IOMatch: Simplifying Open-Set Semi-Supervised Learning with Joint Inliers and Outliers Utilization

Zekun Li[1] Lei Qi[2] Yinghuan Shi[1,*] Yang Gao[1]
[1] Nanjing University [2] Southeast University

Abstract

Semi-supervised learning (SSL) aims to leverage massive unlabeled data when labels are expensive to obtain. Unfortunately, in many real-world applications, the collected unlabeled data will inevitably contain unseen-class outliers not belonging to any of the labeled classes. To deal with the challenging open-set SSL task, the mainstream methods tend to first detect outliers and then filter them out. However, we observe a surprising fact that such approach could result in more severe performance degradation when labels are extremely scarce, as the unreliable outlier detector may wrongly exclude a considerable portion of valuable inliers. To tackle with this issue, we introduce a novel open-set SSL framework, IOMatch, which can jointly utilize inliers and outliers, even when it is difficult to distinguish exactly between them. Specifically, we propose to employ a multi-binary classifier in combination with the standard closed-set classifier for producing unified open-set classification targets, which regard all outliers as a single new class. By adopting these targets as open-set pseudo-labels, we optimize an open-set classifier with all unlabeled samples including both inliers and outliers. Extensive experiments have shown that IOMatch significantly outperforms the baseline methods across different benchmark datasets and different settings despite its remarkable simplicity. Our code and models are available at https://github.com/nukezil/IOMatch.

1. Introduction

Semi-supervised learning (SSL) [5] is a classical machine learning paradigm that attempts to improve a model's performance by utilizing unlabeled data in addition to insufficient labeled data. With a tiny fraction of labeled data, advanced deep SSL methods can achieve the performance of fully supervised methods in some cases, such as image classification [28] and semantic segmentation [36].

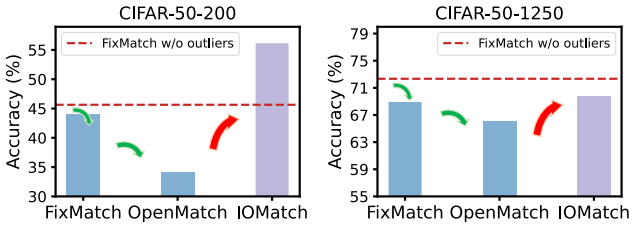

Figure 1. The motivation of our work comes from a surprising fact in open-set semi-supervised learning tasks: An unreliable outlier detector can be more harmful than outliers themselves, because it will wrongly exclude valuable inliers from subsequent training. For this issue, we consider a unified paradigm for utilizing open-set unlabeled data, even when it is difficult to distinguish exactly between inliers and outliers, and thus we propose IOMatch.

Most existing SSL methods rely on the fundamental assumption that labeled and unlabeled data share the same class space. However, it is usually difficult, even impossible, to collect such a unlabeled data set in many real-world applications since we can not manually examine the massive unlabeled data. Therefore, a more challenging scenario arises, where unseen-class outliers not belonging to any of the labeled classes exist in the unlabeled data. Such setting is called Open-Set Semi-Supervised Learning (OSSL) [39].

The negative effects of unseen-class outliers have been observed in a pioneer work [24]. As the research of SSL has grown rapidly in recent years, we extensively evaluate more advanced SSL methods. Some of the key results are shown in Figure 1, in which we plot the performance under standard and open-set SSL as the dash lines and charts, respectively. Taking the classical method, FixMatch [28], as an example, we can observe that adding extra outliers does hurt the classification accuracy compared to the standard SSL setting with no outlier, because it is impossible to obtain correct seen-class pseudo-labels for these outliers. An intuitive approach to handle outliers is to detect and remove them, as OpenMatch [25] does. In particular, it combines FixMatch with an outlier detector. The detector is first pre-trained and then used to retain only inliers for Fix-

*Corresponding author: Yinghuan Shi (syh@nju.edu.cn). Zekun Li, Yinghuan Shi and Yang Gao are with the State Key Laboratory for Novel Software Technology and National Institute of Healthcare Data Science, Nanjing University. Lei Qi is with the School of Computer Science and Engineering, Southeast University. This work is supported by NSFC Program (62222604, 62206052, 62192783), China Postdoctoral Science Foundation Project (2023T160100), Jiangsu Natural Science Foundation Project (BK20210224), and CCF-Lenovo Bule Ocean Research Fund.

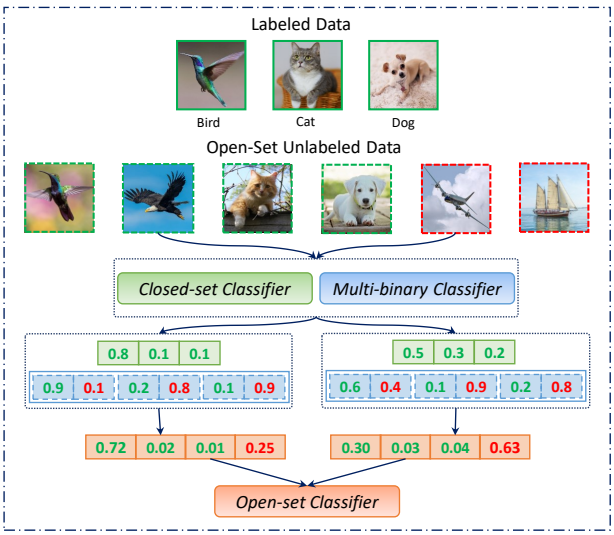

Figure 2. Illustration of joint inliers and outliers utilization. We fuse the predictions of the closed-set classifier and the multi-binary classifier to produce the open-set targets for both inliers and outliers, where outliers are regarded as a single new class (denoted in red). All the open-set unlabeled data will be fully exploited by optimizing an open-set classifier via pseudo-labeling.

Match training. However, we find that such approach actually results in more severe performance degradation especially when labels are extremely scarce. The reason is that the pre-trained detector can be so unreliable that it will wrongly exclude a considerable portion of valuable inliers from subsequent training. In this regard, a surprising fact is that *a bad detector is worse than no detector at all*. Similar to OpenMatch, other existing methods [14, 16, 39] using various outlier detectors also suffer from this issue, as they all follow the detect-and-filter paradigm.

From the above analysis, we can observe that the performance of existing OSSL methods is highly dependent on the unseen-class detection. However, it is difficult indeed to obtain a reliable outlier detector in the early stage of training due to the scarcity of labels. Thus, instead of sending open-set unlabeled samples into different learning branches (*e.g.*, inliers for pseudo-labeling and outliers being thrown away), we are better to deal with them in a unified paradigm. This allows the opportunity to make corrections even if the unseen-class detection is not accurate at the beginning.

In this paper, we consider a novel strategy to jointly utilize inliers and outliers without distinguishing exactly between them, and thus propose a simple yet effective OSSL framework, IOMatch. Along with the standard closed-set classifier, IOMatch adopts a multi-binary classifier [26] that predicts how likely a sample is to be an inlier of each seen class. We fuse the predictions of the two classifiers to produce unified open-set classification targets by regarding all outliers as a new class. These open-set targets are then utilized to train an open-set classifier with both unlabeled inliers and outliers via pseudo-labeling. We illustrate the core idea in Figure 2. Different from the detect-and-filter methods [14, 16, 25, 39], all the network modules of IOMatch are simultaneously optimized, which makes it easy to use.

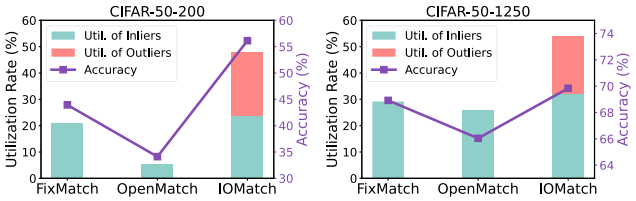

Figure 3. We define the utilization rate of open-set unlabeled data as the ratio of selected correct pseudo-labels to all unlabeled samples. Compared to the previous methods, IOMatch can not only retain more valuable inliers but also utilize additional outliers by adopting open-set targets as pseudo-labels.

We conduct extensive experiments to demonstrate the effectiveness of IOMatch across different benchmark datasets and different settings. The performance gains are significant especially when labels are scarce and class mismatch is severe. For instance, on the CIFAR-100 dataset, IOMatch outperforms the current state-of-the-art by 7.46% and 4.78%, when the proportion of outliers is as high as 80% and 50%, and only 4 labels per seen class are available. Figure 3 explains why IOMatch is able to achieve such improvements: Compared to the existing OSSL methods, IOMatch avoids incorrect exclusion of valuable inliers; Compared to the standard SSL methods, IOMatch can additionally utilize "poisonous" outliers. In a nutshell, with the novel paradigm of joint inliers and outliers utilization, open-set unlabeled data can be more fully exploited by IOMatch.

We summarize our contributions as follows:

- We reveal that existing open-set SSL methods could easily fail due to their unreliable outlier detectors when labels are extremely scarce.
- We propose a novel open-set SSL framework called IOMatch that can jointly utilize both inliers and outliers in a unified paradigm.
- We perform comprehensive experiments across various OSSL settings. In spite of its simplicity, IOMatch significantly outperforms the strong rivals, especially when the tasks are challenging.

