MOOD: Leveraging Out-of-Distribution Data to Enhance Imbalanced Semi-Supervised Learning

Yang Lu, *Senior Member, IEEE*, Xiaolin Huang, Yizhou Chen, Mengke Li, *Member, IEEE*, Yan Yan, *Senior Member, IEEE*, Chen Gong, *Senior Member, IEEE*, and Hanzi Wang, *Senior Member, IEEE*

***Abstract*—The imbalanced semi-supervised learning (SSL) has emerged as a critical research area due to the prevalence of class imbalanced and partially labeled data in real-world scenarios. As the requirement for data volume increases, naturally collected datasets inevitably contain out-of-distribution (OOD) samples. However, the performance of existing imbalanced SSL methods experiences a marked deterioration with OOD data. In this article, we propose an imbalanced SSL method called mixup-OOD (MOOD) to address this issue. The core idea is to "turn waste into treasure," exploring the potential of leveraging seemingly detrimental OOD data to expand the feature space, particularly for tail classes. Specifically, we first filter OOD data from unlabeled data, and then fuse it with labeled data to boost feature diversity for the tail classes. To avoid feature overlapping with OOD data, we develop a push-and-pull (PaP) loss to attract in-distribution (ID) instances toward respective class centroids while repelling OOD samples from them. Extensive experiments show that MOOD achieves superior performance compared with other state-of-the-art methods and exhibits robustness across data with different imbalanced ratios and OOD proportions. The source code is available at: https://github.com/xlhuang132/MOODv2**

***Index Terms*—Feature learning, image classification, long-tailed learning, semi-supervised learning (SSL).**

I. Introduction

DEEP neural networks have been widely used in various learning tasks due to their outstanding performance. However, achieving excellent performance often requires a large amount of labeled and class-balanced data, which can be expensive in real applications. Given limited labeling budgets, the emergence of semi-supervised learning (SSL) provides feasible solutions to improve model generalization on limited labeled data with abundant unlabeled data. Recently, representative SSL methods such as pseudolabeling [1], [2], [3], [4] and consistency regularization [5], [6], [7], [8], [9] have been proposed to enable the model to make full use of the information in unlabeled data which can significantly improve model generalization.

In the typical setting of SSL, it is often assumed that the data are class balanced, meaning that each class has an equal number of samples for both labeled and unlabeled data. However, in naturally collected data, the class distribution may potentially tend to be long-tailed [10]. One particular distribution of class imbalance is the long-tailed distribution [11]. The classes possessing large amounts of data are called "head" classes, while those with only a few data are referred to as "tail" classes. When labeled data are long-tailed, the performance of general SSL methods drops severely because unlabeled data are also likely to be long-tailed. The imbalanced unlabeled data mainly helps improving the head classes, while aggravating the performance degradation of the tail classes. With increasing attention to applying algorithms to real-world scenarios, obtaining an imbalanced-robust SSL model has been extensively studied [12], [13], [14], [15].

A potential solution to alleviate the imbalanced SSL problem is to collect more unlabeled data to balance the overall data distribution. However, naturally collected datasets inevitably contain out-of-distribution (OOD) samples. In some cases, unlabeled data are collected separately from the labeled data, either at different times or places. It may bring OOD data into unlabeled data because the data distribution may change. In this case, OOD data, which does not belong to any class of labeled data, is mixed with in-distribution (ID) data in unlabeled data. For example, for object detection in autonomous driving the samples collected in a new scene may involve undefined classes [16]. Consequently, SSL methods will treat OOD samples in the same way as ID samples, equivalent to introduce noisy samples during training. To address the problem of OOD data, several open-set SSL methods have been proposed [17], [18], [19], [20]. The common strategy is to discard filtered OOD samples or reduce their influence during SSL model training [18], [21]. The consensus behind these methods is that OOD samples are definitely harmful to model training. However, such a way ignores the potential value of OOD samples. Although the label distribution of OOD data

Received 9 April 2024; revised 15 October 2024 and 13 May 2025; accepted 20 May 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62376233, Grant 62431004, Grant U21A20514, Grant 62372388, Grant 62336003, and Grant 12371510; in part by the Natural Science Foundation of Fujian Province under Grant 2024J09001; in part by Guangdong Basic and Applied Basic Research Foundation under Grant 2024A1515010163; in part by the National Key Laboratory of Radar Signal Processing under Grant JKW202403; and in part by the Xiaomi Young Talents Program. *(Corresponding author: Hanzi Wang.)*

Yang Lu, Xiaolin Huang, Yizhou Chen, Yan Yan, and Hanzi Wang are with the Key Laboratory of Multimedia Trusted Perception and Efficient Computing, Ministry of Education of China, Xiamen University, Xiamen 361005, China (e-mail: luyang@xmu.edu.cn; xlhuang@stu.xmu.edu.cn; 36920231153184@stu.xmu.edu.cn; yanyan@xmu.edu.cn; hanzi.wang@xmu.edu.cn).

Mengke Li is with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China (e-mail: csmengkeli@gmail.com).

Chen Gong is with the School of Automation and Intelligent Sensing, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: chen.gong@sjtu.edu.cn).

Digital Object Identifier 10.1109/TNNLS.2025.3573963

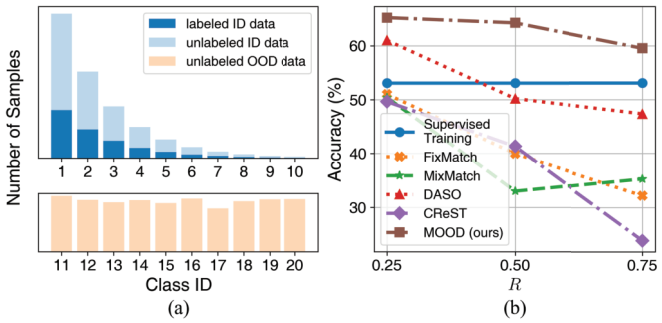


Fig. 1. (a) Example of long-tailed distribution with OOD data. (b) Comparison of accuracy for the tail classes among several SSL methods on CIFAR-10-LT (IF = 100) with different proportions of OOD data in unlabeled data (R).

is out-of-target distribution, the features extracted from them contain potential benefits for model training.

