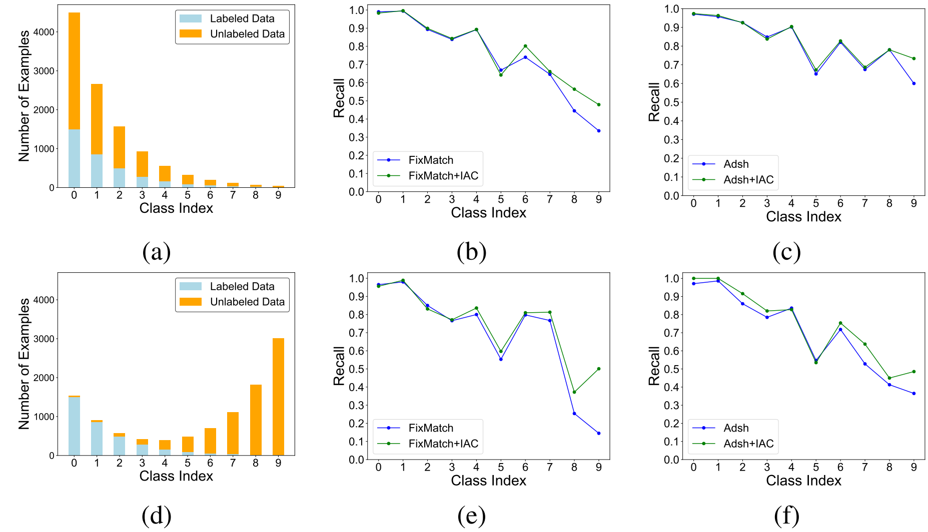


Figure 1: Learning with consistent and inconsistent class distributions between labeled and unlabeled examples. (a) Consistent class distributions between labeled and unlabeled examples. (b)-(c) Recall of the generated pseudo-labels and predicted labels under consistent class distributions, respectively. (d) Inconsistent class distributions between labeled and unlabeled examples. (e)-(f) Recall of the generated pseudo-labels and predicted labels under inconsistent class distributions, respectively.

2 Related Work

Semi-Supervised Learning (SSL): SSL aims to utilize both labeled and unlabeled data for learning. Significant progress has been made in SSL through methods such as pseudo-labeling and consistency regularization [2, 3, 27, 36, 38, 41, 42, 47]. FixMatch [38] is a representative SSL algorithm that uses weak and strong data augmentations along with a fixed high confidence threshold for generating pseudo-labels and enforcing consistency. FlexMatch [47] and FreeMatch [42] further extend SSL by using dynamic threshold strategies for pseudo-label filtering. However, these methods assume uniform label distribution, which is unrealistic in practice due to prevalent class imbalance in labeled and unlabeled data.

Class-Imbalanced Learning (CIL): CIL, also known as long-tailed learning, aims to address classification problems with imbalanced training data. Prior research has primarily focused on one-stage techniques, including re-sampling methods or re-weighting loss functions during training. Re-sampling strategies [4, 8, 14, 17, 31, 37, 40] balance the number of training examples across classes, while re-weighting methods [7, 12] adjust the loss for each class based on a factor inversely proportional to the number of data points in the corresponding class. Recently, two-stage methods [19, 32, 50] have improved the long-tail prediction by separately learning the feature representations and the classifier head. However, applying these techniques to imbalanced SSL is challenging due to the lack of labeled data.

Class-Imbalanced Semi-Supervised Learning (CISSL): Recently, various methods [15, 16, 18, 22, 28, 34, 43] have been proposed to address class imbalance in SSL. DARP [22] introduces a convex optimization method to refine raw pseudo-labels using the class distri-

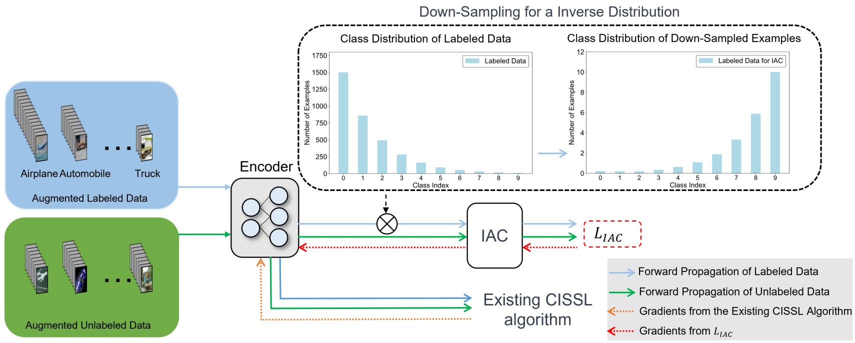


Figure 2: The overall framework of the CISSL method that incorporates our proposed IAC.