2. Related Work

2.1. Semi-Supervised Learning

For mainstream deep SSL approaches, consistency regularization [1] is a crucial technique and has been widely adopted in many works [3, 18, 22, 27, 30]. Briefly speaking, this technique enforces the model to output a consistent prediction on the different perturbed versions of the same sample. Among existing works, FixMatch [28] is one of the most influential SSL methods, which is popular for its sim-

plicity and effectiveness. It improves consistency regularization with strong data augmentation and performs pseudo-labeling based on confidence thresholding. There are many other works that have made important technical contributions to the research of SSL. ReMixMatch [2] introduces distribution alignment and augmentation anchoring. FlexMatch [41] and FreeMatch [34] propose to adjust the class-specific confidence thresholds based on the different learning difficulties. CoMatch [20] and SimMatch [43] incorporate contrastive learning objectives to exploit instance-level similarity. More comprehensive reviews on SSL theories and methods can be found in [31, 33, 37].

Despite the remarkable success on various SSL tasks, all these methods assume that labeled and unlabeled data share the same class space. Such assumption could be difficult to satisfy in real-world applications, which may lead to considerable performance degradation. Therefore, it is necessary to consider the more practical open-set SSL setting.

2.2. Open-Set Semi-Supervised Learning

As the standard closed-set classifier cannot assign correct seen-class pseudo-labels for unseen-class outliers, an intuitive approach is to detect outliers and filter them out before pseudo-labeling. Mainstream OSSL methods adopt such detect-and-exclude strategy to reduce the perturbation from outliers. For example, UASD [7] considers the predictions of the closed-set classifier and use the confidence to identify outliers. Also with the predictions, SAFE-STUDENT [14] defines an energy-discrepancy score to replace the confidence. There are other several methods which introduce additional network modules for unseen-class detection. MTCF [39] adopts a binary classification head which is trained in noisy label optimization paradigm. T2T [16] proposes a cross-modal matching module to predict whether a sample is matched to an assigned one-hot seen-class label. With the similar idea, OpenMatch [25] employs a group of one-vs-all classifiers as the outlier detector.

Although the above OSSL methods are effective when labels are relatively sufficient (*e.g.*, 100 labels per seen class or more), it is hard to achieve satisfactory unseen-class detection performance when the number of labeled samples is extremely limited. In such a challenging scenario, even after a pre-training stage, the outlier detector still does not perform well due to the scarcity of labels. As a consequence, it will tend to wrongly exclude a large portion of unlabeled inliers. Without exposure to these misidentified samples, such errors are quite difficult to rectify, which will lead to more severe performance degradation than that caused by outliers themselves. A few recent methods propose to perform extra pretext tasks, such as rotation recognition [16] and label distribution calibration [14], with the detected outliers. These techniques may mitigate the adverse affects of the unreliable outlier detector, but cannot really address the issue.

Another related learning problem is out-of-distribution (OOD) detection [15], which aims to separate OOD samples from in-distribution (ID) samples. OOD detection has different problem formulation and learning objectives from OSSL, so it is out of the scope of this work. For further discussions about the connections and differences between the two problems, please refer to the supplementary material.

3. IOMatch

3.1. Preliminaries and Overview

We define the open-set semi-supervised learning task as following. For a K-class classification problem, let $\mathcal{X} = \{(\boldsymbol{x}_i, y_i) : i \in (1, \dots, B)\}$ be a batch of B labeled samples, where $\boldsymbol{x}_i$ is a training sample and $y_i \in \{1, \dots, K\}$ is the corresponding label. Let $\mathcal{U} = \{\boldsymbol{u}_i : i \in (1, \dots, \mu B)\}$ be a batch of μB unlabeled samples, where μ is a hyper-parameter that determines the relative sizes of $\mathcal{X}$ and $\mathcal{U}$. In the OSSL task, there exists a subset $\mathcal{U}^{out} \subset \mathcal{U}$, where $\mathcal{U}^{out} = \{\boldsymbol{u}^{out}\}$ and $\boldsymbol{u}^{out}$ does not belong to any of the K seen classes. Then, $\mathcal{U}^{out}$ are called unseen-class *outliers* and the rest of unlabeled samples $\mathcal{U}^{in} = \mathcal{U}/\mathcal{U}^{out}$ are called seen-class *inliers*.

Given a labeled batch $\mathcal{X}$, we apply a random weak transformation function $\mathcal{T}_w(\cdot)$ to obtain the weakly augmented samples. A base encoder network $f(\cdot)$ is employed to extract the features from these samples, *i.e.*, $\boldsymbol{h}_i = f(\mathcal{T}_w(\boldsymbol{x}_i)) \in \mathbb{R}^D$. A closed-set classifier $\phi(\cdot)$ maps the feature $\boldsymbol{h}_i$ into the predicted seen-class probability distribution, *i.e.*, $\boldsymbol{p}_i = \phi(\boldsymbol{h}_i)$. The labeled batch are used to optimize the networks with the standard cross-entropy loss $\mathrm{H}(\cdot, \cdot)$:

$$\mathcal{L}_s(\mathcal{X}) = \frac{1}{B} \sum_{i=1}^{B} \mathrm{H}(y_i, \boldsymbol{p}_i). \tag{1}$$

Additionally, we adopt a projection head $g(\cdot)$ to obtain the low-dimensional embedding $\boldsymbol{z}_i = g(\boldsymbol{h}_i) \in \mathbb{R}^d$ and then a multi-binary classifier $\chi(\cdot)$ to produce the class-wise likelihood of inliers or outliers $\boldsymbol{o}_i = \chi(\boldsymbol{z}_i) \in \mathbb{R}^{2K}$.

For an unlabeled batch $\mathcal{U}$, we apply both the weak and strong augmentation with $\mathcal{T}_w(\cdot)$ and $\mathcal{T}_s(\cdot)$. The same operations as above are performed to obtain $\boldsymbol{h}_i^w$, $\boldsymbol{z}_i^w$, $\boldsymbol{p}_i^w$, and $\boldsymbol{o}_i^w$ for the weakly augmented samples $\mathcal{T}_w(\boldsymbol{u}_i)$; $\boldsymbol{h}_i^s$, $\boldsymbol{z}_i^s$, $\boldsymbol{p}_i^s$ and $\boldsymbol{o}_i^s$ for the strongly augmented samples $\mathcal{T}_s(\boldsymbol{u}_i)$. Moreover, an open-set classifier $\psi(\cdot)$ is introduced for the unlabeled samples to predict the open-set probability distribution, where all outliers are regarded as a single new class.

The overall framework of IOMatch is illustrated in Figure 4. We propose a novel approach to produce unified open-set targets by fusing predictions of the closed-set classifier and the multi-binary classifier. These targets are then used to optimize the closed-set and open-set classifiers to achieve joint inliers and outliers utilization. As an one-stage

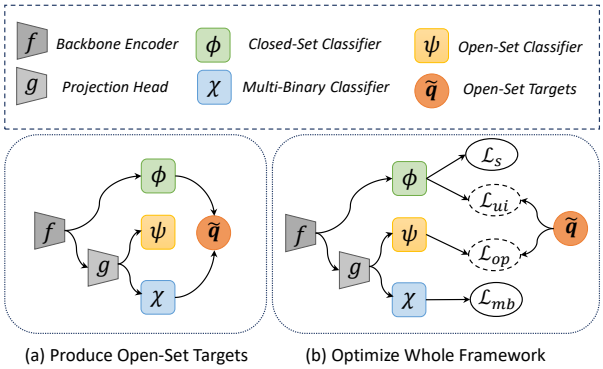

Figure 4. Overview of IOMatch. In each iteration, we first employ the closed-set classifier and the multi-binary classifier to produce the open-set targets, which are then used for selecting high-quality inliers and utilizing outliers. All the network modules in IOMatch are simultaneously optimized with four learning objectives.

method, IOMatch shows remarkable simplicity and is easy to deploy across various OSSL settings.

3.2. Unified Open-Set Targets Production

As the standard closed-set classifier can only assign each sample to one of the seen classes, we employ an additional multi-binary classifier which has been proved capable in related unseen-class detection problems [25, 26, 44]. The multi-binary classifier can be viewed as a combination of K sub-classifiers, *i.e.*, $\chi = \{\chi_k : k \in (1, \dots, K)\}$. Technically, χ_k is the binary classifier for the k-th seen class with the output $\boldsymbol{o}_{i,k} = \chi_k(\boldsymbol{z}_i) \in \mathbb{R}^2$, where $\boldsymbol{o}_{i,k} = (o_{i,k}, \bar{o}_{i,k})$ and $o_{i,k} + \bar{o}_{i,k} = 1$. $\boldsymbol{o}_{i,k}$ is a probability distribution to indicate how likely the sample $\boldsymbol{x}_i$ is to be an inlier or an outlier with respect to the k-th seen class. The hard-negative sampling strategy [26] is adopted to optimize the multi-binary classifier with the labeled samples:

$$\mathcal{L}_{mb}(\mathcal{X}) = \frac{1}{B}\sum_{i=1}^{B}\left(-\log(o_{i,y_i}) - \min_{k \neq y_i}\log(\bar{o}_{i,k})\right). \quad (2)$$

Combining the multi-binary classifier with the closed-set classifier makes it possible to identify outliers. In the previous work, OpenMatch [25], an unlabeled sample $\boldsymbol{u}_i$ is first assigned to one of the K seen classes according to the closed-set prediction, *i.e.*, $\hat{y}_i = \arg\max_k(p^w_{i,k})$. Then, the binary probability $o^w_{i,\hat{y}_i}$ is considered to decide whether the sample is an inlier of the $\hat{y}_i$-th seen class or an unseen-class outlier, with the natural threshold of 0.5. When the labels are relatively sufficient (*e.g.*, 100 labels per class or more), such approach is effective, since the closed-set and the multi-binary classifiers can perform well after a pre-training stage with the labeled samples. However, when the number of labeled samples is limited, the one-hot pseudo-labels for seen classes will be hardly reliable.

