

SCOMatch: Alleviating Overtrusting in Open-set Semi-supervised Learning

Zerun Wang¹, Liuyu Xiang², Lang Huang¹, Jiafeng Mao¹,
Ling Xiao¹, and Toshihiko Yamasaki¹

¹ The University of Tokyo

{ze_wang, langhuang, ling, yamasaki}@cvm.t.u-tokyo.ac.jp
mao@hal.t.u-tokyo.ac.jp

² Beijing University of Posts and Telecommunications
xiangly@bupt.edu.cn

Abstract. Open-set semi-supervised learning (OSSL) leverages practical open-set unlabeled data, comprising both in-distribution (ID) samples from seen classes and out-of-distribution (OOD) samples from unseen classes, for semi-supervised learning (SSL). Prior OSSL methods initially learned the decision boundary between ID and OOD with labeled ID data, subsequently employing self-training to refine this boundary. These methods, however, suffer from the tendency to overtrust the labeled ID data: the scarcity of labeled data caused the distribution bias between the labeled samples and the entire ID data, which misleads the decision boundary to overfit. The subsequent self-training process, based on the overfitted result, fails to rectify this problem. In this paper, we address the overtrusting issue by treating OOD samples as an additional class, forming a new SSL process. Specifically, we propose SCOMatch, a novel OSSL method that 1) selects reliable OOD samples as new labeled data with an OOD memory queue and a corresponding update strategy and 2) integrates the new SSL process into the original task through our **S**imultaneous **C**lose-set and **O**pen-set self-training. SCOMatch refines the decision boundary of ID and OOD classes across the entire dataset, thereby leading to improved results. Extensive experimental results show that SCOMatch significantly outperforms the state-of-the-art methods on various benchmarks. The effectiveness is further verified through ablation studies and visualization. Our code will be available at <https://github.com/komejisatori/SCOMatch>.

Keywords: Open-set problem · Semi-supervised learning

1 Introduction

Semi-supervised learning (SSL) [36] significantly enhances the efficacy of numerous computer vision tasks by leveraging a large number of available unlabeled data for training. A common approach is to train the model on labeled data while using the model to assign pseudo-labels to unlabeled data for self-training.

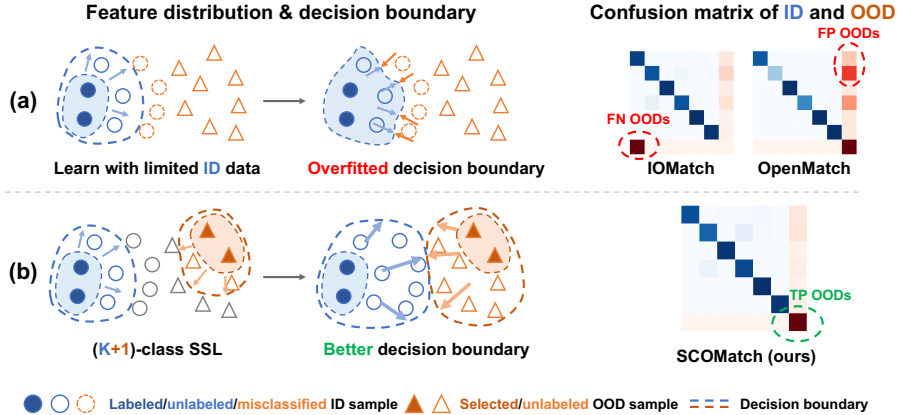


Fig. 1: Comparison between prior OSSL methods and SCOMatch on CIFAR-10 with six ID classes. **(a):** Prior methods overtrust labeled ID data, leading to overfitted decision boundaries that self-training cannot rectify. This results in more false positive or negative OODs in the red circles of the confusion matrix. **(b):** SCOMatch selects reliable OOD samples for $(K+1)$ -class SSL, achieving higher accuracy for both ID and OOD classes.

Conventional SSL methods [18, 27, 29, 34, 35] assume that the classes in unlabeled data are the same as those in labeled data, a.k.a., in-distribution (ID) classes. This assumption greatly differs from real-world applications, where unlabeled data often contains some unseen classes, *i.e.*, the out-of-distribution (OOD) class. Samples from the OOD class can affect the SSL performance, as the model can only assign ID classes to the samples, leading to the model learning incorrect information.

Open-set semi-supervised learning (OSSL) methods [7, 9, 11, 14, 31] are proposed to address this issue. The essence of OSSL lies in identifying OOD samples in unlabeled data so that they can be excluded during the SSL process. Previous OSSL methods typically obtain the ability to identify OOD samples by learning from labeled ID data: they apply OOD detection [19, 24] or prototype learning [20] methods to learn from the distribution of labeled ID data and get the decision boundary that distinguishes ID and OOD samples. Recent OSSL methods [12, 24] further apply self-training methods to improve this decision boundary on unlabeled data.

These methods have improved the effectiveness of SSL using open-set unlabeled data. However, they overtrust the labeled ID data. As illustrated in Figure 1(a): in OSSL, labeled ID data is limited, leading to a distribution bias with the ID data in the entire dataset. As a result, they learn a suboptimal decision boundary overfitted to the labeled data. The following self-training can only be based on this overfitted boundary, thus failing to address this issue. The overtrust problem results in more false-positive or false-negative OODs, as shown in the confusion matrices of Figure 1(a), thereby compromising the performance.

Meanwhile, we noticed that overtrusting does not exist in conventional SSL tasks. We found that the key difference is that in conventional SSL, all classes have labeled data from the beginning. Consequently, each class can start to refine its decision boundary rather than depending on the boundary of other classes. Inspired by this, we aim to treat ‘OOD’ as a new class with its own labeled data and form a new $(K+1)$ -class SSL process (K is the number of ID classes). As illustrated in Figure 1(b): we first select some high-quality OOD samples as the labeled data of the $(K+1)$ -th class. Supervised by this extended labeled dataset, we start a $(K+1)$ -class SSL process. Thus, both ID and OOD classes can refine their decision boundary independently across the entire dataset. This results in better discrimination of both ID and OOD as shown in the confusion matrix of Figure 1(b). While some recent OSSL works [19, 20] also treat ‘OOD’ as a new class, their OOD samples are still based on the decision boundary learned with limited labeled ID data. Thus, their methods do not deviate from the scope of previous methods that overtrust labeled data while ours can alleviate this issue.