This article therefore studies the problem of imbalanced SSL with OOD data, whose application is more general and realistic. Specifically, we assume that labeled data are long-tailed, where only a few classes contain a majority of samples, while a vast number of classes have only a small proportion of samples. The unlabeled data consist of both ID and OOD data. The unlabeled ID data are also long-tailed, similar to the labeled data, while the OOD data do not belong to any ID class. An example of the distribution of our problem is illustrated in Fig. 1(a). We also conducted a preliminary experiment to demonstrate that the performance of existing SSL methods significantly deteriorates when confronted with the OOD data. In Fig. 1(b), we show how the accuracy of the tail classes of the long-tailed CIFAR-10 declines with a different value R, which represents the proportion of OOD data in the unlabeled data. It can be observed that the accuracy of the tail classes for existing SSL methods drops notably as R increases, and they even perform worse than the supervised method that only uses a limited number of labeled data. This finding confirms the fact that the existence of OOD data in unlabeled data worsens the generalization performance of the tail classes.

To address this problem, we propose a novel SSL method, mixup-OOD (MOOD), which innovatively harnesses the untapped potential of OOD data in unlabeled data to improve representation learning for imbalanced data. Instead of discarding the seemingly useless or even detrimental OOD data as other SSL methods do, MOOD turns waste to treasure and maximizes the utilization of all OOD data. T2T [22] and TOOR [23] share a similar idea with our MOOD. T2T incorporates OOD samples for backbone training, while TOOR aims to identify recyclable samples within OOD samples to enhance model training. Different from these methods, MOOD specifically contributes to classifier training, differentiating itself from T2T. Moreover, compared with TOOR, MOOD can also effectively utilize OOD data that has been identified as nonrecyclable by TOOR. MOOD filters OOD data out of unlabeled data using an OOD filter and then fuses them with labeled data through mixup. The fused data help enhancing the feature space of its corresponding class. To further expand the feature space, we propose a specific loss function, push-and-pull (PaP) loss, which pushes ID data toward its class center and pulls OOD data away from ID data. This strategy diversifies the features of OOD data, leading to expanded feature spaces for fused samples. With the synergy of OOD mixup and PaP loss, the feature space of each class achieves a more balanced state such that the generalization ability of the model, particularly for the tail classes, is significantly improved.

Our main contributions can be summarized as follows.

1) We propose MOOD, a simple but effective imbalanced SSL method, to address a more realistic SSL problem, where the distribution is imbalanced and unlabeled data contain OOD data.
2) We explore the approach of leveraging OOD data to enhance learning, rather than simply discarding it. It is accomplished through the fusion of labeled and filtered OOD data and the implementation of a specifically designed PaP loss function.
3) We achieve state-of-the-art performance on multiple SSL benchmarks by significantly improving the performance of the tail classes without compromising that of the head classes.

II. Related Works

A. General SSL Methods

The general SSL methods can be broadly divided into three categories: consistency regularity, pseudolabeling, and hybrid methods that combine both. Consistency regularization [5], [7], [9] encourages the model to output consistent results for the same input data with different forms of augmentation [24], [25]. Pseudolabeling [1], [26], [27] adopts a self-training strategy, using the model prediction to assign pseudolabel unlabeled data for the following supervised training. Hybrid methods combine both consistency regularization and pseudolabeling [28], [29], [30], [31], with FixMatch [28], MixMatch [29], and ReMixMatch [30] being notable examples. FixMatch requires the prediction of the strong view of the data to be consistent with the pseudolabel obtained through the weak view. MixMatch interpolates labeled data as well as unlabeled data with pseudolabels to improve the generalization performance. ReMixMatch further improves MixMatch by aligning the pseudolabels of unlabeled data with the label distribution of supervised data, thereby improving the consistency between the two. Building on the principles of consistency regularization and hybrid methods, SimMatch [32] further enhances SSL by introducing a similarity-matching mechanism. This approach ensures that similar samples maintain consistent representations during training, thereby extending the effectiveness of consistency regularization by emphasizing the relational structure among unlabeled data points.

It is worth noting that both of these methods assume a balanced target data distribution and do not account for OOD samples in the unlabeled data. In more realistic scenarios, data often has two characteristics: 1) ID data follows a natural long-tailed distribution and 2) the unlabeled data may contain OOD data. These characteristics make existing methods vulnerable to distribution shifts and noisy data, thereby reducing perfor-

mance. Therefore, further research and development of new SSL methods are needed to address the challenges in real-world scenarios.

B. Imbalanced SSL Methods

Imbalanced SSL focuses on improving the model performance on tail classes by utilizing unlabeled data to address the first data characteristic. Some methods aim to construct balanced classifiers or use auxiliary classifiers to mitigate the adverse effects of data imbalance on the model. For instance, DASO [14] is a general framework that can be easily integrated with other SSL learning models, which reduces biased predictions by mixing semantic and linear pseudolabels and introducing a novel semantic alignment loss. CoSSL [33] is a joint learning framework for imbalanced SSL, which decouples representation learning and classifier learning, and tightly couples them through pseudolabel generation. The representation module provides a momentum encoder for feature extraction in the other two modules, the classifier module uses the novel tail-class feature enhancement (TFE) to generate a balanced classifier, and the pseudolabel module uses the momentum encoder and a balanced classifier generates pseudolabels representing modules. ABC [34] can utilize the representation layer of the existing SSL algorithm to learn high-quality representations, and introduce a single-layer auxiliary balanced classifier to alleviate the class imbalance problem. Other methods focus on selecting unlabeled data, such as CReST [13], which improves the performance of existing SSL methods on tail classes by generating high-precision pseudolabels, and incorporates progressive distribution alignment for adaptive rebalancing, SaR [12], which is a pseudolabel generation method for adaptive refinement of soft labels, using the generated soft pseudolabels as pseudolabels to produce smaller deviations, thereby obtaining higher quality data for training classifiers. Recently, CDMAD [35] introduced a debiasing mechanism that adjusts for class bias during training and testing, allowing better handling of class distribution mismatches. By measuring and correcting the classifier's bias, CDMAD effectively rebalances the model without additional complexity, enhancing its performance in mismatched scenarios.

However, these imbalanced SSL methods generally assume that unlabeled data only contains known classes, without considering the possible existence of OOD data. This assumption can result in inaccurate pseudolabels when OOD samples are present, leading to a degradation in model performance. Such methods may misclassify OOD samples, confusing the model and ultimately reducing its effectiveness, particularly for tail classes.