bution of unlabeled data, while DASO [34] combines linear and similarity-based classifiers for pseudo-label prediction. Adsh [16] optimizes an adaptive threshold for each class, and CReST [43] proposes a re-sampling method for model refinement. These approaches depend on the classifier's ability to accurately predict unlabeled data. Assuming that labeled and unlabeled data share consistent class distributions, and motivated by the effectiveness of SSL methods like FixMatch [38] in feature extraction from training data, ABC [30] employs an existing SSL algorithm to obtain high-quality representations and simultaneously trains a class-balanced classifier for prediction through down-sampling. Meanwhile, CoSSL [15] initially trains a model using an existing SSL algorithm and then trains a balanced classifier based on the features of both labeled and unlabeled examples. However, SSL classifiers tend to favor majority classes in labeled data when predicting, resulting in substantial pseudo-label biases and ineffective learning of minority classes. This issue becomes more pronounced when the class distributions of labeled and unlabeled data are inconsistent, emphasizing the need to supplement the model's learning of minority classes in labeled data.

Notably, the model structure of IAC shares similarities with that of ABC [30], but there are fundamental differences between the two methods. First, ABC aims to train a balanced classifier for prediction, while the goal of IAC is to alleviate the impact of the imbalanced class distribution on feature learning. Second, ABC down-samples both labeled and unlabeled data to achieve a uniform distribution, while IAC down-samples labeled data to achieve an inverse class distribution.

3 Methodology

3.1 Preliminary

We consider a K-class classification problem with a labeled set $\mathcal{X} = \{(\mathbf{x}_n, y_n)\}_{n=1}^{N}$ and an unlabeled set $\mathcal{U} = \{\mathbf{u}_m\}_{m=1}^{M}$. Each $\mathbf{x}_n \in \mathbb{R}^d$ (d representing the feature dimension) is a training example and $y_n \in \{1,...,K\}$ indicates the corresponding class label. We use N_k and M_k to represent the number of labeled and unlabeled instances, respectively, belonging to class k, where $\sum_{k=1}^{K} N_k = N$ and $\sum_{k=1}^{K} M_k = M$. Without loss of generality, we assume that the K classes in $\mathcal{X}$ are arranged in descending order based on their cardinality, *i.e.*,

$N_1 \geq N_2 \geq ... \geq N_K$. The class distribution of $\mathcal{U}$ is usually unknown in practice. For each training iteration, minibatches $\mathcal{B}_{\mathcal{X}} = \{(\mathbf{x}_b, y_b)\}_{b=1}^{B}$ and $\mathcal{B}_{\mathcal{U}} = \{(\mathbf{u}_b)\}_{b=1}^{\mu B}$ are generated by sampling from $\mathcal{X}$ and $\mathcal{U}$, with minibatch sizes of B and μB, respectively. Our objective is to train a classifier $f : \mathbb{R}^d \rightarrow \{1, ..., K\}$ under a class-balanced test set $\mathcal{T}$.

The proposed IAC is applied to the representation layer of the backbone, aiming at mitigating the prediction errors in pseudo-labels of unlabeled examples, which are caused by inconsistent class distributions between labeled and unlabeled data. It can be effortlessly integrated into various CISSL methods based on FixMatch, which is a fundamental component of numerous CISSL algorithms. For simplicity, we elucidate our algorithm using FixMatch as the backbone. FixMatch employs a supervised loss derived from the weakly augmented labeled data points, e.g., $\alpha(\mathbf{x}_b)$, which are produced by flipping and cropping the image. It also utilizes the consistency regularization loss computed from the weakly augmented unlabeled data point $\alpha(\mathbf{u}_b)$ and strongly augmented unlabeled data point $\mathcal{A}(\mathbf{u}_b)$, generated by Cutout [13] and RandomAugment [11]. Specifically, FixMatch generates pseudo-labels based on the model's predictions on $\alpha(\mathbf{u}_b)$, retaining the pseudo-label only if a high-confidence prediction is produced by the model. The proposed IAC is trained by reusing the weakly and strongly augmented data, together with the IAC loss L_{IAC}. The overall framework of the CISSL method that incorporates our proposed IAC is illustrated in Figure 2.