Following our solution of forming a $(K+1)$ -class SSL process, there are two challenges. First, how to select reliable OOD samples as the new labeled data, since all the OODs were originally unlabeled in this task. Second, how to integrate the new $(K+1)$ -class SSL task into the original K -class task, as the latter is the objective of the OSSL task, and the new task should not affect its performance. To address these challenges, we propose SCOMatch, which consists of the following key designs: 1) for selecting reliable OOD samples, we propose employing an OOD memory queue with a corresponding update strategy to maintain reliable OOD samples as additional labeled data; 2) for integrating the new $(K+1)$ -class SSL with the original task, we propose a simultaneous close-set and open-set self-training process, allowing the open-set and close-set self-training to be jointly optimized on a single classification head.

The experimental results on various benchmarks show that SCOMatch significantly outperforms prior methods and achieves state-of-the-art performance. We also prove the effectiveness of our proposed components through ablation studies and analysis.

Overall, our contributions can be summarized as follows:

- We identify the issue of overtrusting limited labeled ID data in prior OSSL methods and alleviate it by treating OOD as a new class and forming a new $(K+1)$ -class SSL process.
- We propose a novel OSSL method, SCOMatch, which tackles the challenges of reliable OOD sample selection and integrates the new $(K+1)$ -class SSL with the original task through the designed components.
- SCOMatch significantly outperforms prior OSSL methods on various OSSL benchmarks. In particular, we improve the close-set accuracy by 13.4% on TinyImageNet. We also conduct comprehensive experiments and visualization to prove the effectiveness of each component.

2 Related Work

2.1 Semi-supervised learning

Semi-supervised learning (SSL) has attracted considerable attention in recent years. The mainstream SSL methods typically train the model on labeled data while employing consistency regularization [1] to utilize unlabeled data. FixMatch [27] is one of the influential SSL methods. It summarizes some early works [2, 3, 23, 30] and provides a simple and efficient framework. Some of the following SSL methods contribute to various parts of this framework: FlexMatch [34] and FreeMatch [29] dynamically adjust each class’s threshold during different training stages based on learning performance. CoMatch [18] and SimMatch [35] apply contrastive learning strategies to improve semantic similarity in the same class, thus improving the pseudo-label quality. Some works also focus on addressing issues such as class-imbalance [10, 32] and domain adaptation [4, 6]. However, all these SSL methods are based on the assumption that no OOD samples exist in unlabeled data. Therefore, the OOD samples in real-world unlabeled data will affect their performance.

2.2 Open-set semi-supervised learning

Open-set semi-supervised learning, also called safe semi-supervised learning, aims to address the problem of OOD samples in unlabeled data. The key point is to identify OOD samples. Early works attempt to remove OOD samples or mitigate their effect on the SSL process. DS3L [11] assigns a learnable weight on unlabeled data, thereby reducing the contribution of OOD samples to the training loss. UASD [7] applies a threshold on the classification confidence score to identify and remove OOD samples during SSL. MTCTF [31] applies a binary classification head to distinguish ID and OOD samples. T2T [14] uses a cross-modal matching branch for filtering out OOD samples. Later works discover that the OOD samples in unlabeled data, once identified, can also be used for training. OpenMatch [24] applies the one-vs-all (OVA) classification head [25] to detect OOD samples. The detected samples will be used for SSL via consistency regularization. SAFE-STUDENT [12] uses energy discrepancy for labeling OOD samples. Then, the model will be trained to increase the uncertainty of these samples. These methods have improved the OSSL performance. However, as the OOD classification modules are trained on labeled data first, they face the overtrusting issue, as we pointed out.

Although several recent works treat ‘OOD’ as a new class, they do not deviate from the scope of past methods. IOMatch [19] calculates the open-set probability score using the OVA head’s output and then optimizes an $(K+1)$ -class classification head. However, the OVA head still relies on labeled ID data for training as the same in OpenMatch. Ma *et al.* [20] apply the prototype learning method to solve the OSSL problem. They use the labeled data to get prototypes for each ID class first and use the fixed prototypes for labeling ID and OOD samples and train a $(K+1)$ -way classification head. It is also obvious that the prototypes

are based on labeled data only. In contrast, we don't trust the result trained on labeled ID data only. Instead, we start a new $(K+1)$ -class SSL process with labeled IDs and selected OODs simultaneously from the beginning.

OSSL methods also take some techniques from the OOD detection task [13]. However, OOD detection aims to train on a large number of labeled ID data to distinguish OOD samples, which is different from the OSSL setting.

3 Methodology

We aim to solve the overtrusting problem by treating OOD as a new class with its own labeled data and forming a $(K+1)$ -class SSL task. To achieve this, we propose to 1) select reliable OOD samples as the new labeled data and 2) integrate the new $(K+1)$ -class SSL into the original K -class task. In this section, we first introduce the preliminary of SSL, then the overall framework of SCOMatch, and finally, we separately explain how we select reliable OOD samples and combine the two tasks.

3.1 Preliminary

OSSL aims to solve the open-set problem within the SSL frameworks. In this setting, we consider the models trained on both labeled dataset $\mathcal{D}^x = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ and unlabeled dataset $\mathcal{D}^u = \{(\mathbf{u}_i)\}_{i=1}^M$. Here, \mathbf{x}_i and \mathbf{u}_i represent the images, $y_i \in \{1, 2, \dots, K\}$ denotes the label within K ID classes, and $N \ll M$ since the labeled data is limited. Similar to most recent OSSL works, we utilize some basic SSL techniques from FixMatch [27]. Suppose that we have a backbone \mathcal{F} and a classification head \mathcal{C} parametrized by weights \mathbf{C} and subsequently normalized by a softmax function. In each training iteration, we have a batch of B labeled samples $\mathcal{X} = \{(\mathbf{x}_b, y_b)\}_{b=1}^B$ for supervised training, and a batch of μB unlabeled samples $\mathcal{U} = \{(\mathbf{u}_b)\}_{b=1}^{\mu B}$ for self-training. Here, μ controls the relative size of \mathcal{X} and \mathcal{U} .

For the supervised loss on labeled data \mathcal{L}_s , we apply a weak augmentation on images and use the standard cross-entropy H to optimize the model:

$$\mathcal{L}_s = \frac{1}{B} \sum_{i=1}^B H(y_i, \mathcal{C}(\mathcal{F}(\mathbf{x}_i^w))), \quad (1)$$

where \mathbf{x}_i^w denotes the weakly-augmented images.