Aware of this issue, we propose a novel approach to fully fuse the predictions of the two classifiers. Specifically, for each unlabeled sample $\boldsymbol{u}_i$, the seen-class probability distribution is predicted by the closed-set classifier, *i.e.*, $\widetilde{\boldsymbol{p}}_i = \mathrm{DA}(\phi(\boldsymbol{h}^w_i))$, where $\mathrm{DA}(\cdot)$ stands for the distribution alignment strategy proposed by [2] to balance the distribution of the model's predictions and thus prevent them from collapsing to certain classes. As the two classifiers are parameter-independent, $\widetilde{p}_{i,k}$ and $o^w_{i,k}$ are two distinct and complementary predictions on how likely $\boldsymbol{u}_i$ belongs to the k-th seen class. Therefore, for $1 \leq k \leq K$, we use

$$\widetilde{q}_{i,k} = \widetilde{p}_{i,k} \cdot o^w_{i,k} \quad (3)$$

to estimate the probability that $\boldsymbol{u}_i$ belongs to the k-th seen class, when taking the possibility of outliers into consideration. Therefore, the probability that $\boldsymbol{u}_i$ is an outlier not belonging to any of the K seen classes is estimated by

$$\mathcal{S}_i = 1 - \sum_{j=1}^{K}\widetilde{q}_{i,j} = \sum_{j=1}^{K}\widetilde{p}_{i,j} \cdot \bar{o}^w_{i,j}. \quad (4)$$

Putting them all together produces a $(K{+}1)$-way class probability distribution $\widetilde{\boldsymbol{q}}_i \in \mathbb{R}^{K+1}$ by regarding all unseen classes as the virtual $(K{+}1)$-th class:

$$\widetilde{q}_{i,k} = \begin{cases} \widetilde{p}_{i,k} \cdot o^w_{i,k} & \text{if} \quad 1 \leq k \leq K; \\ \sum_{j=1}^{K}\widetilde{p}_{i,j} \cdot \bar{o}^w_{i,j} & \text{if} \quad k = K+1. \end{cases} \quad (5)$$

In this way, we obtain a kind of unified open-set targets for all unlabeled samples, eliminating the need to precisely differentiate between inliers and outliers. This lays the foundation for the joint utilization of both inliers and outliers.

3.3. Joint Inliers and Outliers Utilization

For all the open-set unlabeled samples, we adopt the open-set targets as supervision to train the open-set classifier $\psi(\cdot)$ with its predictions $\boldsymbol{q}^s_i = \psi(\boldsymbol{z}^i_s) \in \mathbb{R}^{K+1}$ on the strongly augmented samples:

$$\mathcal{L}_{op}(\mathcal{U}) = \frac{1}{\mu B}\sum_{i=1}^{\mu B}\mathbb{1}(\max_k(\widetilde{q}_{i,k}) > \tau_q) \cdot \mathrm{H}(\widetilde{\boldsymbol{q}}_i, \boldsymbol{q}^s_i), \quad (6)$$

where $\mathbb{1}(\cdot)$ is the indicator function and τ_q is the confidence threshold. In practice, we usually choose a low value for τ_q so that most of the unlabeled samples can be utilized. Different from the traditional consistency regularization technique, we use $\widetilde{\boldsymbol{q}}_i$ instead of the predictions $\boldsymbol{q}^w_i$ on the weakly augmented samples as supervision. In this way, the generation and utilization of pseudo-labels can be disentangled to alleviate the accumulation of confirmation bias.

As the open-set targets are produced by the closed-set and the multi-binary classifiers, we need to further optimize the two classifiers to obtain better open-set targets. In fact, optimizing the open-set classifier via pseudo-labeling leads

Algorithm 1 Optimization of IOMatch in Every Training Iteration

Input: $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{B}$ and $\{\boldsymbol{u}_i\}_{i=1}^{\mu B}$: Labeled and unlabeled samples. $\mathcal{T}_w(\cdot)$ and $\mathcal{T}_s(\cdot)$: Weak and strong augmentation. $f(\cdot)$: Base encoder. $g(\cdot)$: Projection head. $\phi(\cdot)$: Closed-set classifier. $\chi(\cdot)$: Multi-binary classifier. $\psi(\cdot)$: Open-set classifier. τ_p and τ_q: Confidence thresholds. λ_{mb}, λ_{ui}, λ_{op}: Weights of losses.

$\boldsymbol{h}_i = f(\mathcal{T}_w(\boldsymbol{x}_i)), \boldsymbol{h}_i^w = f(\mathcal{T}_w(\boldsymbol{u}_i)), \boldsymbol{h}_i^s = f(\mathcal{T}_s(\boldsymbol{x}_i))$ ▷ *Obtain the features of the labeled and unlabeled samples.*

$\boldsymbol{z}_i = g(\boldsymbol{h}_i), \boldsymbol{z}_i^w = g(\boldsymbol{h}_i^w), \boldsymbol{z}_i^s = g(\boldsymbol{h}_i^s)$ ▷ *Map the features into the projection space.*

$\boldsymbol{p} = \phi(\boldsymbol{h}_i), \widetilde{\boldsymbol{p}} = \mathrm{DA}(\phi(\boldsymbol{h}_i^w)), \boldsymbol{p}^s = \phi(\boldsymbol{h}_i^s), \boldsymbol{o} = \chi(\boldsymbol{z}_i), \boldsymbol{o}^w = \chi(\boldsymbol{z}_i^w)$ ▷ *Make closed-set and multi-binary predictions.*

$\mathcal{L}_s(\mathcal{X}) = \frac{1}{B}\sum_{i=1}^{B} \mathrm{H}(y_i, \boldsymbol{p}_i)$ ▷ *Calculate the supervised loss.*

$\mathcal{L}_{mb}(\mathcal{X}) = \frac{1}{B}\sum_{i=1}^{B} (-\log(o_{i,y_i}) - \min_{k \neq y_i} \log(\bar{o}_{i,k}))$ ▷ *Calculate the multi-binary loss.*

$\widetilde{q}_{i,k} = \widetilde{p}_{i,k} \cdot o_{i,k}^w (1 \leq k \leq K); \widetilde{q}_{i,K+1} = \mathcal{S}_i = \sum_{j=1}^{K} \widetilde{p}_{i,j} \cdot \bar{o}_{i,j}^w$ ▷ *Produce open-set targets.*

$\mathcal{L}_{op}(\mathcal{U}) = \frac{1}{\mu B}\sum_{i=1}^{\mu B} \mathbb{1}(\max_k(\widetilde{q}_{i,k}) > \tau_q) \cdot \mathrm{H}(\widetilde{\boldsymbol{q}}_i, \boldsymbol{q}_i^s)$ ▷ *Calculate the open-set loss.*

$\mathcal{L}_{ui}(\mathcal{U}) = \frac{1}{\mu B}\sum_{i=1}^{\mu B} \mathbb{1}(\max_k(\widetilde{p}_{i,k}) > \tau_p) \cdot \mathbb{1}(\mathcal{S}_i < 0.5) \cdot \mathrm{H}(\widetilde{\boldsymbol{p}}_i, \boldsymbol{p}_i^s)$ ▷ *Calculate the unlabeled inliers loss.*

Output: The overall loss $\mathcal{L}_{overall} = \mathcal{L}_s + \lambda_{mb}\mathcal{L}_{mb} + \lambda_{ui}\mathcal{L}_{ui} + \lambda_{op}\mathcal{L}_{op}$ to update the network parameters.

to more discriminative features in the projection space and improves the performance of the multi-binary classifier at the same time. Then, for the closed-set classifier, we propose a double filtering strategy to select high-quality seen-class pseudo-labels of inliers:

$$\mathcal{L}_{ui}(\mathcal{U}) = \frac{1}{\mu B}\sum_{i=1}^{\mu B} \mathcal{F}(\boldsymbol{u}_i) \cdot \mathrm{H}(\widetilde{\boldsymbol{p}}_i, \boldsymbol{p}_i^s). \tag{7}$$

$\mathcal{F}(\cdot)$ is the filtering function, which is defined as $\mathcal{F}(\boldsymbol{u}_i) = \mathbb{1}(\max_k(\widetilde{p}_{i,k}) > \tau_p) \cdot \mathbb{1}(\mathcal{S}_i < 0.5)$, where τ_p is another confidence threshold. We use $\mathcal{S}_i$ to exclude the likely outliers and use τ_p to ignore incorrect pseudo-labels of inliers. As these temporarily excluded samples have been utilized by the open-set classifier, the true inliers will be gradually involved in the training, which prevents IOMatch from falling into the same issue as the previous OSSL methods.