3.2 Inverse Auxiliary Classifier

To bolster the feature learning for minority classes in labeled data, we generate a $0/1$ mask $M(\mathbf{x}_b)$ for each labeled data point $\mathbf{x}_b$ based on a Bernoulli distribution $B(\cdot)$, which aims to inverse the class distribution of the labeled data via down-sampling. To ensure sufficient labeled examples for each class during the training of IAC, δ is adopted, to determine the minimum number of examples selected from majority-class data. The supervised loss L_s for IAC is expressed as follows:

$$L_s = \frac{1}{B} \sum_{b=1}^{B} M(\mathbf{x}_b) H(p(y_b), q_b), \tag{1}$$

$$M(\mathbf{x}_b) = Ber(max(\frac{\delta}{N_{y_b}}, (\frac{N_K}{N_{y_b}})^{\rho})), \tag{2}$$

where $p(y_b)$ determines the class distribution for y_b, *i.e.*, the one-hot label. $q_b = p_i(y \mid \alpha(\mathbf{x}_b))$, where $p_i(y \mid \alpha(\mathbf{x}_b))$ represents the predicted class distribution produced by IAC for $\alpha(\mathbf{x}_b)$. The cross-entropy between probability distributions p and q is denoted by $H(p,q)$. δ is the sampling lower bound for each class, which is typically set within a range between 0.1 and 1.0. The parameter ρ quantifies the degree to which the class distribution of the down-sampled labeled data is inverted compared to the original labeled data distribution, which can be set within a range between 1.5 and 2.0. Hence, IAC can not only focus on learning minority-class features, but also take into account the majority-class features, ensuring that different categories can be well-represented in the feature learning process. Additionally, we employ a consistency loss, denoted as L_u, which is calculated in the same manner as the unlabeled loss in the backbone, to encourage the proposed IAC to independently train a high-quality classifier that places more emphasis on the minority classes in the labeled data compared to the classifier in the backbone. The consistency loss L_u for IAC is expressed as follows:

$$L_u = \frac{1}{\mu B} \sum_{b=1}^{\mu B} \mathbb{1}(max(q_b) \geq \tau) H(p_i(y \mid A(\mathbf{u}_b)), \hat{q}_b), \tag{3}$$

where $\mathbb{1}$ is the indicator function and $\hat{q}_b = argmax(q_b)$. $max(q_b)$ is the highest predicted assignment probability, corresponding to the confidence of prediction, and τ is the confidence threshold, which is typically set to 0.95 following FixMatch [38].

3.3 Inverse Distribution Alignment Loss

Since the labeled data used for training after down-sampling are limited, the proposed IDA loss L_{IDA} is applied to enrich the supervision information for the training of IAC through an inverse class distribution. Besides, IDA loss encourages IAC to focus on learning the minority class of labeled data. Unlike traditional distribution alignment strategies [3, 43] which align predictions with the distribution of labeled data, we align IAC's predictions with a distribution that is opposite to the class distribution of original labeled data. This not only expedites the early stages of IAC's training process but also further mitigates the impact of class imbalance in the labeled data on IAC. IDA loss for IAC is expressed as follows:

$$L_{IDA} = H(\tilde{p}(y), \frac{\sum_{b=1}^{\mu B} q_b}{B}), \tag{4}$$

where $\tilde{p}(y) = p((\frac{N_K}{N_y})^{\rho-1})$, which indicates the ideal distribution of the down-sampled labeled data in IAC.

The backbone's performance is improved through end-to-end training of the IAC. We train the proposed model using the total losses from the following loss:

$$L = L_{CISSL} + L_{IAC}, \tag{5}$$

where $L_{IAC} = L_s + \lambda_1 L_u + \lambda_2 L_{IDA}$, and λ_1, λ_2 are fixed scalar hyperparameters denoting the relative weight of L_u and L_{IDA}. L_{CISSL} denotes the loss from the CISSL algorithm. Finally, class labels can be predicted by using the classifier in the CISSL algorithm.

4 Experiments

4.1 Experimental setup

Imbalanced Datasets. We have evaluated our approach on CIFAR-10/100 [26], SVHN [33], and STL-10 [10] datasets, which are widely used in CISSL tasks. The evaluation metrics employed are the overall accuracy (ACC) and F1 Score (F1). Imbalanced versions of these datasets are created by exponentially decreasing the number of examples per class [12]. Following [22, 34], to construct the class-imbalanced training dataset, we use two parameters to denote the imbalance ratio of labeled and unlabeled data, *i.e.*, $\gamma_l = \frac{N_1}{N_k}$ and $\gamma_u = \frac{M_1}{M_k}$, respectively, and we then have $N_k = N_1 \cdot \gamma_l^{-\frac{k-1}{K}}$ and $M_k = M_1 \cdot \gamma_u^{-\frac{k-1}{K}}$ for $1 < k \leq K$.