C. Open-Set SSL Methods

Open-set SSL methods are designed to handle OOD samples for more robust model performance to address the second data characteristic. To reduce the adverse effects of OOD samples on the model, some methods aim to identify and discard OOD samples, while others use the soft labels of the OOD samples during model training. For example, OpenMatch [18] is enhanced based on FixMatch. It employs the OVA classifier, utilizes the confidence score of the sample as an inner layer, and introduces soft consistency regularization, thereby significantly enhancing outlier detection. IOMatch [36], on the other hand, simplifies this process by jointly utilizing both inliers and outliers without the need for a separate outlier detection stage. By treating outliers as a new class and using a multibinary classifier along with a closed-set classifier, IOMatch achieves better performance by leveraging the full set of unlabeled data, avoiding the risk of incorrectly excluding valuable inliers. MTCF [21] treats the open-set semi-supervised problem as multitask learning, including binary classification and general SSL tasks. Meanwhile, some open-set SSL methods selectively utilize the unlabeled samples to achieve results that are at least as good as those achieved using only labeled data. For example, DS3L [17] selectively uses unlabeled data in a metalearning manner for training to ensure that model performance will not be damaged by unlabeled data. AuxMix [37] employs self-supervised learning generalization features to mask unlabeled data that is semantically dissimilar to labeled data and regularizes learning by maximizing the prediction entropy of different auxiliary samples. ACR [38] can effectively utilize unlabeled data of unknown class distributions by introducing the adaptive consistency regularizer to enhance performance. In addition, DAC [39] constructs multiple prediction heads with different biases toward the unlabeled distribution within a single-training process, to detect and suppress OOD samples.

These methods implicitly assume that OOD samples are detrimental to model training and try to eliminate their negative influence without considering the potential value of OOD samples, somehow leading to data waste, as it overlooks the possibility that OOD samples could provide valuable information for improving model robustness and generalization.

III. Proposed Method

This article studies SSL from a more realistic perspective. We consider two factors in real applications that severely degrade the performance of SSL methods. The first factor is the natural long-tailed distribution of real-world data, which can bias SSL models toward the head classes and squeeze the tail classes space, as shown in Fig. 2(a). The second factor is the presence of OOD samples in unlabeled data. Under this circumstance, the OOD samples with the pseudolabels assigned by the SSL methods will introduce noise and further damage the model performance [40]. In a nutshell, OOD samples will exacerbate the original long-tailed problem. Existing SSL methods only address one of these factors. The imbalanced SSL methods are affected by OOD samples, and open-set SSL methods are unable to handle data with natural long-tailed distributions.

To alleviate the problem that OOD samples will exacerbate the original long-tailed problem, we propose MOOD to enhance the tail classes by leveraging the seemingly detrimental OOD samples from a unique perspective. The core idea of MOOD is shown in Fig. 2. In detail, MOOD consists of two parts: OOD mixup (see Section III-B) and feature space

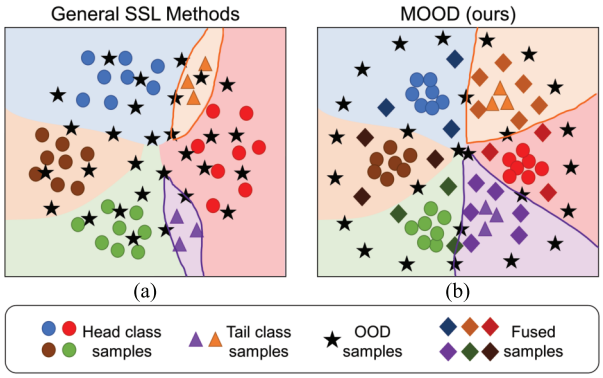


Fig. 2. Schematic to show the advantage of MOOD of leveraging the OOD data. The feature spaces for the tail classes are (a) squeezed by general SSL methods, but (b) expanded by MOOD with samples fused with OOD data.

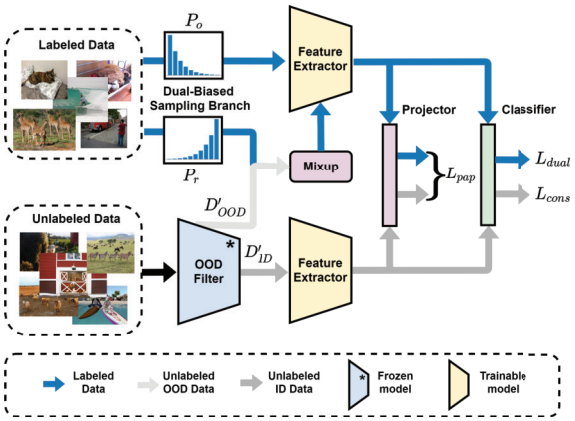


Fig. 3. Overview of the proposed MOOD. The blue and gray arrows represent the loss related to the labeled and unlabeled data, respectively.

expansion (see Section III-C). The OOD mixup increases feature diversity by fusing OOD information with ID data based on the class frequency, where the tail classes are fused with a higher probability. This approach ensures that the tail classes receive significant attention during model training. The feature space expansion utilizes a specifically designed PaP loss, denoted by L_{pap}, to encourage the model to obtain an unsqueezed feature space. By applying L_{pap}, OOD data are pushed away from the class center, allowing the fused samples to contribute significantly to feature space expansion, as shown in Fig. 2(b). The overall framework of MOOD is shown in Fig. 3, and the complete algorithm flowchart can be found in Section III-E.

A. Problem Setting and Notations

In the setting of SSL, we have labeled data $\mathcal{D}_L$ and unlabeled data $\mathcal{D}_U$. Each sample x_i^l in the labeled data $\mathcal{D}_L = \{(x_i^l, y_i^l)\}_{i=1}^N$ is associated with a label $y_i^l \in \{1, \ldots, C\}$. C is the total number of known classes. $N = \sum_{i=1}^C N_i$ is the total number of labeled data and N_i is the number of samples in class i. We assume that the classes are imbalanced and sorted in nonascending order, i.e., $N_i \geq N_j$ for $i < j$. The degree of class imbalance of a dataset is measured by the imbalance factor defined as $\text{IF} = N_1/N_C$. Unlabeled data $\mathcal{D}_U = \{(x_i^u)\}_{i=1}^M$ contains M samples without annotation. We assume that the unlabeled dataset consists of both the ID dataset $\mathcal{D}_I$ and the OOD dataset $\mathcal{D}_O$, i.e., $\mathcal{D}_U = \mathcal{D}_I \cup \mathcal{D}_O$. The samples in $\mathcal{D}_I$ follow the same class distribution as $\mathcal{D}_L$, and therefore, IF is the same for both $\mathcal{D}_L$ and $\mathcal{D}_I$. Samples in $\mathcal{D}_O$ belong to classes other than the known C classes. We aim to learn a model to effectively learn $\mathcal{D}_L$ and $\mathcal{D}_U$ to generalize well under a class-balanced test criterion. We use a typical deep neural network model f, e.g., ResNet [41] or wide ResNet [42], including a feature extractor θ and a classifier ϕ. Given a sample x, the model predicts it as $f(x) = \phi(\theta(x))$.