The overall optimization objective of IOMatch is consistent through the training, which is defined as

$$\mathcal{L}_{overall} = \mathcal{L}_s + \lambda_{mb}\mathcal{L}_{mb} + \lambda_{ui}\mathcal{L}_{ui} + \lambda_{op}\mathcal{L}_{op}, \tag{8}$$

where λ_{mb}, λ_{ui}, and λ_{op} are the weights of each learning objective, respectively. As these learning objectives are all cross-entropy losses[1] with the same order of magnitude, we can simply set $\lambda_{mb} = \lambda_{ui} = \lambda_{op} = 1$. In Algorithm 1, we present the detailed optimization procedure in every training iteration. Different from the existing OSSL methods based on the detect-and-filter strategy, IOMatch is an one-stage framework which omits a sensitive hyperparameter, *i.e.*, the number of epochs for pre-training the outlier detector. Using only cross-entropy losses, IOMatch is much easier to implement than the methods equipped with complex contrastive learning objectives. From these aspects, IOMatch shows remarkable simplicity.

[1]The multi-binary loss $\mathcal{L}_{mb}$ can be viewed as the combination of two binary cross-entropy losses.

3.4. Inference

The well trained encoder, projector, and classifiers of IOMatch will be used for inference. For the closed-set classification task, the closed-set classifier is employed to obtain $\boldsymbol{p}_t = \phi(f(\boldsymbol{x}_t))$ and assign the test sample $\boldsymbol{x}_t$ to the $\hat{y}_t$-th seen class, where $\hat{y}_t = \arg\max_k(p_{t,k}) \in \{1, \ldots, K\}$. For the open-set classification task that regards all unseen-class outliers as a single new class, we consider the open-set probability distribution produced by the open-set classifier, *i.e.*, $\boldsymbol{q}_t = \psi(g(f(\boldsymbol{x}_t)))$. The open-set prediction is given by $\hat{y}_t = \arg\max_k(q_{t,k}) \in \{1, \ldots, K+1\}$. In fact, $\boldsymbol{q}_t$ can also be used for the closed-set task by ignoring its last item, while we still use $\boldsymbol{p}_t$ to be consistent with other methods.

3.5. Connections to Existing Methods

Although both employ the multi-binary classifier for unseen-class detection, IOMatch distinguishes itself from the previous OpenMatch [25] in the following aspects: (1) In IOMatch, we optimize the closed-set classifier and the multi-binary classifier in different feature spaces to mitigate mutual interference. (2) All the network modules in IOMatch are simultaneously optimized, without an extra pre-training stage for the multi-binary classifier. (3) We adopt a novel unified paradigm for jointly utilizing inliers and outliers, which is totally different from the conventional detect-and-exclude strategy.

Compared to the standard SSL methods, like FixMatch [28], IOMatch can properly utilize the outliers to mitigate their negative affects on pseudo-labeling and even be able to achieve additional performance gains from them. Moreover, IOMatch is a general SSL framework that also performs well in the standard SSL setting. For standard SSL tasks, IOMatch can utilize the low-confidence inliers (as a kind of "outliers"), which will be ignored by FixMatch. It will yield significant performance improvements, especially when labels are scarce.

4. Experiments

4.1. Experimental Setup

We construct the open-set SSL benchmarks using public datasets, CIFAR-10/100 [17] and ImageNet [8]. We adopt a similar manner to [25] for splitting seen and unseen classes. We conduct experiments with varying class splits and varying labeled set sizes in order to cover various open-set SSL settings. Both the closed-set and open-set performance of methods are evaluated.

Baselines. For standard SSL methods, we focus on the latest state-of-the-arts, including MixMatch [3], ReMixMatch [2], FixMatch [28], CoMatch [20], FlexMatch [41], SimMatch [43] and FreeMatch [34]. We exclude earlier deep SSL methods [18, 22, 27, 30] because these methods perform worse than a model trained only with labeled data on OSSL tasks [7, 10, 24]. For open-set SSL methods, we consider the published works, including UASD [7], DS^{3}L [10], MTCF [39], T2T [16], OpenMatch [25] and SAFE-STUDENT [14].

Closed-Set Evaluation. In this work, we mainly consider the closed-set classification accuracy on the test data from seen classes only, which measures the ability of models to utilize open-set unlabeled data for helping seen-class classification. We follow USB [33] to report the best results of all epochs to avoid unfair comparisons caused by different convergence speeds. Each task is conducted with three different random seeds and the results are expressed as mean values with standard deviation.

Open-Set Evaluation. For open-set SSL methods, we additionally evaluate their classification performance on open-set test data including both seen and unseen classes. In testing, we regard all unseen classes as a single new class, *i.e.*, the (*K*+1)-th class. Considering the open-set test data can be extremely class-imbalanced, since the number of outliers is much larger than that of inliers from each seen class, we adopt Balanced Accuracy (BA) [4] as the open-set classification accuracy, which is defined as

$$BA = \frac{1}{K+1}\sum_{k=1}^{K+1} Recall_k, \tag{9}$$

where $Recall_k$ is the recall score of the k-th class. For each method, the evaluation uses its best checkpoint model in terms of the closed-set performance.

Fairness of Comparisons. We have taken utmost care to ensure fair comparisons in our evaluation. Firstly, we create a unified test bed using the USB codebase [33]. For the standard SSL methods, we follow the re-implementations provided by USB as they yield better results than the published ones under the standard SSL setting. As for the previous open-set SSL methods, we incorporate their released code into our test bed. Because our experimental setup differs from those of the previous works (as ours involves fewer labels, making it more challenging), we first evaluate these re-implemented methods in their original setups and observe the results that are close to or higher than those reported in the published papers, which verifies the correctness of our re-implementations. Moreover, for the hyperparameters that are common to different methods, we make sure that they have consistent values. As for method-specific hyperparameters, we refer to the optimal values provided in their original papers. Experiments of each setting are performed using the same backbone networks, the same data splits, and the same random seeds.

4.2. Main Results

4.2.1 CIFAR-10 and CIFAR-100

For CIFAR-10, we use the animal classes as seen classes and the others as unseen classes, resulting in a seen/unseen class split of $6/4$. CIFAR-100 consists of 100 classes from 20 super-classes. We split the super-classes into seen and unseen so that inliers and outliers will belong to different super-classes. We use the first 4, 10, or 16 super-classes as seen classes, resulting in three splits of $20/80$, $50/50$, and $80/20$, respectively. For both CIFAR-10 and CIFAR-100, we randomly select 4 or 25 samples from the training set of each seen class as the labeled data and use the rest of the training set as the unlabeled data. We use WRN-28-2 [40] as the backbone encoder. We use an identical set of hyperparameters, which is $\{\lambda_{mb} = \lambda_{ui} = \lambda_{op} = 1, \tau_p = 0.95, \tau_q = 0.5, \mu = 7, B = 64, N_e = 256, N_i = 1024\}$, across all tasks. N_e indicates the total number of training epochs and N_i is the number of iterations per epoch.

For the closed-set classification tasks, we compare the proposed IOMatch with thirteen latest standard and open-set SSL methods. For convenience, we denote the tasks on CIFAR-10 with 6 seen classes, 4 and 25 labeled samples per class as `CIFAR-6-24` and `CIFAR-6-150`, respectively. The denotations are similar for other tasks. We report the performance of the closed-set classifier to be consistent with other baselines. The results are presented in the Table 1. With respect to the closed-set classification accuracy, IOMatch achieves best performance in most tasks. When the class mismatch is severe and the labels are scarce, the improvements are quite remarkable. In particular, IOMatch outperforms the strongest rivals by 3.60%, 7.46% and 4.78% on `CIFAR-6-24`, `CIFAR-20-80`, and `CIFAR-50-200`, respectively.

When more labeled samples are available and fewer unlabeled outliers exist, the performance gains of IOMatch would be smaller. The reason is that, in these less challenging tasks, the current state-of-the-art SSL method like SimMatch [43], can be relatively robust to the outliers with the help of its intricate contrastive learning objective. However, IOMatch can achieve better or comparable perfor-

Table 1. Closed-set classification accuracy (%) on the *seen-class* test data of CIFAR-10/100 with varying seen/unseen class splits and labeled set sizes. We report the mean with standard deviation over 3 runs of different random seeds.