For the self-training loss on unlabeled data \mathcal{L}_u , we first get the model's prediction given an unlabeled sample with weak augmentation: $\mathbf{q}_i = \mathcal{C}(\mathcal{F}(\mathbf{u}_i^w))$. Then, we apply thresholding to select those samples with high confidence, instantiated by an indicator function $\mathbb{1}(\max(\mathbf{q}_i) > \tau)$ where τ is a manual threshold. The selected predictions are then converted to one-hot pseudo-labels $\hat{q}_i = \arg \max(\mathbf{q}_i)$. For training, we apply strong augmentation on the same samples and use cross-entropy for consistency regularization:

$$\mathcal{L}_u = \frac{1}{\mu B} \sum_{i=1}^{\mu B} \mathbb{1}(\max(\mathbf{q}_i) > \tau) H(\hat{q}_i, \mathcal{C}(\mathcal{F}(\mathbf{u}_i^s))), \quad (2)$$

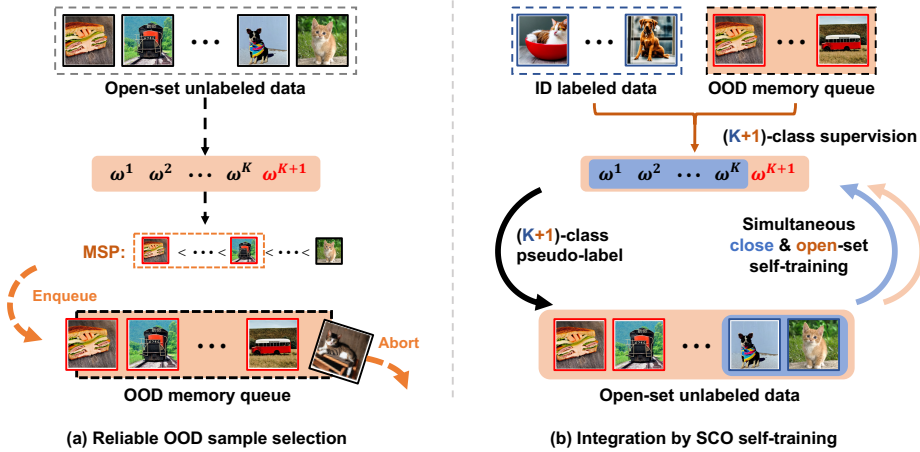


Fig. 2: The training process of SCOMatch. The $(K+1)$ -classification head is the only head structure in our model (the backbone is not depicted for simplification). **(a)** The OOD sample selection by our proposed OOD memory queue and corresponding update strategy. **(b)** The integration of the original K -class task and the new $(K+1)$ -class SSL by our simultaneous close-set and open-set self-training. These two processes function concurrently with the same model, but we separate them for the clarity of explanation. Here, the animal classes are ID and others are OOD.

where \mathbf{u}_i^s denotes the unlabeled image with strong augmentation.

The model is optimized by the total combined loss of $\mathcal{L}_s + \lambda \mathcal{L}_u$, where λ is the hyperparameter controlling the trade-off between these two terms. Following prior works, the Exponential Moving Average (EMA) [28] technique is applied to enhance training stability.

3.2 Overall framework

We propose SCOMatch with 1) the OOD memory queue and corresponding update strategy and 2) simultaneous close-set and open-set self-training (SCO self-training) to alleviate overtrusting. Figure 2 presents the structure and training process of our SCOMatch. The $(K+1)$ -class classification head \mathcal{C} with parameters $\mathbf{C} = [\mathbf{w}^1, \mathbf{w}^2, \dots, \mathbf{w}^{K+1}]$ is the only head structure in SCOMatch, where first K elements are for the K ID classes and the last dimension \mathbf{w}^{K+1} is newly introduced for the ‘OOD’ class. Figure 2(a) illustrates the OOD memory queue and corresponding data strategy for reliable OOD sample selection. Figure 2(b) illustrates the integration of original K -class tasks and the new $(K+1)$ -class SSL task by our SCO self-training. We provide detailed introductions to each part in the subsequent sub-sections.

3.3 Reliable OOD sample selection

The cornerstone of our proposed method is to construct a new $(K+1)$ -class dataset. This necessitates the selection of reliable, high-quality OOD samples. However, there are two major problems: 1) All the OOD samples are unlabeled, inevitably introducing noise during the selection; 2) Overtrusting, as in prior OSSL methods, should be avoided when we select from the unlabeled data. To alleviate these issues, we propose the following well-designed OOD memory queue and corresponding update strategy to obtain reliable OOD samples, as illustrated in Figure 2(a).

The OOD memory queue. We construct a first-in first-out memory queue $\mathcal{D}^m = \{(\mathbf{o}_i, y_i^{ood})\}_{i=1}^{N_m}$ with a relatively small size N_m to store the selected images $\{\mathbf{o}_i\}$ that belong to the OOD class with $y^{ood} = K + 1$.

Update strategy. Prior works use the OOD detection head or scoring methods, which are learned with limited ID labeled data, to detect OODs. In contrast, we seek to leverage the knowledge of the entire $(K+1)$ -class task to avoid overtrusting. Considering Maximum Softmax Probability (MSP) [13] is a reliable scoring method for detecting OODs, we adapt it to our $(K+1)$ -class head \mathcal{C} to select OODs from unlabeled data:

$$\text{MSP}^{\mathcal{C}}(\mathbf{u}) = \max_{i \in \{1, \dots, K\}} \mathcal{C}(\mathcal{F}(\mathbf{u}^w))^i, \quad (3)$$

which means that we consider the maximum probability among the first K classes of the entire $(K+1)$ -class head.

During training, we sort the MSP of samples in each unlabeled batch and push K_m samples with the smallest MSP into the OOD memory queue \mathcal{D}^m , since samples with smaller MSP are more likely to be OOD. Meanwhile, the oldest sample in the queue will be aborted since it is first-in first-out.

Our proposed OOD memory queue and update strategy can tackle the above challenges: 1) The first-in first-out scheme with relatively small N_m avoids the noisy or overtrusted sample remaining in the queue for a long time. 2) We keep K_m relatively small. Therefore, only the samples with the lowest MSP score (indicating high confidence of being OOD) will be selected. Experimental results prove our design’s effectiveness and the selected samples’ quality.