Compared Methods. We have integrated IAC into six CISSL methods, as well as the original SSL method, *i.e.*, FixMatch [38], and assessed their performance both with and without the proposed IAC. The CISSL methods examined include DARP [22], ABC [30],

Algorithm	CIFAR-10							
	$N_1 = 500, M_1 = 4000$					$N_1 = 1500, M_1 = 3000$		
	$\gamma_l = 150$	$\gamma_l = 100$				$\gamma_l = 150$	$\gamma_l = 100$	
	$\gamma_u = 150$	$\gamma_u = 150$	$\gamma_u = 100$	$\gamma_u = 1$	$\gamma_u = \frac{1}{100}$	$\gamma_u = 150$	$\gamma_u = 100$	$\gamma_u = \frac{1}{100}$
Supervised	43.66/36.60	46.47/41.46	46.47/41.46	46.47/41.46	46.47/41.46	60.37/57.09	63.39/61.14	63.39/61.14
FixMatch	64.10/59.14	71.47/70.04	74.17/73.43	77.66/73.30	60.05/54.78	73.20/72.30	77.66/76.21	71.57/69.78
w/IAC	**67.81/64.92**	**73.26/72.42**	**74.72/74.02**	**78.54/78.27**	**65.26/61.68**	**74.44/73.80**	**78.10/77.75**	**75.27/74.51**
CReST	72.28/70.79	73.66/72.48	78.72/78.45	92.86/92.20	71.80/68.35	74.00/73.21	79.22/78.48	86.80/86.61
w/IAC	**73.03/71.62**	**75.71/72.97**	**79.02/78.12**	**92.89/92.26**	**79.47/76.96**	**75.22/73.38**	**79.77/79.65**	**87.61/87.44**
Adsh	67.47/64.26	73.03/72.23	76.34/75.95	82.22/82.57	66.91/66.39	74.00/73.09	78.15/77.70	70.09/69.33
w/IAC	**70.39/68.71**	**74.75/74.00**	**77.23/76.88**	**85.51/85.76**	**71.17/71.15**	**74.35/73.60**	**79.40/79.14**	**74.09/74.11**
DARP	68.99/**67.05**	74.10/73.22	75.34/74.81	71.69/68.51	61.44/56.19	**74.34**/72.41	78.27/77.90	69.31/67.34
w/IAC	**69.01**/66.50	**74.59/73.85**	**75.85/75.32**	**84.09/83.96**	**64.87/61.05**	74.01/**73.18**	**78.92/78.62**	**75.31/74.81**
DASO	64.84/62.03	67.71/66.34	69.87/69.09	71.57/70.75	69.77/69.44	72.68/**72.12**	76.90/76.63	78.03/78.14
w/IAC	**67.13/65.67**	**68.02/66.65**	**71.19/70.64**	**74.55/74.37**	**78.80/78.96**	**72.70**/71.99	**77.09/76.86**	**81.65/81.78**
ABC	76.41/75.78	80.63/80.49	81.88/81.79	87.07/87.09	79.43/79.22	80.73/80.67	84.25/84.23	83.76/83.54
w/IAC	**78.54/78.24**	**80.69/80.53**	**81.93/81.82**	**88.32/87.49**	**81.94/81.57**	**82.05/82.03**	**84.49/84.48**	**84.65/84.59**
CoSSL	76.95/76.98	80.68/80.79	82.55/82.52	85.98/85.95	72.27/72.04	**83.13/83.30**	84.98/85.03	74.12/73.22
w/IAC	**78.79/78.94**	**80.81/80.99**	**82.85/82.91**	**86.89/86.88**	**76.23/76.07**	83.11/83.20	**85.37/85.41**	**76.64/75.77**

Table 1: Comparison results (ACC/F1) on CIFAR-10.

CReST [43], DASO [34], Adsh [16], and CoSSL [15]. Additionally, we have conducted comparative analysis using supervised learning as a reference baseline.

Training and evaluation. Our experiments are conducted under the uniform codebase USB [41] for fair comparison, with experimental setups mirroring those utilized in DASO [34]. Specifically, the selected backbone network is the Wide ResNet-28-2 [46]. We employ an SGD optimizer with a learning rate of 0.03 and a weight decay of 5e-4. The optimizer operates over 256 training epochs, each comprising 1024 iterations. The batch size is set to 64 for the labeled set and 128 for the unlabeled set. For evaluation, we employ the Exponential Moving Average (EMA) network, updating its parameters at each step, in line with DARP [22].

4.2 Main Results

First, we apply IAC as a plug-in module to FixMatch and various CISSL methods respectively, assessing its effectiveness on CIFAR-10 under varying levels of imbalance ratio within both labeled and unlabeled data. The results of these experiments are detailed in Table 1. Overall, the results demonstrate that these CISSL methods with IAC consistently outperform their counterparts without IAC, particularly when the distributions of labeled and unlabeled data are notably inconsistent. For instance, when $N_1 = 500$, $M_1 = 4000$ and the class distribution of labeled examples is opposite to that of unlabeled examples ($\gamma_l = 100$ and $\gamma_u = \frac{1}{100}$), DASO with IAC demonstrates a 9% increase in terms of accuracy compared with their original version without IAC. Similarly, CoSSL with IAC exhibits a 4% increase in terms of accuracy compared with its original version without IAC. It is also observable from the results that, due to the utilization of unlabeled data, SSL methods significantly outperform supervised learning.