B. Ood Mixup

Inspired by the commonly adopted strategy for feature space expansion for long-tailed learning [43], [44], [45], we generate new training samples by OOD mixup. This augmentation operation is fundamental and indispensable in MOOD. We first detect OOD data from unlabeled data by an OOD filter which is trained on all labeled data and unlabeled data using instance contrastive learning [46]. In contrast to previous approaches, such as open-set SSL methods [17], [21], which merely discard OOD samples, MOOD leverages the information from detected OOD data. Specifically, MOOD fuses OOD data into the labeled data to expand the feature space. Specifically, MOOD leverages the information of the detected OOD data by fusing them with labeled data to expand the feature space, instead of simply discarding OOD samples like most open-set SSL methods. The filtered OOD data $\mathcal{D}'_{\text{OOD}}$ may not be identical to the ground-truth OOD set $\mathcal{D}_{\text{OOD}}$ due to false positives and false negatives during OOD filtering. Then, a labeled sample $(x^l, y^l) \in \mathcal{D}_L$ is randomly fused with an OOD sample $x^o \in \mathcal{D}'_{\text{OOD}}$ to obtain a fused sample $\tilde{x}$

$$\tilde{x} = \lambda x^l + (1 - \lambda)\, x^o \tag{1}$$

where λ controls the fusion ratio of two samples. It should be noted that the mixup approach used in MOOD differs from the typical mixup methods [47] in two ways. First, we set $\lambda = \max(\lambda', 1 - \lambda')$ to prevent the labeled sample information from being dominated by OOD sample information in the fused sample, where λ' is drawn from a beta distribution with parameter α. Second, we use y^l directly as the label of the fused sample $\tilde{x}$. The leveraging OOD data for sample generation has the advantage of improving feature diversity while avoiding semantic confusion among known classes.

OOD mixup is applied to all the known classes, not just the tail classes. Therefore, to prevent the model from being biased toward the head classes, we incorporate OOD mixup into a dual-biased sampling branch, which jointly samples data from reversed class probability distribution P_r and the original class probability distribution P_o. Specifically, the reversed class probability is obtained by $P_r = (1/Z)[p_1, p_2, \ldots, p_C]^{\text{T}}$, where $p_i = N_1/N_i$ and Z are the normalization term. To enhance the generalization ability of the tail classes, we only apply OOD mixup to data sampled from P_r, thus generating more fused samples for tail classes. Consequently, the sampled data in the dual-biased sampling branch for model training is class-balanced with increased sample diversity. The dual-biased sampling branch is similar to the bilateral branch in

BBN [48], with the key difference being that the samples from P_r are fused with the filtered OOD samples.

After sampling from the dual-biased sampling branch and conducting OOD mixup, the samples are fed into the shared feature extractor θ with a classifier ϕ. The loss function is calculated by the sum of losses for both branches

$$L_{\text{dual}} = \mathop{\mathbb{E}}_{(x,y)\sim P_o}\left[l\left(f\left(x\right),y\right)\right] + \mathop{\mathbb{E}}_{(\tilde{x},y)\sim P_r}\left[l\left(f\left(\tilde{x}\right),y\right)\right] \tag{2}$$

where $l(\cdot,\cdot)$ is the common cross-entropy loss, $f(x)$ is the prediction of model to data x, and y is its ground truth.

Regarding leveraging OOD samples, we take two key issues into account.

1) How Are Ood Samples Filtered From $\mathcal{D}_U$?: One key step in OOD mixup is to obtain the filtered OOD set $\mathcal{D}'_{\text{OOD}}$. To avoid the influence of data imbalance in the label space, we adopt the nearest neighbor-based OOD detection approach [49]. First, we obtain a pretrained model using the unsupervised contrastive learning method with InfoNCE loss [50] on all available data $\mathcal{D} = \mathcal{D}_L \bigcup \mathcal{D}_U$. The training loss is formulated as follows:

$$L_{\text{unsup}} = -\left(\frac{1}{|\mathcal{D}|}\sum_{i=1}^{|\mathcal{D}|}\log\frac{\exp\left(\mathbf{z}_i\cdot\mathbf{z}'/T_1\right)}{\sum_{j=1,j\neq i}^{|\mathcal{D}|}\exp\left(\mathbf{z}_i\cdot\mathbf{z}_j/T_1\right)}\right) \tag{3}$$

where z' belongs to the positive pair set of the ith sample. Then, we compute the distance to the kth nearest neighbor in the low-dimensional embedding space between unlabeled and labeled data and use a threshold-based criterion to distinguish ID and OOD data in the unlabeled data. To do this, we use the normalized low-dimensional embedding $\mathbf{z} = z(x)/\|z(x)\|_2$ for OOD filtering. Let $\mathcal{Z}_L = \{\mathbf{z}_i\}_{i=1}^{N}$ be the normalized embedding set of labeled data. For an unlabeled data point x_u with embedding $\mathbf{z}_u$, we calculate the Euclidean distances $\|\mathbf{z}_u - \mathbf{z}_i\|_2$, where $\mathbf{z}_i \in \mathcal{Z}_L$. We identify x_u as ID sample if $\|\mathbf{z}_u - \mathbf{z}_{(k)}\|_2 \leq \eta$, and as an OOD sample otherwise. Here, $\mathbf{z}_{(k)}$ is the embedding of its kth nearest neighbor, and η is the distance threshold chosen as in [49]. Finally, we obtain the filtered ID dataset $\mathcal{D}'_{\text{ID}}$ and filtered OOD dataset $\mathcal{D}'_{\text{OOD}}$. The complete algorithm of the OOD filter is shown in Section III-E. The filtered OOD samples are leveraged to be fused with labeled data, and the filtered ID samples are utilized in other loss functions (as described in Sections III-C and III-D). More results on the accuracy of the OOD filter are presented in Section IV-E. The backbone architecture of the OOD filter is based on ResNet-34, which is the same architecture used by the standard SSL model.

2) Why Are Labeled Data Fused With Filtered Ood Data Instead of All Unlabeled Data?: The unlabeled data contain both ID and OOD data, and the ID data share the same semantics as labeled data. Directly fusing unlabeled data with data sampled from P_r can expand the feature space of tail classes, but it can seriously harm the performance of other classes, particularly the head classes. Instead, leveraging OOD data for data fusion can expand the feature spaces of the tail classes without affecting other classes or causing semantic confusion. In Section IV-D, we conduct ablation experiments to compare the numerical results between fusing with all unlabeled data and fusing with only the filtered OOD data, which supports our hypothesis. The idea of utilizing OOD data for feature space enhancement is also discussed in [51] and [52]. OpenMix [52] integrates the labeled data from known categories with pseudolabeled data from unknown categories to establish learning relationships, facilitating the identification of novel classes in open-world scenarios. The key difference between MOOD and OpenMix lies in their treatment of OOD data. MOOD does not focus on the specific classification information of OOD data but rather on its contribution to the feature space of tail classes. In contrast, OpenMix aims to identify new classes and incorporate them through the label fusion to assist in model training. However, the OOD mixup approach differs from these methods in that it fuses OOD data with labeled data to particularly help tail classes for imbalanced SSL.