3.4 Integration by SCO self-training

With the selected reliable OOD samples, we construct a new $(K+1)$ -class labeled data for OSSL. However, the K -class close-set and $(K+1)$ -class open-set tasks inherently conflict as they attempt to cluster the embedded features from the backbone into different distributions. We empirically found that the two tasks will interfere with each other if we follow the usual multi-task paradigm and use two separate heads for each task.

This is solved by our proposed SCO self-training as illustrated in Figure 2(b): with the new labeled data, the model is only supervised by the $(K+1)$ -class task. Then, the model is used for generating $(K+1)$ -class pseudo-label. After that, we

simultaneously use the first K -class pseudo-labels for close-set self-training on \mathcal{C}^K and all the $(K+1)$ -class pseudo-labels for open-set self-training on \mathcal{C} . Our SCO self-training allows two classification tasks with conflicting class numbers to be trained together and jointly serve the ultimate goal of the OSSL task.

It is worth noting that while prior OSSL methods are supervised by K -class labeled ID data, our head \mathcal{C} is only supervised by the new $(K+1)$ -class labeled data combination, which avoids the head overfitting to the labeled ID data. Meanwhile, although OOD samples actually originate from multiple categories, we find that treating OODs as a single class can address the overtrusting issue and enhance the effectiveness of OSSL. This is because the goal of OSSL tasks is not distinctions within OOD categories, and having the model pay additional attention to classifying among OODs within a limited parameter space could potentially affect the distinction between ID classes.

(K+1)-class supervision. For each iteration, we use one batch of labeled ID data $\mathbf{x} \in \mathcal{X}$ and randomly select the same number of OOD samples from the OOD memory queue $\mathbf{o} \in D^m$ for $(K+1)$ -class supervision on the entire head \mathcal{C} :

$$\mathcal{L}_s^{id} = \frac{1}{B} \sum_{i=1}^B H(y_i, \mathcal{C}(\mathcal{F}(\mathbf{x}_i^w))), \quad (4)$$

$$\mathcal{L}_s^{ood} = \frac{1}{B} \sum_{i=1}^B H(y_i^{ood}, \mathcal{C}(\mathcal{F}(\mathbf{o}_i))). \quad (5)$$

The two losses, \mathcal{L}_s^{id} for using labeled ID data and \mathcal{L}_s^{ood} for using selected OOD samples, are combined together as the supervision loss \mathcal{L}_s in SSL to train the entire head. We use the cross-entropy loss following Eq. 1.

(K+1)-class pseudo-labeling. With the supervision of our $(K+1)$ -class labeled data, we use \mathcal{C} for $(K+1)$ -class pseudo-labeling with the thresholding method $\mathbb{1}(\max(\mathbf{q}_i) > \tau')$ defined in the preliminary. Here, $\tau' = \{\tau, \tau^{ood}\}$ denotes that we use τ^{ood} of the new OOD class and the original SSL threshold τ for ID classes. Following other OSSL methods, we adjust the threshold τ^{ood} dynamically since the OOD class may actually contain more than one unseen class, its learning difficulty is different from ID classes. Inspired by the Curriculum Pseudo Labeling (CPL) [34], we adjust the threshold τ^{ood} by:

$$\tau_t^{ood} = \frac{\sigma_t(c^{ood})}{\sum_{i=1}^K \sigma_t(c^i)} \cdot \tau, \quad (6)$$

where $\sigma_t(c^{ood})$ denotes the number of OOD samples with the confidence score above the original threshold τ . $\sum_{i=1}^K \sigma_t(c^i)$ denotes the number of samples above τ across all the ID classes. At training iteration t , the threshold of the OOD class τ_t^{ood} is scaled by the ratio between $\sigma_t(c^{ood})$ and $\sum_{i=1}^K \sigma_t(c^i)$. We set upper and lower bounds $[\tau_{min}, \tau]$ for τ^{ood} to avoid unstable training.

Open-set self-training. We directly use the $(K+1)$ -class pseudo label on \mathcal{C} for open-set self-training. We use both weak-augmented and strong-augmented

Algorithm 1 Training process of our SCOMatch**Input:** Labeled data: $\mathcal{D}^x = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$, Unlabeled data: $\mathcal{D}^u = \{(\mathbf{u}_i)\}_{i=1}^M$ **Parameters:** SSL: τ, λ, α ; SCOMatch: number of enqueue samples K_m **Output:** Teacher and student model $\theta_t, \theta_s = \{\mathcal{F}, \mathcal{C}\}$.

- 1: **for** semi-supervised learning iterations **do**
- 2: Update the smallest K_m unlabeled samples into the OOD memory queue based on MSP in Eq. 3.
- 3: Compute supervision loss \mathcal{L}_s^{id} with labeled ID data by Eq. 4.
- 4: Get OOD samples from OOD memory queue: $\mathcal{D}^m = \{(\mathbf{o}_i, y_i^{ood})\}_{i=1}^{N_m}$.
- 5: Compute supervision loss \mathcal{L}_s^{ood} with selected OOD data by Eq. 5.
- 6: Compute open-set self-training loss \mathcal{L}_u^{open} by Eq. 7.
- 7: Compute close-set self-training loss \mathcal{L}_u^{close} by Eq. 8.
- 8: Jointly training student model with the loss \mathcal{L} by Eq. 9.
- 9: Update the threshold for the OOD class τ^{ood} using CPL in Eq. 6.
- 10: Update student model: $\theta_s \leftarrow \theta_s - \nabla \mathcal{L}$.
- 11: Update teacher model by EMA: $\theta_t \leftarrow \alpha \theta_t + (1 - \alpha) \theta_s$
- 12: **end for**
- 13: **return** Parameters θ_t, θ_s of teacher and student model

images $\mathbf{u}_i^w, \mathbf{u}_i^s$:

$$\mathcal{L}_u^{open} = \frac{1}{2\mu B} \sum_{i=1}^{\mu B} \mathbb{1}(\max(\mathbf{q}_i) > \tau') \sum_{\mathbf{u}_i \in \{\mathbf{u}_i^w, \mathbf{u}_i^s\}} H(\hat{q}_i, \mathcal{C}(\mathcal{F}(\mathbf{u}_i))), \quad (7)$$

where $\mathbf{q}_i = \mathcal{C}(\mathcal{F}(\mathbf{u}_i^w))$ and $\hat{q}_i = \arg \max(\mathbf{q}_i)$ denotes the pseudo-label process defined in the preliminary.