Meanwhile, we conduct extensive experiments on SVHN, CIFAR-100 and STL-10 to demonstrate the effectiveness of the proposed IAC. The experimental results are reported in Table 2, showing the same trend as those for CIFAR-10. As is observed, these CISSL algorithms with IAC achieve competitive or even better performance when compared with

Algorithm	SVHN		CIFAR-100				STL-10	
	$N_1 = 500, M_1 = 4000$		$N_1 = 150, M_1 = 300$				$N_1 = 150, M = 100k$	
	$\gamma_l = 100$		$\gamma_l = 10$		$\gamma_l = 15$		$\gamma_l = 10$	$\gamma_l = 20$
	$\gamma_u = 100$	$\gamma_u = \frac{1}{100}$	$\gamma_u = 10$	$\gamma_u = \frac{1}{10}$	$\gamma_u = 15$	$\gamma_u = \frac{1}{15}$	$\gamma_u : N/A$	$\gamma_u : N/A$
Supervised	82.29/79.73	82.29/79.73	48.23/46.61	48.23/46.61	45.92/43.80	45.92/43.80	46.42/44.62	40.04/35.76
FixMatch	91.93/90.73	89.72/88.20	57.96/56.47	56.91/56.06	54.50/52.34	53.78/52.38	65.61/63.97	54.56/48.89
w/IAC	**91.95/90.67**	**92.66/92.34**	**58.58/57.22**	**59.33/58.66**	**55.74/53.80**	**56.40/55.44**	**67.01/64.94**	**55.85/50.95**
CReST	93.12/**91.56**	92.13/**91.72**	57.09/55.55	59.17/58.33	54.54/52.08	56.38/55.38	67.01/61.94	59.23/54.46
w/IAC	**93.27**/90.78	**92.47**/90.98	**58.57/57.23**	**61.06/60.26**	**54.90/52.14**	**58.57/57.55**	**67.73/62.15**	**60.26/54.57**
Adsh	92.32/91.29	87.60/87.08	58.20/57.21	55.22/54.74	54.14/52.57	51.80/50.92	69.68/70.03	65.50/64.68
w/IAC	**92.34/91.36**	**91.48/91.10**	**58.33/57.33**	**57.25/56.95**	**54.53/52.83**	**54.39/53.83**	**70.48/70.35**	**66.33/64.94**
DARP	91.81,90.60	91.95/90.99	57.88/56.69	57.82/56.91	54.49/52.61	54.69/53.43	63.94/61.85	55.26/51.81
w/IAC	**91.84/90.74**	**93.27/92.94**	**58.55/57.43**	**59.13/58.45**	**55.16/53.13**	**56.94/55.93**	**66.83/65.63**	**57.63/54.28**
DASO	88.59/87.19	89.54/88.72	58.67/56.51	59.31/58.22	55.11/52.37	56.18/54.72	69.38/68.54	58.08/53.66
w/IAC	**89.01/87.76**	**93.00/92.67**	**58.77/57.50**	**59.70/59.06**	**55.56/53.48**	**56.89/55.87**	**70.55/69.62**	**62.54/60.43**
ABC	93.75/**93.12**	92.74/**92.39**	59.83/58.64	59.88/59.02	**56.87/55.47**	57.25/56.36	70.83/69.92	65.69/63.87
w/IAC	**93.76**/93.01	**92.76**/92.27	**60.15/59.14**	**60.51/59.81**	56.73/55.35	**57.73/56.72**	**71.99/71.16**	**67.64/66.81**
CoSSL	92.68/91.69	90.27/88.03	58.58/57.58	58.11/57.45	56.21/55.31	55.56/54.53	70.75/70.15	64.85/64.09
w/IAC	**92.97/92.29**	**92.46/91.57**	**59.45/58.53**	**59.67/59.10**	**56.43/55.62**	**56.33/55.51**	**71.14/70.54**	**65.18/64.34**

Table 2: Comparison results (ACC/F1) on SVHN, CIFAR-100 and STL-10.

the corresponding models. As the distribution of labeled data in CIFAR-100 is relatively balanced overall, the improvement gained by IAC is less pronounced. It is noteworthy that some of the unlabeled data of SLT-10 belong to unknown classes, yet our method can still improve the accuracy of the CISSL algorithms.