	Dataset		CIFAR-10		CIFAR-100					
	Class split (Seen / Unseen)		6 / 4		20 / 80		50 / 50		80 / 20	
	Number of labels per class		4	25	4	25	4	25	4	25
Standard SSL	MixMatch [3]	NeurIPS'19	43.08 ± 1.79	63.13 ± 0.64	28.13 ± 5.06	51.28 ± 1.45	26.97 ± 0.46	56.93 ± 0.84	28.35 ± 0.83	53.77 ± 0.97
	ReMixMatch [2]	ICLR'20	72.82 ± 1.81	87.08 ± 1.12	36.02 ± 3.56	61.83 ± 0.81	37.57 ± 1.54	65.80 ± 1.33	40.64 ± 2.97	62.90 ± 1.07
	FixMatch [28]	NeurIPS'20	81.58 ± 6.63	92.94 ± 0.80	46.27 ± 0.64	66.45 ± 0.74	48.93 ± 5.05	68.77 ± 0.89	43.06 ± 1.21	64.44 ± 0.51
	CoMatch [20]	ICCV'21	86.08 ± 1.08	92.57 ± 0.47	43.53 ± 3.01	66.82 ± 1.37	43.17 ± 0.55	67.85 ± 1.17	37.89 ± 1.22	62.04 ± 0.08
	FlexMatch [41]	NeurIPS'21	73.34 ± 4.42	86.44 ± 3.72	37.93 ± 4.49	62.68 ± 2.02	44.10 ± 1.88	68.98 ± 0.94	43.44 ± 2.40	64.34 ± 0.64
	SimMatch [43]	CVPR'22	79.84 ± 4.76	90.07 ± 2.44	36.93 ± 5.72	67.23 ± 1.13	51.53 ± 2.02	69.71 ± 1.44	50.32 ± 2.57	**65.68 ± 1.43**
	FreeMatch [34]	ICLR'23	79.26 ± 4.11	92.27 ± 0.15	45.18 ± 8.36	64.62 ± 0.79	50.26 ± 1.92	68.57 ± 0.27	47.34 ± 0.57	64.41 ± 0.55
Open-Set SSL	UASD [7]	AAAI'20	35.25 ± 1.07	56.42 ± 1.34	29.78 ± 4.28	53.78 ± 0.67	29.08 ± 1.44	54.24 ± 1.10	26.41 ± 2.16	50.33 ± 0.62
	DS^{3}L [10]	ICML'20	39.09 ± 1.24	51.83 ± 1.06	19.70 ± 1.98	41.78 ± 1.45	21.62 ± 0.54	47.41 ± 0.61	20.10 ± 0.48	40.51 ± 1.02
	MTCF [39]	ECCV'20	49.15 ± 6.12	74.42 ± 2.95	32.58 ± 3.36	55.93 ± 1.66	35.35 ± 2.39	57.72 ± 0.20	25.40 ± 1.20	54.59 ± 0.49
	T2T [16]	ICCV'21	73.89 ± 1.55	85.69 ± 1.90	44.23 ± 2.27	65.60 ± 0.71	39.31 ± 1.16	68.59 ± 0.92	38.16 ± 0.59	63.86 ± 0.32
	OpenMatch [25]	NeurIPS'21	43.63 ± 3.26	66.27 ± 1.86	37.45 ± 2.67	62.70 ± 1.76	33.74 ± 0.38	66.53 ± 0.54	28.54 ± 1.15	61.23 ± 0.81
	SAFE-STUDENT [14]	CVPR'22	59.28 ± 1.18	77.87 ± 0.14	34.53 ± 0.67	58.07 ± 1.40	35.84 ± 0.86	62.75 ± 0.38	34.17 ± 0.69	57.99 ± 0.34
	IOMatch	**Ours**	**89.68 ± 2.04**	**93.87 ± 0.16**	**53.73 ± 2.12**	**67.28 ± 1.10**	**56.31 ± 2.29**	**69.77 ± 0.58**	**50.83 ± 0.99**	64.75 ± 0.52

Table 2. Open-set classification balanced accuracy (%) on the *open-set* test data of CIFAR-10/100, which consist of samples from all the seen and unseen classes. We report the mean with standard deviation over 3 runs of different random seeds.

	Dataset		CIFAR-10		CIFAR-100					
	Class split (Seen / Unseen)		6 / 4		20 / 80		50 / 50		80 / 20	
	Number of labels per class		4	25	4	25	4	25	4	25
Open-Set SSL	UASD [7]	AAAI'20	17.10 ± 0.32	36.01 ± 0.22	10.50 ± 0.83	26.96 ± 0.53	6.92 ± 0.55	32.23 ± 0.54	5.77 ± 0.21	27.61 ± 1.15
	DS3L [10]	ICML'20	30.89 ± 0.33	40.45 ± 0.77	12.56 ± 1.21	34.35 ± 0.41	12.14 ± 0.39	35.17 ± 0.48	11.10 ± 1.27	29.09 ± 0.31
	MTCF [39]	ECCV'20	33.35 ± 7.21	46.13 ± 0.54	8.12 ± 2.10	26.60 ± 3.66	4.13 ± 0.37	38.36 ± 0.29	1.46 ± 0.17	30.75 ± 0.52
	T2T [16]	ICCV'21	50.57 ± 0.38	61.10 ± 0.39	17.17 ± 1.37	37.18 ± 0.60	12.74 ± 2.66	44.24 ± 0.42	34.23 ± 0.57	51.41 ± 0.96
	OpenMatch [25]	NeurIPS'21	14.37 ± 0.05	20.35 ± 3.50	8.77 ± 2.84	39.89 ± 1.16	7.00 ± 0.02	49.75 ± 1.08	6.30 ± 0.87	44.83 ± 0.62
	SAFE-STUDENT [14]	CVPR'22	45.27 ± 0.36	52.78 ± 0.64	15.94 ± 1.07	28.83 ± 0.46	23.98 ± 0.88	46.71 ± 1.74	29.43 ± 0.66	50.48 ± 0.61
	IOMatch	**Ours**	**75.08 ± 1.92**	**78.96 ± 0.08**	**45.94 ± 1.70**	**58.52 ± 0.48**	**46.36 ± 1.93**	**60.78 ± 0.71**	**39.96 ± 0.95**	**54.39 ± 0.38**

mance with less computation overhead. Furthermore, we intend to demonstrate that IOMatch is also compatible with these potent techniques. When coupled with the auxiliary self-supervised learning objectives [2], the performance of IOMatch can be further enhanced, surpassing the baselines entirely, as shown in Table 5.

Because the standard SSL methods do not have the capability to detect unseen-class outliers, we perform the open-set evaluation only with the open-set SSL methods. From the results presented in Table 2, it is clear that IOMatch outperforms all the baselines by large margins. As we have discussed previously, the outlier detectors in these methods suffer severely from the label scarcity and tend to wrongly detect the vast majority of inliers as outliers, which results in the bad performance, especially when only 4 labels per class are available.

In Table 2, the outliers used for testing are similar to those processed during training, as we use the original test sets of CIFAR10/100, which include all the 10/100 classes. In order to evaluate the classification performance on the wild open-set test data, we also conduct the experiments with the test set containing foreign outliers from different datasets than CIFAR10/100. We observe that IOMatch can still achieve impressive open-set performance for this case. We present the detailed setting and corresponding results in the supplementary material.

4.2.2 ImageNet

Following [25], we choose ImageNet-30 [18], which is a subset of ImageNet [8] containing 30 classes. The first 20 classes are used as seen classes and the rest as unseen classes. For each seen class, we randomly select 1% or 5% of images with labels (13 or 65 samples per class, respectively) and the rest of images are unlabeled. Considering the high computation overhead, we adopt ResNet-18 [13] as the backbone encoder and set $\{B = 32, \mu = 1, N_e = 100\}$ to finish the experiments in reasonable time. Other hyperparameters are kept consistent with the previous experiments

Table 3. Close-set and open-set accuracy (%) on ImageNet-30 with the class split of 20/10. We report the mean with standard deviation over 3 runs of different random seeds.

Evaluation	Closed-Set		Open-Set	
Labeled ratio	1%	5%	1%	5%
FixMatch	52.52 ± 3.82	78.55 ± 1.46	–	–
CoMatch	62.92 ± 0.90	79.17 ± 0.42	–	–
SimMatch	64.15 ± 0.94	80.23 ± 0.53	–	–
T2T	63.70 ± 0.83	78.87 ± 0.49	48.81 ± 0.88	58.51 ± 0.41
OpenMatch	56.35 ± 3.35	73.90 ± 1.05	21.80 ± 1.90	57.25 ± 0.76
SAFE-STUDENT	58.38 ± 2.34	75.85 ± 0.99	44.08 ± 2.09	55.25 ± 1.46
IOMatch	**69.18 ± 1.68**	**81.43 ± 0.78**	**57.71 ± 2.69**	**73.94 ± 0.99**

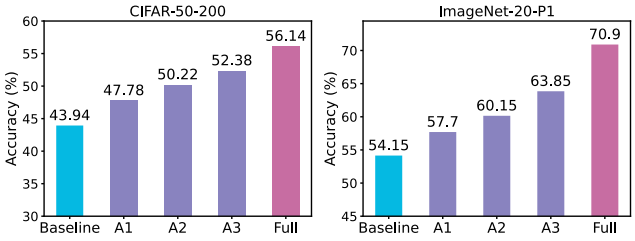

Figure 5. Ablation results on different combinations of learning objectives. "A1", "A2", and "A3" stand for the frameworks optimized with $\{\mathcal{L}_s, \mathcal{L}_{ui}\}$, $\{\mathcal{L}_s, \mathcal{L}_{mb}, \mathcal{L}_{ui}\}$, $\{\mathcal{L}_s, \mathcal{L}_{mb}, \mathcal{L}_{op}\}$, respectively. We compare the performance with FixMatch ("Baseline") and the full version of IOMatch ("Full").

on CIFAR10/100. Similarly, we denote the two tasks as `ImageNet-20-P1` and `ImageNet-20-P5`.