Close-set self-training. As the ultimate goal of OSSL is to better classify ID samples, we find conducting additional close-set self-training on \mathcal{C}^K , which is the first K dimensions of \mathcal{C} , rather than another individual head can further enhance performance. This may be because the ID samples further correct the distribution of \mathcal{C}^K , and improve the accuracy of selecting OOD samples for the OOD memory queue based on MSP. For the close-set self-training loss \mathcal{L}_u^{close} , we filter the OODs in pseudo label:

$$\mathcal{L}_u^{close} = \frac{1}{\mu B} \sum_{i=1}^{\mu B} \mathbb{1}(\hat{q}_i \leq K) \mathbb{1}(\max(\mathbf{q}_i) > \tau') H(\hat{q}_i, \mathcal{C}^K(\mathcal{F}(\mathbf{u}_i^s))), \quad (8)$$

where $\mathbb{1}(\hat{q}_i \leq K)$ denotes that we filter out the OODs in pseudo-label.

Finally, the model is trained with the losses mentioned above:

$$\mathcal{L} = \underbrace{\mathcal{L}_s^{id} + \mathcal{L}_s^{ood}}_{(K+1)\text{-class supervision}} + \lambda \underbrace{(\mathcal{L}_u^{open} + \mathcal{L}_u^{close})}_{\text{SCO self-training}}, \quad (9)$$

where λ controls the weight of self-training losses. Algorithm 1 summarizes the whole training process.

4 Experiments

4.1 Datasets and metrics

Datasets. We construct OSSL benchmarks following Ma *et al.* [20] and OpenMatch [24] with five datasets: MNIST [16], CIFAR-10 [15], CIFAR-100 [15], TinyImageNet [8], and IN-30 (a subset of ImageNet [8] containing 30 classes).

Following Ma *et al.* [20], we fix the number of labeled and unlabeled images and select various mismatch ratios, which means the ratio of OOD samples in the unlabeled data. For example, a mismatch ratio of 0.3 means 30% of the unlabeled data are OODs. In detail, we choose six ID classes (‘0’-‘5’) with ten labeled samples per class and 30,000 unlabeled samples for MNIST; six ID classes (the animal classes) with 400 labeled samples per class and 20,000 unlabeled samples for CIFAR-10; the first 50 classes as ID with 100 labeled samples per class and 40000 unlabeled samples for CIFAR-100; the first 100 classes as ID with 100 samples per class and 40,000 unlabeled samples for TinyImageNet.

Following OpenMatch [24], we use all images in each dataset and select various numbers of labeled images and ID classes. For CIFAR-10, we choose the same six ID classes as above. For CIFAR-100, we choose the first 55/80 classes as ID (the classes are sorted by their super-classes first). For IN-30, we choose the first 20 classes as ID.

Evaluation metrics. Following previous OSSL methods, we use close-set classification accuracy, open-set classification accuracy, and Area under the ROC Curve (AUC) [13] as evaluation metrics. We use the result of \mathcal{C}^K for close-set classification accuracy and \mathcal{C} for the other two metrics. Close-set classification accuracy considers only the accuracy of ID classes in test data. The open-set classification accuracy and AUC additionally measure the ability to identify OOD data. Open-set classification accuracy considers the accuracy of both ID and OOD classes. AUC is the standard evaluation protocol of novelty detection. We use the predicted probability of the $(K+1)$ -th class as the score when calculating AUC. We report the mean result with standard deviation over three runs of different random seeds.

4.2 Implementation details

For a fair comparison, we use the same backbone network with prior OSSL works for each task. For MNIST, we use the two-layer CNN in DS3L [11]. For CIFAR-10, CIFAR-100, and TinyImageNet, we use the WideResNet28-2 [33]. For IN-30, we use the ResNet-18. For hyperparameters, we set a relatively small size $N_m = 8 \times K$ for the OOD memory queue regarding the class number K . We set $K_m = 1$ as the enqueue number per iteration. The SSL hyperparameters settings follow FixMatch. The other hyperparameters and detailed training settings are reported in the supplementary material.

Table 1: Close-set classification accuracy (%) of different methods on 4 datasets.

Dataset			MNIST		CIFAR-10		CIFAR-100		TinyImageNet	
Mismatch ratio			0.3	0.6	0.3	0.6	0.3	0.6	0.3	0.6
SSL	PL [17]	ICML'13	90.0±0.7	86.0±0.6	75.8±0.8	74.6±0.7	60.2±0.3	57.5±0.6	36.6±0.6	35.8±0.4
	Pi-Model [26]	NeurIPS'16	92.4±0.6	86.6±0.5	75.7±0.7	74.5±1.0	59.4±0.3	57.9±0.3	36.9±0.4	36.4±0.5
	VAT [22]	TPAMI'19	94.5±0.3	90.4±0.3	76.9±0.6	75.0±0.5	61.8±0.4	59.6±0.6	36.7±0.5	36.3±0.6
	FixMatch [27]	NeurIPS'20	-	-	81.5±0.2	80.9±0.3	65.9±0.3	65.2±0.3	-	-
OSSL	DS3L [11]	ICML'20	96.8±0.3	94.5±0.4	78.1±0.4	76.9±0.5	-	-	-	-
	UASD [7]	AAAI'20	96.2±0.6	94.3±0.8	77.6±0.4	76.0±0.4	61.8±0.4	58.4±0.5	37.1±0.7	36.9±0.6
	MTCF [31]	ECCV'20	93.7±0.5	88.5±0.3	85.5±0.6	81.7±0.5	63.1±0.6	61.1±0.3	37.0±0.5	36.6±0.4
	CL [5]	AAAI'21	96.9±0.1	95.6±0.4	83.2±0.4	82.1±0.4	63.6±0.4	61.5±0.5	37.3±0.7	36.7±0.8
	OpenMatch [24]	NeurIPS'21	97.8±0.2	96.0±0.2	88.2±0.2	85.5±0.3	68.7±0.1	68.4±0.2	37.9±0.4	37.0±0.3
	T2T [14]	ICCV'21	98.4±0.1	96.2±0.2	89.0±0.4	86.9±0.2	69.8±0.2	68.0±0.2	39.1±0.3	37.3±0.3
	SAFE-STUDENT [12]	CVPR'22	98.3±0.3	96.5±0.1	85.7±0.3	83.8±0.1	68.4±0.2	68.2±0.1	37.7±0.3	37.1±0.3
	Ma <i>et al.</i> [20]	ICCV'23	98.7±0.2	96.9±0.1	91.4±0.3	89.1±0.1	72.5±0.2	70.4±0.1	40.8±0.3	39.9±0.3
SCOMatch	Ours	99.0±0.1	99.0±0.1	92.2±0.2	90.2±0.6	75.3±0.5	73.5±0.2	54.2±0.3	52.3±0.3	

Table 2: AUC for OOD class identification of different methods on MNIST.