4.3 Detailed Analysis

We visualize the training curves for the test accuracy throughout the training phase for each CISSL algorithm on CIFAR-10 dataset with $N_1 = 500$, $M_1 = 4000$, $\gamma_l = 100$ and $\gamma_u = \frac{1}{100}$ in Figure 3. Besides, we visualize the representations of testing data through t-SNE, as depicted in Figure 4. As shown in Figure 3, IAC substantially boosts the test accuracy for the six CISSL methods. As depicted in Figure 4, IAC can improve the quality of features produced by both traditional SSL and CISSL methods. Specifically, IAC can enhance the discriminability of the majority and minority classes in the traditional SSL method, *i.e.*, FixMatch [38], by observing Figure 4(a) and Figure 4(b). Simultaneously, in the comparison between Figure 4(c) and Figure 4(d), IAC mitigates the degradation of feature quality caused by adjusting pseudo-labels in CISSL methods, e.g., DASO [34].

4.4 Ablation Study

To investigate the effects of each component of IAC, we conduct an ablation study on CIFAR-10 with $N_1 = 500$, $M_1 = 4000$, and $\gamma_l = 100$, taking CoSSL+IAC as an example. The results of CoSSL and CoSSL+IAC are presented in Table 3, where each row signifies the application of IAC in the FixMatch-based CoSSL algorithm. The default configuration is $\rho = 1.9$, $\delta = 0.1$, $\lambda_1 = 1.0$ and $\lambda_2 = 0.003$. According to [15], CoSSL employs a two-stage training approach. In the first stage, the model is trained directly through FixMatch. Subsequently, techniques like mixup [48] are implemented to train a balanced classifier based on the features obtained in the previous stage. Based on Table 3, We can draw the following conclusions: 1) The statistics in the table illustrate that the incorporation of L_u can signif-

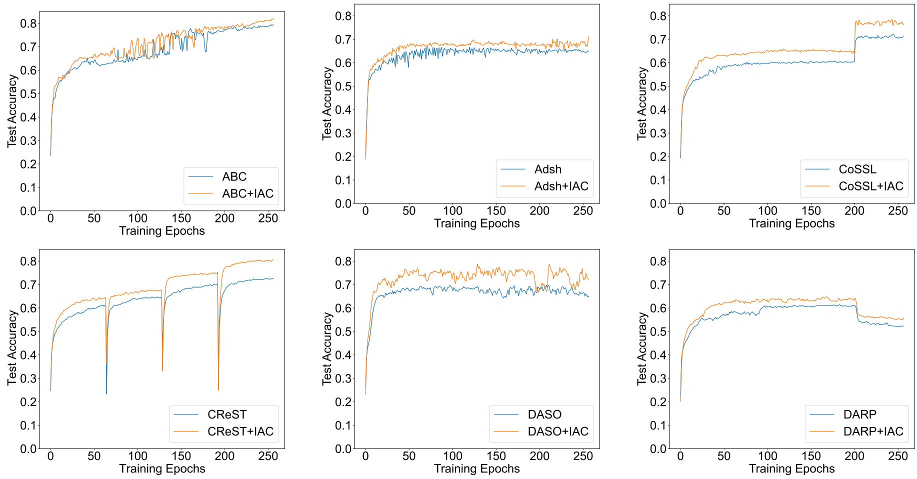
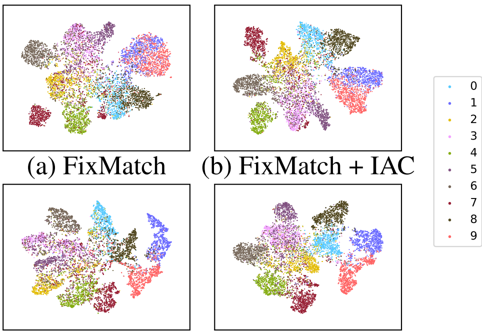


Figure 3: Training curves for the test accuracy on CIFAR-10, where $N_1 = 500$, $M_1 = 4000$, $\gamma_l = 100$ and $\gamma_u = \frac{1}{100}$.

Algorithm	CIFAR-10	
	$N_1 = 500, M_1 = 4000$	
	$\gamma_l = 100$	
	$\gamma_u = 100$	$\gamma_u = \frac{1}{100}$
CoSSL	82.55	72.27
CoSSL+IAC	**82.79**	**76.23**
Without L_u	81.80	71.63
Without L_{IDA}	82.65	74.63
$\lambda_1 = 2.0$	82.41	**77.53**
$\lambda_2 = 1.0$	82.47	75.37
$\rho = 1.0$	82.58	72.16
$\rho = 2.5$	82.78	75.83
Without δ	82.74	75.96
$\delta = 1.0$	82.76	70.34
Without δ, $\rho = 2.5$	82.10	73.21