We select the methods achieving better performance for the complete evaluation with three different seeds. The results including closed-set and open-set classification accuracy on ImageNet-30 are presented in Table 3. On this more complex and more challenging benchmark dataset, IOMatch also demonstrates its superiority in both closed-set and open-set performance. The performance can be further improved, if we use deeper backbone networks, larger batch size, and more training epochs. Nevertheless, the current results have demonstrated the effectiveness of IOMatch when computational resources are relatively limited.

4.3. Ablation Analysis and Discussions

To better understand why IOMatch can obtain state-of-the-art results on OSSL tasks, we perform extensive ablation studies on the learning objectives and corresponding hyperparameters. Besides, we present some important additional results and discuss the current design and further improvements of IOMatch in depth.

Learning Objectives. With the standard closed-set classifier, IOMatch additionally introduces a multi-binary classifier and an open-set classifier. To examine the effects of these modules, we ablate their corresponding objectives,

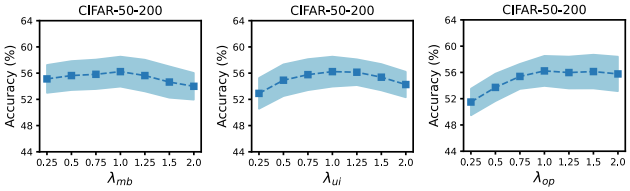

Figure 6. Performance with different values of each weight (*i.e.*, λ_{mb}, λ_{ui} and λ_{op}). It is shown that setting all the weights to 1 is a simple yet appropriate choice.

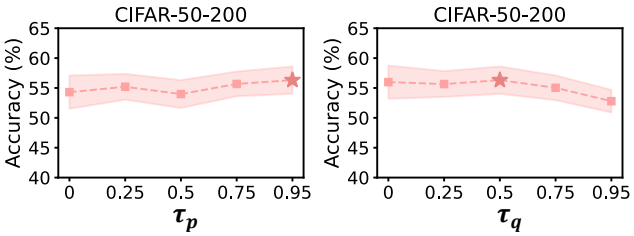

Figure 7. We vary the confidence thresholds, τ_p and τ_q, respectively. The set $\{\tau_p = 0.95, \tau_q = 0.5\}$ gives the best performance.

$\mathcal{L}_{ui}$, $\mathcal{L}_{mb}$, and $\mathcal{L}_{op}$, respectively. The results are presented in Figure 5. Comparing "A2" with "A1", using the multi-binary classifier alone can bring some improvement as it can help to select more accurate closed-set pseudo-labels. From the results of "A3", we find the unsupervised training of the closed-set classifier is still important for producing better open-set targets. Most importantly, the comparisons demonstrate that the joint inliers and outliers utilization achieved by $\mathcal{L}_{op}$ is the key to the success.

Weights of Losses. We separately set the value of each weight (*i.e.*, λ_{mb}, λ_{ui} and λ_{op}) to traverse $\{0.25, 0.5, 0.75, 1, 1.25, 1.5, 2\}$, and control the other two weights to be 1. And please note that we have already discussed the extreme cases where the weights are set to 0 in the above ablation study. As shown in the Figure 6, the performance remains relatively stable when the weights are close to 1; whereas the weights that are too small or too large may lead to performance degradation. Since the learning objectives are all cross-entropy losses with the same order of magnitude, it is reasonable to balance them with similar weights, which is well supported by the experimental observations.

Confidence Thresholds. We adopt the different confidence thresholds (*i.e.*, τ_p and τ_q) for the closed-set classifier and the open-set classifier. We present the results of varying τ_p and τ_q values separately in Figure 7. It is shown that the performance is relatively robust to the value of τ_p. Even with $\tau_p = 0$, the unseen-class scores $\mathcal{S}_i$ can be used for selecting high-quality pseudo-labels alone. However, it is still helpful to choose a higher threshold (*e.g.*, $\tau_p = 0.95$ as we adopt across the tasks). As for τ_q, we should choose a lower value (*e.g.*, $\tau_q \leq 0.75$) for fully utilizing the outliers with low confidence.

Table 4. Closed-set classification accuracy (%) of several methods in the standard SSL setting (presented in the column of "SSL") compared to the performance in the OSSL setting.

Task	CIFAR-50-200		CIFAR-50-1250	
Setting	OSSL	SSL	OSSL	SSL
FixMatch	43.94	45.64	68.92	72.74
SimMatch	49.98	51.76	69.70	**73.66**
OpenMatch	37.60	39.16	66.54	67.80
IOMatch	**56.14**	**55.94**	**69.84**	73.28

Table 5. Closed-set classification accuracy (%) of IOMatch extended with auxiliary self-supervised learning objectives.

Dataset	CIFAR100			
Class split	50 / 50		80 / 20	
Number of labels	4	25	4	25
IOMatch	56.14	69.84	49.89	64.28
w/ Contrastive	57.08	70.80	50.25	65.92
w/ Rotation	**58.92**	**71.54**	**50.90**	**66.50**

Decoupled Feature Spaces. Different from OpenMatch [25], we optimize the multi-binary classifier (and the open-set classifier) in a different feature space than the closed-set classifier, which is implemented by a projection head. We find experimentally that such design is important. For instance, if we put all the three classifiers in the same feature space (*i.e.*, directly connected to the backbone encoder), the performance on `CIFAR-50-200` and `CIFAR-50-1250` will be reduced by about 2.2% and 0.8%, respectively.

Performance on Standard SSL. We also evaluate the proposed IOMatch in the standard SSL setting where no outlier exists in unlabeled data. The results are presented in Table 4. It is shown that IOMatch is also a strong method for standard SSL, which can achieve significantly better performance when labels are scarce. When the number of labels is relatively more adequate, IOMatch can still achieve impressive performance comparable to that of advanced methods. Moreover, on the task `CIFAR-50-200`, the performance of IOMatch in the open-set setting is even better than that in the standard setting, which is made possible by the full exploitation of outliers.

Extensions of IOMatch. The inherent simplicity of IOMatch lends itself to the integration of other potent techniques within the framework, thereby further enhancing its performance. We explore the incorporation of self-supervised learning approaches that have exhibited remarkable effectiveness in previous methods [2, 20, 43]. Specifically, we adopt the contrastive learning objective from SimMatch [43] and the rotation recognition pretext task from ReMixMatch [2]. In the following, we introduce the implementation of the rotation recognition objective in the extended IOMatch, and the details about the contrastive learning objective can be found in the supplementary material. For each unlabeled image $\boldsymbol{u}_i$, we rotate $\boldsymbol{u}_i$ by an angle of $\angle_i$ degrees and obtain $\text{Rotate}(\boldsymbol{u}_i, \angle_i)$, where $\angle_i$ is sampled uniformly from $\angle_i \sim \{0, 90, 180, 270\}$. We add an auxiliary classifier $\theta(\cdot)$ (implemented as a fully connected layer) connected to the backbone encoder, which predicts the rotation degree among the four options, *i.e.*, $\boldsymbol{a} = \theta(f(\text{Rotate}(\boldsymbol{u}_i, \angle_i))) \in \mathbb{R}^4$. The rotation recognition loss is defined as:

$$\mathcal{L}_{rot} = \frac{1}{\mu B} \sum_{i=1}^{\mu B} \text{H}(\text{OneHot}(\angle_i), \boldsymbol{a}). \tag{10}$$

The results in Table 5 demonstrate substantial performance improvements stemming from these self-supervised add-ons. It shows that our proposed IOMatch is high extensible and has great potential for enhancement.

Training Efficiency. The network parameters (15.2M) of IOMatch are only about 3% more than those (14.7M) of FixMatch [28], which results in very little additional overhead. Besides, IOMatch does not require memory banks used in contrastive-based methods [20, 43], which significantly reduces the usage of GPU memory especially for large scale datasets. Therefore, IOMatch shows high training efficiency for both time and memory costs.

Limitations and Future Work. Finally, we would like to discuss the limitations of the current work as well as the future directions to further improve it. In the proposed IOMatch framework, we adopt the pre-defined fixed confidence thresholds fro all classes, which could be less flexible in more complex tasks. Inspired from recent works [34, 41], we will consider the dynamic threshold adjusting strategy for IOMatch. Besides, this work only considers the most common class space mismatch case, where the classes of labeled data form a subset of those in the unlabeled data. We will also explore other open-set scenarios, such as the intersectional mismatch, where not all labeled classes are present in the unlabeled data.

5. Conclusion

In this paper, we first investigate how unseen-class outliers affect the performance of the latest standard SSL methods and reveal why existing open-set SSL methods may fail when labels are extremely scarce. Inspired from the surprising fact that an unreliable outlier detector is more harmful than outliers themselves, we propose IOMatch, which adopts a novel unified paradigm for jointly utilizing open-set unlabeled data, without distinguishing exactly between inliers and outliers. Despite of its remarkable simplicity, IOMatch significantly outperforms current state-of-the-arts across various settings. We believe that the introduction of such simple but effective framework will facilitate the application of SSL methods in real-world practical scenarios.