Mismatch ratio			0.1	0.2	0.3	0.4	0.5	0.6	Avg
Probabilities [13]	ICLR'17		84.3±0.9	84.3±0.9	84.3±0.9	84.3±0.9	84.3±0.9	84.3±0.9	84.3
DS3L [11]	ICML'20		93.1±0.4	91.7±0.2	90.6±0.1	90.5±0.5	89.1±0.2	85.1±0.8	90.9
SAFE-STUDENT [12]	CVPR'22		97.3±0.2	96.5±0.1	96.0±0.9	94.6±0.9	93.5±0.3	91.4±0.2	95.3
Ma <i>et al.</i> [20]	ICCV'23		98.0±0.2	97.5±0.1	97.0±0.2	96.1±0.4	94.8±0.2	93.0±0.2	96.4
SCOMatch	Ours		98.0±0.2	98.3±0.2	98.0±0.1	98.3±0.2	98.0±0.1	98.2±0.2	98.1

4.3 Comparison with SOTA methods

Various mismatch ratios. We first prove the effectiveness of SCOMatch by comparing it with state-of-the-art SSL and OSSL methods with different mismatch ratios. Here, we use $\mu = 6$ to align the unlabeled batch size of our rivals [14]. Following Ma *et al.* [20], we report the close-set classification accuracy of ID classes in Table 1. SCOMatch outperforms rivals on all the datasets and mismatch ratios, especially on the more challenging CIFAR-100 and TinyImageNet. On the hardest dataset, TinyImageNet, we achieve 54.2% accuracy under the mismatch ratio of 0.3, which surpasses about 13.4% on the strongest rival [20]. The AUC results in Table 2 evaluate the ability of OOD identification on MNIST with different ratios.

Various numbers of labeled data and ID classes. We further prove the effectiveness of SCOMatch with different numbers of labeled data and ID classes. Here, we use $\mu = 4$ to align the unlabeled batch size with SSB [9]. Table 3 and Table 4 show the results of close-set classification accuracy and AUC, respectively. The results suggest that SCOMatch still outperforms its rivals.

Table 3: Close-set classification accuracy (%) of different methods.

Dataset		CIFAR-10		CIFAR-100				IN-30
No. of ID / OOD classes		6 / 4		55 / 45		80 / 20		20 / 10
No. of labeled samples		25	50	25	50	25	50	10%
FixMatch [27]	NeurIPS’20	91.9±0.2	91.3±0.2	69.9±0.0	73.3±0.6	63.6±0.4	67.1±0.1	87.1±0.4
MTCF [31]	ECCV’20	71.9±10.1	81.0±5.2	58.1±2.1	66.0±0.8	52.3±0.1	59.2±0.0	82.4±0.7
OpenMatch [24]	NeurIPS’21	54.9±2.3	91.3±1.2	67.1±1.4	71.9±1.1	52.1±4.8	66.9±0.2	89.6±1.0
T2T [14]	ICCV’21	83.2±1.0	90.6±0.1	65.7±0.9	70.7±0.1	47.6±10.4	64.2±0.6	88.9±0.1
SSB [9]	ICCV’23	91.7±0.2	92.2±0.3	70.6±0.4	73.7±0.8	64.2±0.4	68.0±0.2	-
IOMatch [19]	ICCV’23	93.0±0.1	93.3±0.1	69.2±0.3	72.6±0.3	63.7±0.5	67.6±0.1	89.2±0.3
SCOMatch	Ours	94.1±0.4	94.0±0.1	71.1±1.2	74.3±0.8	64.4±0.4	70.0±0.4	91.0±0.1

Table 4: AUC for OOD class identification of different methods.

Dataset		CIFAR-10		CIFAR-100				IN-30
No. of ID / OOD classes		6 / 4		55 / 45		80 / 20		20 / 10
No. of labeled samples		25	50	25	50	25	50	10%
FixMatch [27]	NeurIPS’20	37.4±0.8	39.4±0.2	54.5±1.0	55.8±1.0	41.4±0.1	44.3±0.8	88.6±0.5
MTCF [31]	ECCV’20	92.0±3.5	94.5±2.0	76.9±1.5	72.5±0.2	69.2±0.9	72.4±1.9	93.8±0.8
OpenMatch [24]	NeurIPS’21	62.5±4.2	99.4±0.2	84.9±0.1	87.0±0.2	74.9±3.8	86.2±0.5	96.4±0.7
T2T [14]	ICCV’21	34.9±27.5	23.9±8.5	53.0±6.2	59.5±1.5	50.5±9.2	61.4±21.1	84.5±0.1
SSB [9]	ICCV’23	99.4±0.4	99.6±0.2	89.4±0.4	90.6±0.5	90.3±1.3	85.3±2.1	-
IOMatch [19]	ICCV’23	53.5±0.1	60.6±1.0	70.2±0.3	71.9±0.4	63.2±0.2	63.9±1.7	89.4±0.5
SCOMatch	Ours	99.9±0.0	99.7±0.3	90.3±0.8	91.3±0.4	94.0±1.6	85.4±0.3	97.5±0.5

4.4 Ablation studies and discussions

Ablation study on the head structure. We conduct ablation studies to prove the effectiveness of using a single classification head: without ‘Single head’ in Table 5 (Column 2) means that we use two separate classification heads: one for K -class, supervised with labeled ID data and performing close-set self-training, the other for $(K+1)$ -class, supervised with both labeled ID data and OOD data from our queue, then perform open-set self-training. The results prove that two separate heads actually impact the performance. As we have discussed, the two heads conflict with each other as they attempt to cluster the embedded feature into different distributions. Moreover, only labeled ID data supervises the K -class head, causing SCOMatch to overtrust the labeled data.