C. Feature Space Expansion

Fusing with OOD data increases sample diversity, which helps expand the feature space for tail classes without causing semantic confusion among classes. However, during model training, if the features of the OOD data gradually overlap with labeled data, the features of the fused data for tail classes may still be compressed, which undermines the goal of expanding the feature space for these classes. To address this issue, we propose a feature space expansion mechanism to further expand the feature space of the tail classes. Specifically, we encourage the filtered OOD data to be far away from all class centers, allowing the fused data to be distributed around the labeled data to achieve the feature space expansion. Conversely, we encourage the filtered ID data and labeled data to be close to their corresponding class centers. To implement the feature space expansion, we propose a loss function called PaP loss, formulated as follows:

$$L_{\text{pap}} = -\left(\frac{1}{|\mathcal{D}'_L|}\sum_{i=1}^{|\mathcal{D}'_L|}\log\frac{\exp\left(\mathbf{v}_i\cdot\boldsymbol{\mu}_{\tilde{y}_i}/T_1\right)}{\sum_{j=1,j\neq\tilde{y}_i}^{C}\exp\left(\mathbf{v}_i\cdot\boldsymbol{\mu}_j/T_1\right)} + \frac{1}{|\mathcal{D}'_O|}\sum_{i=1}^{|\mathcal{D}'_O|}\log\frac{\exp\left(\mathbf{v}_i\cdot\hat{\mathbf{v}}_i/T_2\right)}{\sum_{j=1}^{C}\exp\left(\mathbf{v}_i\cdot\boldsymbol{\mu}_j/T_2\right)}\right) \tag{4}$$

where $\mathcal{D}'_L = \mathcal{D}_L \bigcup \mathcal{D}'_I$ contains all labeled data and filtered ID data. The corresponding label $\tilde{y}_i$ is used to denote the ground-truth label y_i for labeled data or the pseudolabel $\hat{y}_i$ assigned for filtered ID data. The dot represents the inner product of vectors. T_1 and T_2 are the temperature scaling factors. $\mathbf{v}_i = g(\theta(x_i))$ is used to represent the low-dimensional feature for the ith sample projected by a projector g, a two-layer neural network. $\hat{\mathbf{v}}_i$ is the feature of the strong augmented version of the ith filtered OOD sample. We use $\boldsymbol{\mu}_j$ to represent the class center of the jth class. To obtain a stable class center, we follow [14] to update $\boldsymbol{\mu}_j$ by calculating the mean feature of the related memory bank.

The two terms in PaP loss address two perspectives to achieve the goal of better feature space expansion. The first term pulls all filtered ID data and labeled data closer to their corresponding class center. The second term pushes the filtered OOD data away from all class centers. Meanwhile,

we encourage the distance between the features of different augmented versions of each OOD sample to become smaller. When OOD samples are pushed away from the class centers by L_{pap}, the feature space of the tail classes can be further expanded through the OOD mixup mechanism. OOD mixup and PaP loss work together and form the core of MOOD.

D. Overall Training Objective

We adopt the cross-entropy loss to utilize supervised information and the proposed PaP loss to constrain the optimization of the feature extractor. The filtered ID data in unlabeled data are used to impose consistent regular constraints on the model similar to FixMatch [28]. Specifically, we first generate a corresponding pseudolabel for each filtered unlabeled ID data. To obtain pseudolabel, we compute the predicted class distribution of the weakly augmented version of the unlabeled data point x_i: $q_i = f(a(x_i))$, and use $\hat{y}_i = \arg\max(q_i)$ as its pseudolabel. Then, we train the model to produce predicted class distribution $\tilde{q} = f(A(x_i))$ of its strongly augmented version and constrain it to be consistent with the pseudolabels $\hat{y}$ by

$$L_{\text{cons}} = \frac{1}{|\mathcal{D}_l'|} \sum_{i=1}^{|\mathcal{D}_l'|} \mathbb{1}\left(\max\left(q_i\right) > \tau\right) H\left(\hat{y}_i, f\left(A\left(x_i\right)\right)\right) \tag{5}$$

where τ is a hyperparameter denoting the threshold and $\mathbb{1}$ is the indicator function. H is simply the cross-entropy loss as same as FixMatch. It measures the difference between the pseudolabel generated from the weakly augmented input and the prediction from the strongly augmented input. The overall training objective function consists of three terms

$$L_{\text{total}} = L_{\text{dual}} + \lambda_{\text{cons}} L_{\text{cons}} + \lambda_{\text{pap}} L_{\text{pap}} \tag{6}$$

where λ_{cons} and λ_{pap} are both hyperparameters, indicating the weight of each loss item.

E. Algorithm Flowchart

The complete pseudocode of MOOD is shown in Algorithm 1. The OOD data detected by an OOD filter are fused with labeled data by mixup, and then fed into the dual-biased sampling branch for joint training with three loss functions. The blue and gray arrows represent the loss related to the labeled and unlabeled data, respectively. The complete training pseudocode for the OOD filter is shown in Algorithm 2. The OOD filter's backbone architecture is based on ResNet-34, which has been trained using a self-supervised learning approach. The OOD samples are detected based on a threshold value.

Algorithm 1 Proposed MOOD Algorithm

Input: Labeled data $\mathcal{D}_L$, unlabeled data $\mathcal{D}_U$, original class probability distribution P_o, reversed class probability distribution P_r, feature extractor θ, classifier ϕ, projector g, total epochs E, I iterations per epoch and learning rate r.
Output: Feature extractor θ, classifier ϕ.

```
Initialize feature extractor θ, classifier φ and projector g;
D'_ID, D'_OOD ← detect_ood(D_L, D_U);
for each epoch t = 1, 2, ..., E do
    for each iteration i = 1, 2, ..., I do
        x̃ ← mixup(x^i_OOD, (x, y) ~ P_r);
        x ← concat((x, y) ~ P_o, x̃);
        compute L_total by Eq. (6);
        update θ, φ, g by L_total and r;
    end for
end for
```