References

[1] Philip Bachman, Ouais Alsharif, and Doina Precup. Learning with pseudo-ensembles. In *NeurIPS*, 2014. 2

[2] David Berthelot, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. In *ICLR*, 2020. 3, 4, 6, 7, 9, 12, 13

[3] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *NeurIPS*, 2019. 2, 6, 7

[4] Kay Henning Brodersen, Cheng Soon Ong, Klaas Enno Stephan, and Joachim M Buhmann. The balanced accuracy and its posterior distribution. In *ICPR*, pages 3121–3124, 2010. 6

[5] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-supervised learning (chapelle, o. et al., eds.; 2006) [book reviews]. *IEEE TNN*, 20(3):542–542, 2009. 1

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, pages 1597–1607, 2020. 13

[7] Yanbei Chen, Xiatian Zhu, Wei Li, and Shaogang Gong. Semi-supervised learning under class distribution mismatch. In *AAAI*, pages 3569–3576, 2020. 3, 6, 7, 12, 14

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *ICCV*, pages 248–255, 2009. 6, 7

[9] Yue Duan, Lei Qi, Lei Wang, Luping Zhou, and Yinghuan Shi. Rda: Reciprocal distribution alignment for robust semi-supervised learning. In *ECCV*, pages 533–549, 2022. 12

[10] Lan-Zhe Guo, Zhen-Yu Zhang, Yuan Jiang, Yu-Feng Li, and Zhi-Hua Zhou. Safe deep semi-supervised learning for unseen-class unlabeled data. In *ICML*, pages 3897–3906, 2020. 6, 7, 12, 14

[11] Lan-Zhe Guo, Zhi Zhou, and Yu-Feng Li. Robust deep semi-supervised learning: A brief introduction. *arXiv preprint arXiv:2202.05975*, 2022. 12

[12] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, pages 9729–9738, 2020. 13

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 7

[14] Rundong He, Zhongyi Han, Xiankai Lu, and Yilong Yin. Safe-student for safe deep semi-supervised learning with unseen-class unlabeled data. In *CVPR*, pages 14585–14594, 2022. 2, 3, 6, 7, 12, 14

[15] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *ICLR*, 2017. 3, 12

[16] Junkai Huang, Chaowei Fang, Weikai Chen, Zhenhua Chai, Xiaolin Wei, Pengxu Wei, Liang Lin, and Guanbin Li. Trash to treasure: harvesting ood data with cross-modal matching for open-set semi-supervised learning. In *ICCV*, pages 8310–8319, 2021. 2, 3, 6, 7, 12, 14

[17] Alex Krizhevsky et al. Learning multiple layers of features from tiny images. 2009. 6

[18] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *ICLR*, 2017. 2, 6, 7

[19] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *NeurIPS*, 2018. 12

[20] Junnan Li, Caiming Xiong, and Steven CH Hoi. Comatch: Semi-supervised learning with contrastive graph regularization. In *ICCV*, pages 9475–9484, 2021. 3, 6, 7, 9, 13

[21] Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. In *NeurIPS*, pages 21464–21475, 2020. 12

[22] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE TPAMI*, 41(8):1979–1993, 2018. 2, 6

[23] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011. 13

[24] Avital Oliver, Augustus Odena, Colin A Raffel, Ekin Dogus Cubuk, and Ian Goodfellow. Realistic evaluation of deep semi-supervised learning algorithms. In *NeurIPS*, 2018. 1, 6, 12

[25] Kuniaki Saito, Donghyun Kim, and Kate Saenko. Openmatch: Open-set consistency regularization for semi-supervised learning with outliers. In *NeurIPS*, 2021. 1, 2, 3, 4, 5, 6, 7, 9, 14

[26] Kuniaki Saito and Kate Saenko. Ovanet: One-vs-all network for universal domain adaptation. In *ICCV*, pages 9000–9009, 2021. 2, 4

[27] Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In *NeurIPS*, 2016. 2, 6

[28] 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. In *NeurIPS*, 2020. 1, 2, 5, 6, 7, 9

[29] Yiyou Sun, Chuan Guo, and Yixuan Li. React: Out-of-distribution detection with rectified activations. In *NeurIPS*, pages 144–157, 2021. 12

[30] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *NeurIPS*, 2017. 2, 6

[31] Jesper E Van Engelen and Holger H Hoos. A survey on semi-supervised learning. *Machine Learning*, 109(2):373–440, 2020. 3

[32] Haoqi Wang, Zhizhong Li, Litong Feng, and Wayne Zhang. Vim: Out-of-distribution with virtual-logit matching. In *CVPR*, pages 4921–4930, 2022. 12

[33] Yidong Wang, Hao Chen, Yue Fan, Wang Sun, Ran Tao, Wenxin Hou, Renjie Wang, Linyi Yang, Zhi Zhou, Lan-Zhe Guo, Heli Qi, Zhen Wu, Yu-Feng Li, Satoshi Nakamura, Wei Ye, Marios Savvides, Bhiksha Raj, Takahiro Shinozaki,

Bernt Schiele, Jindong Wang, Xing Xie, and Yue Zhang. Usb: A unified semi-supervised learning benchmark for classification. In *NeurIPS*, 2022. 3, 6

[34] Yidong Wang, Hao Chen, Qiang Heng, Wenxin Hou, Marios Savvides, Takahiro Shinozaki, Bhiksha Raj, Zhen Wu, and Jindong Wang. Freematch: Self-adaptive thresholding for semi-supervised learning. In *ICLR*, 2023. 3, 6, 7, 9

[35] Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. Generalized out-of-distribution detection: A survey. *arXiv preprint arXiv:2110.11334*, 2021. 12

[36] Lihe Yang, Wei Zhuo, Lei Qi, Yinghuan Shi, and Yang Gao. St++: Make self-training work better for semi-supervised semantic segmentation. In *CVPR*, 2022. 1

[37] Xiangli Yang, Zixing Song, Irwin King, and Zenglin Xu. A survey on deep semi-supervised learning. *IEEE TKDE*, pages 1–20, 2022. 3

[38] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. 13

[39] Qing Yu, Daiki Ikami, Go Irie, and Kiyoharu Aizawa. Multi-task curriculum framework for open-set semi-supervised learning. In *ECCV*, pages 438–454, 2020. 1, 2, 3, 6, 7, 12, 13, 14

[40] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *BMVC*, 2016. 6

[41] Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. In *NeurIPS*, 2021. 3, 6, 7, 9

[42] Zhen Zhao, Luping Zhou, Yue Duan, Lei Wang, Lei Qi, and Yinghuan Shi. Dc-ssl: Addressing mismatched class distribution in semi-supervised learning. In *CVPR*, pages 9757–9765, 2022. 12

[43] Mingkai Zheng, Shan You, Lang Huang, Fei Wang, Chen Qian, and Chang Xu. Simmatch: Semi-supervised learning with similarity matching. In *CVPR*, pages 14471–14481, 2022. 3, 6, 7, 9, 13

[44] Ronghang Zhu and Sheng Li. Crossmatch: Cross-classifier consistency regularization for open-set single domain generalization. In *ICLR*, 2022. 4

IOMatch: Simplifying Open-Set Semi-Supervised Learning with Joint Inliers and Outliers Utilization
Supplementary Material

A. Open-Set Semi-Supervised Learning Setting

A.1. Class Space Mismatch

Open-Set Semi-Supervised Learning (OSSL) assumes that labeled and unlabeled data have different class spaces, which can be referred by the term, *Class Space Mismatch*. Let $\mathcal{C}_l$ and $\mathcal{C}_u$ be the class sets of labeled and unlabeled data. Several pioneer works [7, 24] assume that $\mathcal{C}_l \nsubseteq \mathcal{C}_u$ and $\mathcal{C}_u \nsubseteq \mathcal{C}_l$, while more recent OSSL works [10, 14, 16, 39] focus on the case where $\mathcal{C}_l \subset \mathcal{C}_u$. For this point, we share a similar opinion with [11]: As it is usually much easier to collect unlabeled data than labeled data, it is more likely for unlabeled data to have more categories than labeled data. Thus, we assume $\mathcal{C}_l \subset \mathcal{C}_u$ in this work.

Remark. A broader concept is *Class Distribution Mismatch* [9, 42]. If we denote the marginal class distributions of labeled and unlabeled data as $\boldsymbol{p}_l(y)$ and $\boldsymbol{p}_u(y)$, then the class distribution mismatch in SSL indicates that $p_l(y) \neq p_u(y)$. The class space mismatch can be also viewed as such a case, where $p_l(y \in \mathcal{C}_u/\mathcal{C}_l) = 0 \neq p_u(y \in \mathcal{C}_u/\mathcal{C}_l)$. In this work, we just focus on the class space mismatch, which is the most common and problematic case of class distribution mismatch [11].

A.2. Connections to Out-of-Distribution Detection

Out-of-distribution (OOD) detection [15] aims to detect OOD samples existing in test data by assigning higher OOD scores to OOD samples than ID samples. Representative works design the OOD scores using the predicted logits and probabilities [21, 29], or using the information in feature space [19], or combining both of them [32]. More comprehensive reviews can be found in [35].