Ablation study on SCO self-training. We also conduct ablation studies on each loss component in SCO self-training in Table 5. The results show that both close-set and open-set self-training contribute to the performance of SCOMatch (Columns 3 and 4). As we have discussed, open-set self-training is necessary for constructing the $(K+1)$ -class SSL task, thus improving the performance of

Table 5: Ablation studies of the head structure and the SCO self-training in SCOMatch. The model is trained on CIFAR-10 (6 ID classes) and CIFAR-100 (55 ID classes) with 25 labeled images per class. Column 1 indicates FixMatch.

Single head	SCO self-training		CIFAR-10			CIFAR-100		
	\mathcal{L}_u^{close}	\mathcal{L}_u^{open}	ACC_{close}	ACC_{open}	AUC	ACC_{close}	ACC_{open}	AUC
			91.7	55.0	37.1	69.9	38.4	53.5
	✓	✓	90.5	54.3	61.4	69.1	39.7	72.1
✓		✓	93.2	95.2	99.0	68.7	71.6	88.5
✓	✓		93.5	80.6	90.9	69.0	58.7	75.0
✓	✓	✓	94.4	95.4	99.3	73.3	74.4	90.6

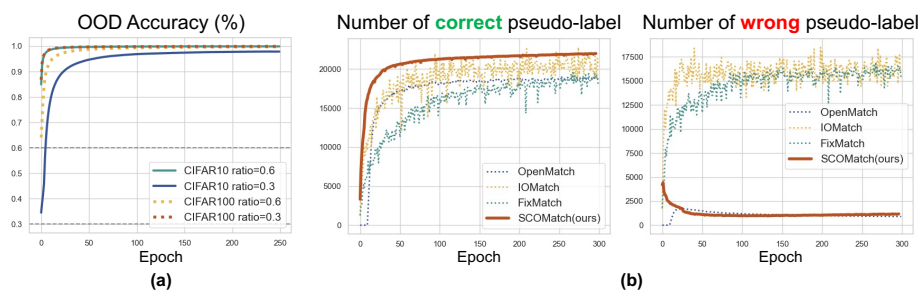


Fig. 3: (a): The quality of OOD samples in the OOD memory queue during training. The grey dashed line represents the actual ratio of OOD samples in unlabeled data. (b): Correct and wrong pseudo-label number of four methods during training on CIFAR-10 with 50 labeled images.

identifying OODs with unlabeled data. Meanwhile, close-set self-training can calibrate the distribution of \mathcal{C}^K and improve the quality of OOD sample selection. Thus serving the ultimate goal of the OSSSL task, *i.e.*, better classifying ID samples. Note that we cannot use the OOD memory queue without \mathcal{L}_s^{ood} and SCOMatch will fall back to FixMatch (Column 1).

Analysis of the OOD memory queue. We track the quality of the OOD samples in our queue during training. Figure 3(a) illustrates the average ratio of real OOD samples in the queue per epoch. The model is trained on CIFAR-10 and CIFAR100 with different mismatch ratios. It can be observed that the queue is noisy in the early stage, but the quality is rapidly improved with training. Note that we actually have 50 OOD classes in CIFAR-100, and the quality remains reliable. This verifies the effectiveness of the proposed OOD memory queue and corresponding update strategy.

Can SCOMatch alleviate overtrusting? We further prove that SCOMatch can alleviate the overtrusting problem by exploring the pseudo-label quality. Figure 3(b) shows the number of correct and wrong pseudo-labels during SSL.

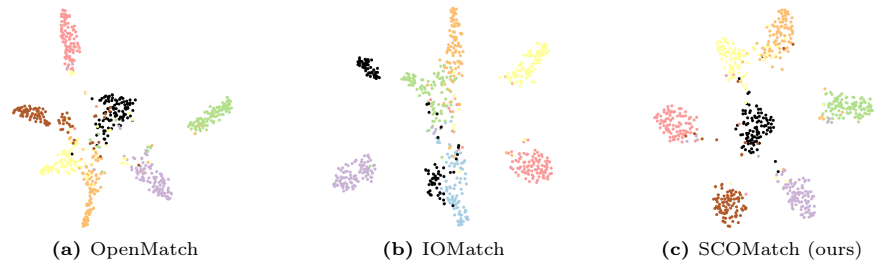


Fig. 4: t-SNE visualization results of randomly selected 100 samples from CIFAR-10 test data. Black dots denote the features of OOD samples. Other colors are ID samples.

Pseudo-labels are the samples from unlabeled data with confidence scores above the SSL threshold. The wrong pseudo-label means the class prediction differs from the annotation, while the correct pseudo-label means the prediction matches the annotation. The results show that SCOMatch generates more correct and less wrong pseudo-labels. Such high-quality pseudo-label leads to better performance. Meanwhile, FixMatch, OpenMatch, and IOMatch either generate a large number of wrong pseudo-labels or fewer correct ones. This phenomenon is consistent with the decision boundary and confusion matrix illustrated in Figure 1.

Analysis of the feature distribution. We apply t-SNE [21] visualization on the feature after the backbone for randomly selected 100 test samples in CIFAR-10. Figure 4 shows the result of OpenMatch, IOMatch, and SCOMatch. It shows that SCOMatch can form a better decision boundary between different classes, including the OOD class (black dots). The model is trained on CIFAR-10 with 50 labeled ID images and six ID classes as in Table 3.

Limitations and future direction. SCOMatch only considers the OOD samples of different class spaces within the same domain. The other open-set problems caused by domain difference (*e.g.*, OODs from cartoon drawings or synthetic images) are not considered and will be a potential direction.

5 Conclusion

In this paper, we identified the issue of overtrusting limited labeled ID data in prior OSSL methods. To alleviate this, we treat OOD as a new class with its own labeled data and form a new SSL process. Specifically, we proposed SCOMatch, a novel OSSL method that 1) selects reliable OOD samples as additional labeled data by our OOD memory queue and corresponding update strategy and 2) integrates the new SSL process with the original task with the proposed simultaneous close-set and open-set self-training. Experimental results on various OSSL benchmarks demonstrated that SCOMatch achieves better performance compared with state-of-the-art methods. We also conducted ablation studies to validate the effectiveness of each component of our method. With SCOMatch, more existing unlabeled data can be effectively leveraged to improve the model’s performance without the need for additional manual labeling and filtering.

Acknowledgements

Liuyu Xiang is supported by the National Natural Science Foundation of China (No. 62301066).