Algorithm 2 Algorithm of Detecting OOD Data

Input: Labeled data $\mathcal{D}_L$, unlabeled data $\mathcal{D}_U$, feature extractor θ, projector g, the k-th nearest neighbor, distance percentile threshold τ, total epochs E, I iterations per epoch and learning rate r.
Output: Detected OOD data $\mathcal{D}'_{\text{OOD}}$ and ID data $\mathcal{D}'_{\text{ID}}$.

```
Initialize feature extractor θ and projector g;
for each epoch t = 1, 2, ..., E do
    for each iteration i = 1, 2, ..., I do
        (x) ~ D_L ∪ D_U;
        x', x'' ← transform(x);
        compute L_unsup by Eq. (3);
        update θ, g by L_unsup and r;
    end for
end for
Z_l, Z_U ← extract(θ, g);
Dist_l, Dist_u ← distance(Z_l, Z_u, k);
Dist'_l ← sort(Dist_l);
obtain distance threshold η by τ and Dist'_l;
D'_OOD, D'_ID ← compare(Dist_u, η);
```

IV. Experiments

In Section IV-A, we introduce the experimental datasets, comparison methods, and training details. In Section IV-B, we conduct experiments to assess the efficacy of MOOD on multiple benchmark datasets and compare it with relevant state-of-the-art methods. In Section IV-C, we demonstrate the exceptional performance of MOOD in learning tail class features by visualizing the features extracted from the backbone. In Section IV-D, we conduct detailed experiments on the contribution of each component of MOOD. In Section IV-E, we evaluate the performance of OOD filters, confirming that using the OOD filter directly will lead to the information loss, while MOOD maximizes the utilization of unlabeled data. In Section IV-F, we investigate the influence of different backbones on MOOD.

A. Experimental Settings

1) Comparison Methods: We compare the proposed MOOD with several related methods, including: 1) a supervised method using the cross-entropy function as the loss; 2) general SSL methods FixMatch [28] and MixMatch [29]; 3) imbalanced SSL methods CReST [13] and DASO [14]; and 4) open-set SSL methods MTCF [21] and OpenMatch [18].

2) Imbalanced Datasets With Ood Data: We conduct SSL experiments by configuring both the labeled data and the unlabeled ID data to follow the same long-tailed distribution,

TABLE I

COMPARISON OF AVERAGE ACCURACY (%) WITH THE STANDARD SSL METHODS AND THE IMBALANCED SSL METHODS ON CIFAR-10-LT AND CIFAR-100-LT WITH A DIFFERENT IF. BOLD VALUES ARE THE BEST AND UNDER-LINED VALUES COME NEXT. THE SAME CONVENTION APPLIES TO THE SUBSEQUENT TABLES

		Average Accuracy (%)						Tail Accuracy (%)					
	R	0.25	0.50	0.75	0.25	0.50	0.75	0.25	0.50	0.75	0.25	0.50	0.75
	IF	50			100			50			100		
CIFAR-10	MixMatch	70.78	68.05	65.39	65.85	59.69	60.13	53.68	49.30	48.62	50.40	33.00	36.90
	FixMatch	72.83	69.66	63.32	68.26	61.48	57.81	59.65	53.83	44.40	51.00	39.95	33.80
	CReST	74.37	69.69	62.84	65.70	61.62	52.93	65.15	53.60	43.12	49.63	41.30	23.85
	DASO	76.57	73.57	68.94	71.49	66.18	64.89	67.62	62.45	56.85	61.00	50.20	47.35
	MOOD	**78.35**	**78.49**	**75.17**	**74.71**	**73.74**	**69.16**	**75.28**	**72.83**	**65.65**	**65.22**	**64.25**	**59.52**
	IF	20			30			20			30		
CIFAR-100	MixMatch	37.33	37.43	35.21	35.96	33.61	32.73	20.17	19.40	19.09	14.00	14.57	14.66
	FixMatch	36.89	37.01	30.62	36.74	31.81	28.24	16.97	17.14	10.86	15.86	13.23	10.86
	CReST	31.63	30.13	29.17	29.68	27.19	25.52	16.51	13.46	13.46	10.91	9.31	8.63
	DASO	40.65	**39.69**	36.61	**37.85**	36.52	32.71	20.37	18.57	17.80	16.86	14.43	11.89
	MOOD	**40.98**	38.94	**37.59**	36.95	**36.95**	**36.15**	**29.29**	**26.46**	**23.46**	**22.43**	**24.09**	**19.37**

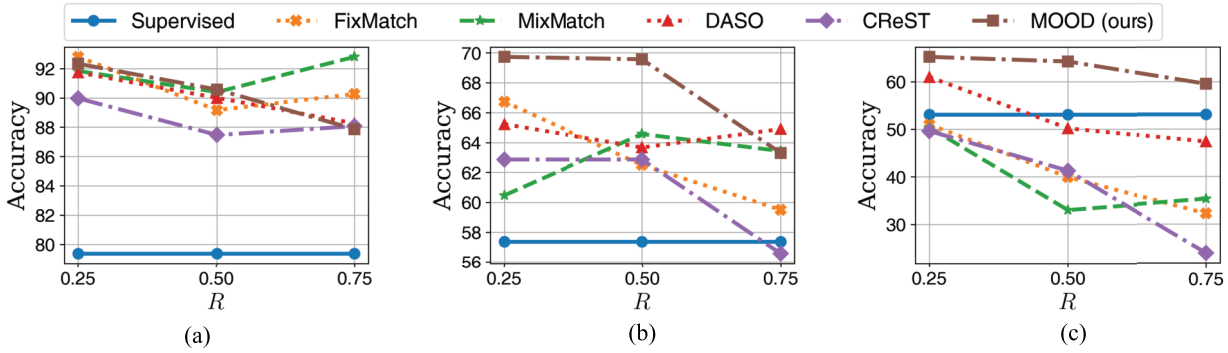


Fig. 4. Average accuracy (%) of three groups. (a) Head, (b) medium, and (c) tail. The experiments are conducted on CIFAR-10-LT with different values of R, IF = 100 and β = 30%. The backbone is ResNet-34.

with the inclusion of OOD samples in the unlabeled data. CIFAR-10/100 [53] and SVHN [54] are used as the in-distribution dataset, which is commonly adopted in the SSL literature [28]. We split a total of 5000 samples (500 samples per class for CIFAR-10/SVHN and 50 samples per class for CIFAR-100) from the original training dataset as validation data, while the original test dataset is used for testing, and further split the remaining training data (45 000 for CIFAR-10/100 and 68 357 for SVHN) into labeled and unlabeled data. We randomly select β = 10% and β = 30% of samples from training data to create the labeled set, and randomly discard training images per class according to the imbalance factor IF. IF takes the value from the set [20, 30, 50, 100]. Specifically, we denote N_1 and M_1 as the number of head class samples in labeled data and unlabeled ID data, $N_i = (\text{IF})^{-(i-1/C-1)} \cdot N_1$ and the same for M_i, while N_1 = 1500 and M_1 = 3000 for CIFAR-10/SVHN with β = 30% labeled data and N_1 = 50 and M_1 = 400 for CIFAR-100 with β = 10% labeled data. Tiny ImageNet [55] is used as the OOD dataset, which contains 10 000 test images from 200 classes. We use R to control the proportion of OOD samples to unlabeled samples, where the total amount of unlabeled data is fixed, and we randomly replace the original unlabeled ID samples with OOD samples according to R.