Although unseen-class outliers can be also regarded as a kind of OOD samples, OOD detection is largely different from open-set SSL in the following aspects. Firstly, OOD detection tasks usually assume that sufficient labeled ID samples are provided for training (and no OOD sample exists), which cannot be satisfied in OSSL. It is a key reason why OOD detection methods cannot be directly applied in OSSL for detecting outliers. Secondly, the main objective of OOD detection is to separate OOD samples from ID samples, which can viewed as a binary classification task. However, the motivation of OSSL is to fully exploit open-set unlabeled samples for improving the model's performance on multi-class classification tasks. Therefore, a model good at OOD detection could not perform well on ID (seen-class) classification. This is the reason why we adopt Balanced Accuracy (BA) rather than AUROC, which is widely used in OOD detection, for open-set evaluation.

B. Distribution Alignment Strategy

For the distribution alignment (DA) strategy, we simply follow the implementation from ReMixMatch [2]. Specifically, we maintain a running average of the model's predictions on unlabeled data, denoted by $\boldsymbol{p}_{avg}$. The marginal class distribution $\boldsymbol{p}_{mrgl}$ is estimated based on the labeled samples in training (which is the uniform distribution in our setting). Given the model's prediction $\boldsymbol{p}_i^w = \phi(f(\mathcal{T}_w(\boldsymbol{u}_i)))$ on an weakly augmented unlabeled sample $\mathcal{T}_w(\boldsymbol{u}_i)$, we scale $\boldsymbol{p}_i^w$ by the ratio $\boldsymbol{p}_{mrgl}/\boldsymbol{p}_{avg}$ and normalize the result as a valid probability distribution:

$$\widetilde{\boldsymbol{p}}_i = \text{Normalize}(\boldsymbol{p}_i^w \cdot \frac{\boldsymbol{p}_{mrgl}}{\boldsymbol{p}_{avg}}), \tag{11}$$

where $\text{Normalize}(\boldsymbol{p})_i = p_i / \sum_j p_j$. $\boldsymbol{p}_i^w$ is then used as the seen-class prediction for producing the unified open-set target and training the closed-set classifier via pseudo-labeling. $\boldsymbol{p}_{avg}$ is computed with the predictions over the last 128 batches.

In practice, we find the DA strategy is effective when the number of classes is relatively large (*e.g.*, for CIFAR-100 and ImageNet-30). However, for CIFAR-10 with fewer classes, the DA strategy may lead to performance degradation instead. The reason could be that the presence of unseen-class outliers interferes with the estimation of $\boldsymbol{p}_{avg}$. Thus, we do not apply the DA strategy in the tasks on CIFAR-10.

C. Extensions with Self-Supervision

IOMatch is such a simple framework that we can easily incorporate other powerful techniques with it to further improve the performance. Recently, self-supervised learning objectives including pretext tasks [**?**] and contrastive learning [6, 12] have shown strong performance in SSL [2, 20, 43]. We find experimentally that the self-supervised modules can also bring performance gains to IOMatch (see Table 5 in the paper). Here we introduce the details of the extensions of IOMatch.

It is quite easy to incorporate the rotation recognition pretext task with IOMatch. For each unlabeled image $\boldsymbol{u}_i$, we rotate $\boldsymbol{u}_i$ by an angle of $\angle_i$ degrees and obtain $\text{Rotate}(\boldsymbol{u}_i, \angle_i)$, where $\angle_i$ is sampled uniformly from $\angle_i \sim \{0, 90, 180, 270\}$. We add an auxiliary classifier $\theta(\cdot)$ (implemented as a fully connected layer) connected to the backbone encoder, which predicts the rotation degree among the four options, *i.e.*, $\boldsymbol{a} = \theta(f(\text{Rotate}(\boldsymbol{u}_i, \angle_i))) \in \mathbb{R}^4$. The rotation prediction loss is defined as:

$$\mathcal{L}_{rot} = \frac{1}{\mu B} \sum_{i=1}^{\mu B} \text{H}(\text{OneHot}(\angle_i), \boldsymbol{a}). \tag{12}$$

We implement the contrastive learning objective following SimMatch [43]. Given the projected features of all labeled samples $\{\boldsymbol{z}_l : l \in (1, \dots, N_l)\}$ (maintained in a memory bank), the instance similarities between each unlabeled sample $\boldsymbol{u}_i$ and all labeled samples are defined as $\boldsymbol{r}_i$:

$$\boldsymbol{r}_{i,l}^{w/s} = \frac{\exp(\text{sim}(\boldsymbol{z}_i^{w/s}, \boldsymbol{z}_l))}{\sum_{j=1}^{N_l} \exp(\text{sim}(\boldsymbol{z}_i^{w/s}, \boldsymbol{z}_j))}, \tag{13}$$

where $\text{sim}(\boldsymbol{u}, \boldsymbol{v}) = \boldsymbol{u}^\intercal \boldsymbol{v} / \|\boldsymbol{u}\| \|\boldsymbol{v}\|$, and $t = 0.1$ is the temperature parameter. The similarity target $\widetilde{\boldsymbol{r}}$ is then generated by scaling $\boldsymbol{r}_i^w$ with $\widetilde{\boldsymbol{p}}_i$. The contrastive loss is defined as:

$$\mathcal{L}_{con} = \frac{1}{\mu B} \sum_{i=1}^{\mu B} \text{H}(\widetilde{\boldsymbol{r}}_i, \boldsymbol{r}_i^s). \tag{14}$$

As the above two self-supervised objectives are both standard cross-entropy losses, we can simply add them to the total loss with the weights $\mathcal{L}_{rot} = \mathcal{L}_{con} = 1$. In spite of the promising results, the extensions of IOMatch introduce extra network modules (*e.g.*, the rotation classifier and the memory bank) and thus extra training costs. It is noteworthy that, as a simple yet effective OSSL framework, IOMatch can outperform the complicated baselines on most tasks even without these extra learning objectives.

D. Inference

We use the standard closed-set classifier for the inference in the closed-set classification task, in order to ensure fair comparisons with other baselines. In fact, the open-set classifier can also be used for closed-set classification by ignoring the last item of $\boldsymbol{q}_t$. We find experimentally that in this case, the predictions made by $\phi(\cdot)$ and $\psi(\cdot)$ are mostly the same. The difference in closed-set accuracy is usually less than 0.5%. In the paper, we evaluate the closed-set performance using the closed-set classifier to keep consistent with other methods. However, we can just employ a single open-set classifier $\psi(\cdot)$ for both the close-set and open-set classification tasks for the sake of simplicity.

E. Open-Set Evaluation with Foreign Outliers

We have performed open-set evaluation with the test sets of CIFAR-10/100 (see Table 2 of the paper), which consist of all seen and unseen classes observed during training. In such case, unseen-class outliers in testing are similar to those in training. As the seen and unseen classes come from the same dataset, we denote them as the **intra-dataset** test data. Here we also consider the **inter-class** case where additional foreign outliers come from different datasets than CIFAR10/100. In particular, we add samples from SVHN [23], LSUN [38], and synthetic Gaussian and uniform noise images [39] as part of the testing data.

The results are shown in 6. Since the added foreign outliers are more dissimilar to the inliers, they are easier to identify. Therefore, the open-set accuracy on the inter-dataset test data is a little higher than that on the intra-dataset test data, while the difference is not significant.

Table 6. Open-set classification balanced accuracy (%) on the **inter-dataset** open-set test data, which contain samples from different datasets than CIFAR10/100.

	Dataset		CIFAR-10		CIFAR-100					
	Class split (Seen / Unseen)		6 / 4		20 / 80		50 / 50		80 / 20	
	Number of labels per class		4	25	4	25	4	25	4	25
Open-Set SSL	UASD [7]	AAAI'20	18.32 ± 0.61	35.78 ± 0.22	11.03 ± 0.43	27.35 ± 0.33	7.03 ± 0.45	31.94 ± 0.74	5.92 ± 0.35	27.83 ± 0.85
	DS3L [10]	ICML'20	31.38 ± 0.52	40.92 ± 0.68	13.05 ± 1.03	35.03 ± 0.47	11.84 ± 0.79	34.88 ± 0.57	11.38 ± 0.89	29.32 ± 0.38
	MTCF [39]	ECCV'20	28.35 ± 4.84	46.06 ± 0.69	8.16 ± 2.12	26.77 ± 3.70	4.14 ± 0.38	38.04 ± 0.15	1.46 ± 0.17	30.51 ± 0.27
	T2T [16]	ICCV'21	51.35 ± 1.76	61.78 ± 0.89	17.82 ± 1.57	37.78 ± 0.73	12.33 ± 1.87	43.86 ± 0.71	34.45 ± 0.67	51.77 ± 1.03
	OpenMatch [25]	NeurIPS'21	14.37 ± 0.05	20.31 ± 3.49	8.77 ± 2.83	39.96 ± 1.17	9.97 ± 0.37	49.56 ± 1.15	6.31 ± 0.88	44.77 ± 0.58
	SAFE-STUDENT [14]	CVPR'22	46.37 ± 0.61	54.23 ± 0.42	16.31 ± 0.88	29.44 ± 0.56	23.31 ± 0.93	46.91 ± 1.42	29.52 ± 0.55	50.83 ± 0.41
	IOMatch	**Ours**	**77.82 ± 2.48**	**82.44 ± 0.54**	**46.97 ± 2.05**	**60.30 ± 0.99**	**46.09 ± 1.98**	**60.64 ± 0.79**	**40.08 ± 0.75**	**54.57 ± 0.30**