References

1. Bachman, P., Alsharif, O., Precup, D.: Learning with pseudo-ensembles. *Advances in Neural Information Processing Systems* **27** (2014)
2. Berthelot, D., Carlini, N., Cubuk, E.D., Kurakin, A., Sohn, K., Zhang, H., Raffel, C.: Remixmatch: Semi-supervised learning with distribution matching and augmentation anchoring. In: *International Conference on Learning Representations* (2020)
3. Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., Raffel, C.A.: Mixmatch: A holistic approach to semi-supervised learning. *Advances in Neural Information Processing Systems* **32** (2019)
4. Berthelot, D., Roelofs, R., Sohn, K., Carlini, N., Kurakin, A.: Adamatch: A unified approach to semi-supervised learning and domain adaptation. In: *International Conference on Learning Representations* (2021)
5. Cascante-Bonilla, P., Tan, F., Qi, Y., Ordonez, V.: Curriculum labeling: Revisiting pseudo-labeling for semi-supervised learning. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. vol. 35, pp. 6912–6920 (2021)
6. Chen, H.Y., Chien, J.T.: Deep semi-supervised learning for domain adaptation. In: *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing*. pp. 1–6 (2015)
7. Chen, Y., Zhu, X., Li, W., Gong, S.: Semi-supervised learning under class distribution mismatch. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. vol. 34, pp. 3569–3576 (2020)
8. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 248–255 (2009)
9. Fan, Y., Kulkova, A., Dai, D., Schiele, B.: Ssb: Simple but strong baseline for boosting performance of open-set semi-supervised learning. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 16068–16078 (2023)
10. Guo, L.Z., Li, Y.F.: Class-imbalanced semi-supervised learning with adaptive thresholding. In: *International Conference on Machine Learning*. pp. 8082–8094 (2022)
11. Guo, L.Z., Zhang, Z.Y., Jiang, Y., Li, Y.F., Zhou, Z.H.: Safe deep semi-supervised learning for unseen-class unlabeled data. In: *International Conference on Machine Learning*. pp. 3897–3906 (2020)
12. He, R., Han, Z., Lu, X., Yin, Y.: Safe-student for safe deep semi-supervised learning with unseen-class unlabeled data. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 14585–14594 (2022)
13. Hendrycks, D., Gimpel, K.: A baseline for detecting misclassified and out-of-distribution examples in neural networks. In: *International Conference on Learning Representations* (2017)
14. Huang, J., Fang, C., Chen, W., Chai, Z., Wei, X., Wei, P., Lin, L., Li, G.: Trash to treasure: Harvesting ood data with cross-modal matching for open-set semi-supervised learning. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 8310–8319 (2021)

15. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
16. LeCun, Y.: The mnist database of handwritten digits. <http://yann.lecun.com/exdb/mnist/> (1998)
17. Lee, D.H., et al.: Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In: Workshop on challenges in representation learning, International Conference on Machine Learning. vol. 3, p. 896 (2013)
18. Li, J., Xiong, C., Hoi, S.C.: Comatch: Semi-supervised learning with contrastive graph regularization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9475–9484 (2021)
19. Li, Z., Qi, L., Shi, Y., Gao, Y.: Iomatch: Simplifying open-set semi-supervised learning with joint inliers and outliers utilization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 15870–15879 (2023)
20. Ma, Q., Gao, J., Zhan, B., Guo, Y., Zhou, J., Wang, Y.: Rethinking safe semi-supervised learning: Transferring the open-set problem to a close-set one. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 16370–16379 (2023)
21. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. *Journal of Machine Learning Research* **9**(11) (2008)
22. Miyato, T., Maeda, S.i., Koyama, M., Ishii, S.: Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **41**(8), 1979–1993 (2018)
23. Rasmus, A., Berglund, M., Honkala, M., Valpola, H., Raiko, T.: Semi-supervised learning with ladder networks. *Advances in Neural Information Processing Systems* **28** (2015)
24. Saito, K., Kim, D., Saenko, K.: Openmatch: Open-set semi-supervised learning with open-set consistency regularization. *Advances in Neural Information Processing Systems* **34**, 25956–25967 (2021)
25. Saito, K., Saenko, K.: Ovanet: One-vs-all network for universal domain adaptation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9000–9009 (2021)
26. Sajjadi, M., Javanmardi, M., Tasdizen, T.: Regularization with stochastic transformations and perturbations for deep semi-supervised learning. *Advances in Neural Information Processing Systems* **29** (2016)
27. Sohn, K., Berthelot, D., Carlini, N., Zhang, Z., Zhang, H., Raffel, C.A., Cubuk, E.D., Kurakin, A., Li, C.L.: Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in Neural Information Processing Systems* **33**, 596–608 (2020)
28. Tarvainen, A., Valpola, H.: Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in Neural Information Processing Systems* **30** (2017)
29. Wang, Y., Chen, H., Heng, Q., Hou, W., Fan, Y., Wu, Z., Wang, J., Savvides, M., Shinozaki, T., Raj, B., Schiele, B., Xie, X.: Freematch: Self-adaptive thresholding for semi-supervised learning. In: International Conference on Learning Representations (2023)
30. Xie, Q., Dai, Z., Hovy, E., Luong, T., Le, Q.: Unsupervised data augmentation for consistency training. *Advances in Neural Information Processing Systems* **33**, 6256–6268 (2020)
31. Yu, Q., Ikami, D., Irie, G., Aizawa, K.: Multi-task curriculum framework for open-set semi-supervised learning. In: European Conference on Computer Vision (2020)

32. Yu, Z., Li, Y., Lee, Y.J.: Inpl: Pseudo-labeling the inliers first for imbalanced semi-supervised learning. In: International Conference on Learning Representations (2022)
33. Zagoruyko, S., Komodakis, N.: Wide residual networks. In: British Machine Vision Conference (2016)
34. Zhang, B., Wang, Y., Hou, W., Wu, H., Wang, J., Okumura, M., Shinozaki, T.: Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. *Advances in Neural Information Processing Systems* **34**, 18408–18419 (2021)
35. Zheng, M., You, S., Huang, L., Wang, F., Qian, C., Xu, C.: Simmatch: Semi-supervised learning with similarity matching. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14471–14481 (2022)
36. Zhou, Z.H., Zhou, Z.H.: Semi-supervised learning. *Machine Learning* pp. 315–341 (2021)