3) Implementation Details: We employ ResNet-34 [41] as our backbone architecture. Experiments related to different backbone architectures can be found in Section IV-F. Each compared method is trained for 250 000 iterations during standard training, with validation performed every 500 iterations. For CIFAR-10 and SVHN, we set the batch size of labeled data to 64 and the batch size of unlabeled data to 128. For CIFAR-100, the batch size of labeled data and unlabeled data is set to 16 and 32, respectively. We use the SGD optimizer with a basic learning rate of 0.03, momentum of 0.9, and weight decay of $1e^{-4}$. The temperature scaling factor T_1 and T_2 in L_{pap} are set to 0.1 and 0.007, respectively. The threshold

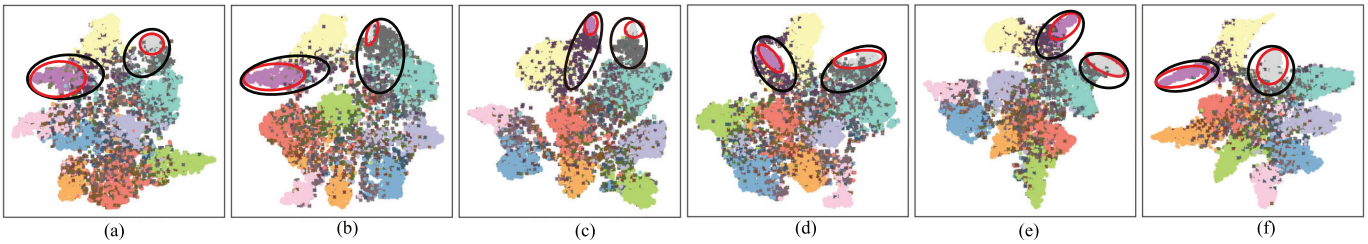


Fig. 5. t-SNE visualization of feature space of CIFAR-10-LT testing set, trained on CIFAR-10-LT with IF = 100 and R = 0.5. Ellipses show the feature space area of the tail classes 9 and 10. The red and black ellipses represent the clusters of correctly classified and ground-truth samples, respectively. (a) Supervised. (b) FixMatch. (c) MixMatch. (d) CReST. (e) DASO. (f) MOOD (ours).

TABLE II
COMPARISONS WITH THE OPEN-SET SSL METHODS ON CIFAR-10 AND CIFAR-100 BOTH WITH TINYIMAGENET AS OOD DATA

Dataset	CIFAR-10 ($IF = 100$)			CIFAR-100 ($IF = 100$)		
R	0.25	0.50	0.75	0.25	0.50	0.75
Supervised		62.25			25.61	
OpenMatch	64.96	62.29	60.74	29.47	28.71	26.29
MTCF	58.31	57.58	57.57	31.04	31.41	29.78
MOOD	**74.71**	**73.74**	**69.16**	**34.43**	**32.55**	**32.29**

TABLE III
ACCURACY (%) ON SVHN UNDER DIFFERENT R WITH FIXED IF = 100 AND β = 30%

Dataset	SVHN		
R	0.25	0.50	0.75
Supervised		84.54	
MixMatch	85.66	85.13	84.74
FixMatch	**88.33**	87.33	85.95
CReST	88.24	86.48	85.95
DASO	88.10	84.07	84.80
MOOD (ours)	88.23	**87.41**	**87.23**

τ in L_{cons} is set to 0.95. λ_{cons} and λ_{pap} in (6) are set to 1.0 and 0.2, respectively.

4) Evaluation Criteria: In the experiments, our performance evaluation refers to the standard protocol [28], measuring average top-1 accuracy (%) for each class. In addition, we categorize the classes of the ID dataset into three groups (head, medium, and tail) according to the number of samples in each class, with {3, 3, 4}, {3, 3, 4}, and {30, 35, 35} for CIFAR-10, SVHN, and CIFAR-100, respectively. We then compare the average accuracy of each group to further evaluate the performance of our method.

B. Numerical Results

To demonstrate the robustness of MOOD to the data imbalance factor and the proportion of OOD samples in unlabeled data, we compare the performance of MOOD and other methods on CIFAR-10, CIFAR-100, and SVHN datasets under different combinations of IF and R.

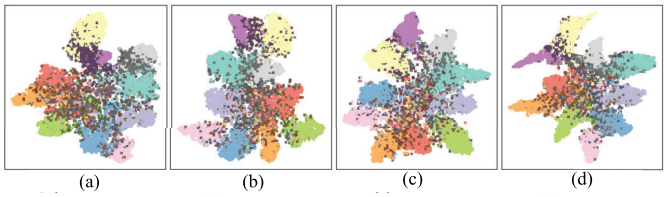


Fig. 6. t-SNE visualization of ablation experiments on CIFAR-10-LT testing set with IF = 100 and R = 0.5. (a) w. DB. (b) w. FU. (c) w. FO. (d) w. PaP.

TABLE IV
INFLUENCE OF THE HYPERPARAMETERS λ_1 AND λ_2 CONTROLS THE WEIGHT OF THE PAP LOSS AND THE OOD MIXUP LOSS. EXPERIMENTS ARE CONDUCTED WITH β = 30%, IF 100 AND R 0.5 FOR CIFAR-10. * INDICATES THE DEFAULT SETTING

(a) Influence of λ_2 with fixed $\lambda_1 = 1.0$					
λ_2	0.025	0.05	0.1*	0.2	0.3
Accuracy (%)	70.7	**71.0**	70.3	70.5	63.5
(b) Influence of λ_1 with fixed $\lambda_2 = 0.1$					
λ_1	0.5	0.75	1.0*	1.25	1.5
Accuracy (%)	66.4	67.3	**70.3**	69.7	69.4

The results of average accuracy and tail group accuracy are presented in Table I, showcasing the robustness of our method across various combinations of IF and R. Our approach exhibits superior performance on tail classes, surpassing DASO by margins of 7.66%, 10.38%, 8.8%, 4.22%, 14.05%, and 12.17% in different settings of CIFAR-10. Similarly, for various configurations of CIFAR-100, our method outperforms DASO by 8.92%, 7.89%, 5.66%, 5.57%, 9.66%, and 7.48%.