Table 3: The ablation study for our method on CoSSL on CIFAR-10



(a) FixMatch (b) FixMatch + IAC

(c) DASO (d) DASO + IAC

Figure 4: t-SNE visualization of testing data on CIFAR-10, $N_1 = 500$, $M_1 = 4000$, $\gamma_l = 100$ and $\gamma_u = \frac{1}{100}$.

icantly boost the model's performance. Besides, the enhancement provided by IAC is not dependent on fine-tuning of hyperparameters. 2) By comparing the results across different columns, it is revealed that the application of IAC is particularly beneficial when the class distributions of labeled and unlabeled data are inconsistent. 3) By comparing the results on the second, ninth, and final rows, we find that choosing an appropriate value for δ can lead to a satisfactory prediction, when adopting an excessively large ρ. 4) By comparing the results on the second and fifth rows, we can conclude that a relatively large λ_1 can improve the performance in case of inconsistent class distributions, but it will slightly decrease the performance in case of consistent class distributions. 5) By comparing the results on the second, fourth, and sixth rows, it can be inferred that a suitable λ_2 is helpful to improve the model performance.

5 Conclusion

In this paper, we propose a simple yet effective plug-in module, *i.e.*, IAC, to enhance the performance of existing CISSL algorithms particularly in situations with substantial inconsistency between the class distributions of labeled and unlabeled data. With the assistance of the inverse distribution alignment loss and the inverse auxiliary classifier, IAC focuses on minority classes in labeled data that are often neglected by the backbone. Extensive experiments have demonstrated that existing CISSL methods tend to underperform when the class distributions of labeled and unlabeled data are inconsistent, and the application of IAC can effectively mitigate this issue. Despite the effectiveness, one potential limitation of IAC is the difficulty in determining the optimal hyperparameters. In the future, we plan to reduce the reliance on hyperparameters for IAC and expand its application to a broader range of scenarios.

Acknowledgement

This research is supported by NSF of China (Nos: 62336003, 12371510), NSF of Jiangsu Province (No: BZ2021013), NSF for Distinguished Young Scholar of Jiangsu Province (No: BK20220080), the Fundamental Research Funds for the Central Universities (Nos: 30920032202, 30921013114), and CAAI-Huawei MindSpore Open Fund.

References

[1] Samy Bengio. Sharing representations for long tail computer vision problems. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, pages 1–1, 2015.

[2] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. *Advances in neural information processing systems*, 32, 2019.

[3] David Berthelot, Nicholas Carlini, Ekin Dogus Cubuk, Alexey Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution matching and augmentation anchoring. In *International Conference on Learning Representations*, 2020.

[4] Mateusz Buda, Atsuto Maki, and Maciej A Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural networks*, 106:249–259, 2018.

[5] Saul Calderon-Ramirez, Shengxiang Yang, Armaghan Moemeni, David Elizondo, Simon Colreavy-Donnelly, Luis Fernando Chavarria-Estrada, and Miguel A Molina-Cabello. Correcting data imbalance for semi-supervised covid-19 detection using x-ray chest images. *Applied Soft Computing*, 111:107692, 2021.

[6] Alberto Cano and Bartosz Krawczyk. Rose: Robust online self-adjusting ensemble for continual learning on imbalanced drifting data streams. *Machine Learning*, 111(7): 2561–2599, 2022.

[7] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in neural information processing systems*, 32, 2019.

[8] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[9] Hao Chen, Ran Tao, Yue Fan, Yidong Wang, Jindong Wang, Bernt Schiele, Xing Xie, Bhiksha Raj, and Marios Savvides. Softmatch: Addressing the quantity-quality trade-off in semi-supervised learning. *International Conference on Learning Representations*, 2023.

[10] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011.

[11] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703, 2020.

[12] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9268–9277, 2019.

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

[14] Chris Drummond, Robert C Holte, et al. C4. 5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling. In *Workshop on learning from imbalanced datasets II*, volume 11, pages 1–8, 2003.

[15] Yue Fan, Dengxin Dai, Anna Kukleva, and Bernt Schiele. Cossl: Co-learning of representation and classifier for imbalanced semi-supervised learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14574–14584, 2022.

[16] Lan-Zhe Guo and Yu-Feng Li. Class-imbalanced semi-supervised learning with adaptive thresholding. In *International Conference on Machine Learning*, pages 8082–8094. PMLR, 2022.

[17] Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. Borderline-smote: a new over-sampling method in imbalanced data sets learning. In *Advances in Intelligent Computing: International Conference on Intelligent Computing, ICIC 2005, Hefei, China, August 23-26, 2005, Proceedings, Part I 1*, pages 878–887. Springer, 2005.

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

[19] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. *arXiv preprint arXiv:1910.09217*, 2019.

[20] Dong-Jin Kim, Xiao Sun, Jinsoo Choi, Stephen Lin, and In So Kweon. Detecting human-object interactions with action co-occurrence priors. In *Proceedings of the European conference on computer vision (ECCV)*, pages 718–736, 2020.

[21] Dong-Jin Kim, Xiao Sun, Jinsoo Choi, Stephen Lin, and In So Kweon. Acp++: Action co-occurrence priors for human-object interaction detection. *IEEE Transactions on Image Processing*, 30:9150–9163, 2021.

[22] Jaehyung Kim, Youngbum Hur, Sejun Park, Eunho Yang, Sung Ju Hwang, and Jinwoo Shin. Distribution aligning refinery of pseudo-label for imbalanced semi-supervised learning. *Advances in neural information processing systems*, 33, 2020.

[23] Jiwon Kim, Youngjo Min, Daehwan Kim, Gyuseong Lee, Junyoung Seo, Kwangrok Ryoo, and Seungryong Kim. Conmatch: Semi-supervised learning with confidence-guided consistency regularization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 674–690, 2022.

[24] Jakub Klikowski and Michał Woźniak. Deterministic sampling classifier with weighted bagging for drifted imbalanced data stream classification. *Applied Soft Computing*, 122: 108855, 2022.

[25] Łukasz Korycki and Bartosz Krawczyk. Concept drift detection from multi-class imbalanced data streams. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pages 1068–1079. IEEE, 2021.

[26] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Handbook of Systemic Autoimmune Diseases*, 1(4), 2009.

[27] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016.

[28] Justin Lazarow, Kihyuk Sohn, Chen-Yu Lee, Chun-Liang Li, Zizhao Zhang, and Tomas Pfister. Unifying distribution alignment as a loss for imbalanced semi-supervised learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5644–5653, 2023.

[29] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896, 2013.

[30] Hyuck Lee, Seungjae Shin, and Heeyoung Kim. Abc: Auxiliary balanced classifier for class-imbalanced semi-supervised learning. *Advances in neural information processing systems*, 34, 2021.

[31] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens Van Der Maaten. Exploring the limits of weakly supervised pretraining. In *Proceedings of the European conference on computer vision (ECCV)*, pages 181–196, 2018.

[32] Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. *arXiv preprint arXiv:2007.07314*, 2020.

[33] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *Proceedings of the NeurIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

[34] Youngtaek Oh, Dong-Jin Kim, and In So Kweon. Daso: Distribution-aware semantics-oriented pseudo-label for imbalanced semi-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9786–9796, 2022.

[35] Antti Rasmus, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko. Semi-supervised learning with ladder networks. *Advances in neural information processing systems*, 28, 2015.

[36] Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. *Advances in neural information processing systems*, 29, 2016.

[37] Li Shen, Zhouchen Lin, and Qingming Huang. Relay backpropagation for effective learning of deep convolutional neural networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 467–482, 2016.

[38] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33, 2020.

[39] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017.

[40] Hao Wang, Qilong Wang, Fan Yang, Weiqi Zhang, and Wangmeng Zuo. Data augmentation for object detection via progressive and selective instance-switching. *arXiv preprint arXiv:1906.00358*, 2019.

[41] Yidong Wang, Hao Chen, Yue Fan, Wang Sun, Ran Tao, Wenxin Hou, Renjie Wang, Linyi Yang, Zhi Zhou, Lan-Zhe Guo, et al. Usb: A unified semi-supervised learning benchmark for classification. *Advances in Neural Information Processing Systems*, 35, 2022.

[42] Yidong Wang, Haoxing Chen, Qiang Heng, Wenxin Hou, Marios Savvides, Takahiro Shinozaki, Bhiksha Raj, Zhen Wu, and Jindong Wang. Freematch: Self-adaptive thresholding for semi-supervised learning. *arXiv preprint arXiv:2205.07246*, 2022.

[43] Chen Wei, Kihyuk Sohn, Clayton Mellina, Alan Yuille, and Fan Yang. Crest: A class-rebalancing self-training framework for imbalanced semi-supervised learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10857–10866, 2021.

[44] Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. *Advances in neural information processing systems*, 33, 2020.

[45] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10687–10698, 2020.

[46] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *British Machine Vision Conference 2016*. British Machine Vision Association, 2016.

[47] Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. *Advances in Neural Information Processing Systems*, 34, 2021.

[48] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[49] Songyang Zhang, Zeming Li, Shipeng Yan, Xuming He, and Jian Sun. Distribution alignment: A unified framework for long-tail visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2361–2370, 2021.

[50] Boyan Zhou, Quan Cui, Xiu-Shen Wei, and Zhao-Min Chen. Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9719–9728, 2020.