Moreover, Fig. 4 shows the average accuracy of three groups under different values of R with a fixed IF = 100 on CIFAR-10. We can observe that the performance of the head classes does not necessarily decrease with the increase of R, while the performance of the tail classes deteriorates and even some falls below the level of the supervised method. This illustrates that more OOD samples worsen the performance of the tail classes for other methods and that our method can mitigate this situation. The above results indicate that MOOD is more robust to complex real-world scenarios and more effective in addressing the challenge of recognizing the tail classes.

The performance comparison with the open-set SSL method (MTCF [21] and OpenMatch [18]) on the CIFAR-100 is

TABLE V
RESULTS OF ABLATION EXPERIMENTS ON CIFAR-10-LT UNDER THE SETTING OF IF = 100 AND R = 0.5

ID	Component				$\beta = 10\%$				$\beta = 30\%$			
	DB	FU	FO	PaP	Head	Medium	Tail	Acc.(%)	Head	Medium	Tail	Acc.(%)
1	-	-	-	-	**90.47**	41.43	18.45	46.95	**90.93**	**71.23**	37.85	63.79
2	✓	-	-	-	80.37	37.23	21.83	44.01	85.93	58.93	46.52	62.07
3	✓	✓	-	-	86.27	31.73	36.48	49.99	83.70	62.87	63.90	69.53
4	✓	-	✓	-	75.77	44.83	35.08	50.21	90.73	68.63	59.27	71.52
5	✓	-	✓	✓	82.23	**64.50**	**39.25**	**59.72**	90.57	69.57	**64.25**	**73.74**

shown in Table II. We control the distribution of the dataset by setting different R with a fixed IF = 100. Although the performances of MTCF and OpenMatch do not show a significant drop under a different R compared with other methods, the overall performances of these two methods are much lower than others due to their failure to address the data imbalance problem.

The experimental results on SVHN are shown in Table III, which demonstrates the competitive performance of our MOOD on SVHN. N_1 and M_1 are set to 1500 and 3000, respectively. When the ratio of OOD samples is high, our method exhibits comparable performance to other methods; while the number of OOD samples increases, our method exhibits its robustness.

C. Visualized Results

In order to better observe the feature space learned by each method, we use t-SNE to visualize the high-dimensional feature space before the classification layer. Fig. 5 shows the t-SNE visualization of the feature space about CIFAR-10 testing set. To depict the feature space of the tail classes, we employ ellipses with two colors to encompass the corresponding data clusters. The red ellipses cover the clusters of correctly classified tail class samples, while the black ellipses cover the ground-truth tail class samples. The feature space generalization of the tail classes can thus be evaluated by the overlap between two ellipses. By comparing the overlaps between red and black ellipses, we find that the overlaps of the compared methods are quite small, indicating squeezed feature space. In contrast, the feature space overlap of MOOD is much larger, demonstrating significant performance improvement in tail classes. In addition, the feature space of each class in Fig. 5(f) tends to stretch outwards, further supporting our idea that the feature space of the tail classes expands outwards rather than squeezing the feature space of other classes inward due to OOD samples.

D. Ablation Experiments

We conduct hyperparameter ablation on λ_1 and λ_2. As shown in Table IV, the results remain stable across a range of values, indicating that our method is not sensitive to the exact choice of hyperparameters. We further present ablation experiments on four major components of MOOD, namely, dual-biased sampling branch (DB), fusing labeled data with unlabeled data (FU), fusing labeled data with filtered unlabeled OOD data (FO), and the PaP loss item (PaP). In addition to the overall average accuracy, we evaluate the respective average results of the three groups. As shown in Table V, *Experiment ID-1* presents the baseline model without any modification, which performs well on the head classes but

TABLE VI
ACCURACY (%) OF DIFFERENT FEATURE LOSSES, PCL, CCL, AND ICL. β IS SET TO 30%, IF 100 AND R 0.5 FOR CIFAR-10. AS FOR CIFAR-100, IF IS SET TO 50 AND R 0.5

Dataset	CIFAR-10				CIFAR-100			
Group	Head	Med.	Tail	Acc.	Head	Med.	Tail	Acc.
PCL	87.20	64.77	48.27	64.90	**51.37**	27.34	11.63	27.10
CCL	89.70	67.00	40.40	63.17	49.13	27.94	**11.86**	28.67
ICL	80.90	62.00	53.20	64.15	43.53	19.77	10.69	23.26
PAP	**90.57**	**69.57**	**64.25**	**73.74**	48.33	**30.71**	11.57	**29.30**

TABLE VII
RESULTS ABOUT THE OOD FILTER WITH RESNET 34 AS THE BACKBONE AND CORRESPONDING ACCURACY UNDER DIFFERENT R SETTINGS WITH FIXED IF = 100 AND β = 30% ON CIFAR-10. THIS TABLE PRESENTS THE TPR AND TNR OF THE OOD FILTER

Dataset	IF	50			100		
	R	0.25	0.50	0.75	0.25	0.50	0.75
CIFAR-10	TPR	94.09	93.53	92.99	94.09	93.53	92.99
	TNR	99.76	100.00	100.00	99.76	100.00	100.00
	IF	20			30		
CIFAR-100	TPR	94.00	94.23	95.09	94.65	95.37	95.04
	TNR	99.75	99.92	100.00	99.11	100.00	99.95

TABLE VIII
COMPARISON OF FILTER-1 [49] BASED ON NEAREST NEIGHBORS AND FILTER-2 [56] BASED ON COMPACT FEATURE REPRESENTATION, UNDER DIFFERENT THRESHOLDS R ON CIFAR-10 WITH IF = 50, USING RESNET-34 AS THE BACKBONE. WE REPORT TPR, TNR, AND CLASSIFICATION ACCURACY (ACC. %)

Method	R	TPR	TNR	Acc
Filter-1	0.25	94.09	99.76	78.35
	0.50	93.53	100.00	78.49
	0.75	92.99	100.00	75.17
Filter-2	0.25	95.62	80.85	73.56
	0.50	96.28	89.40	74.35
	0.75	97.24	89.48	70.99

TABLE IX

COMPARISON OF AVERAGE ACCURACY (%) AND TAIL ACCURACY (%) WITH THE SSL METHODS WITH OOD FILTER ON CIFAR-10-LT WITH $\beta = 30\%$. WE HIGHLIGHT THE PERFORMANCE IMPROVEMENT IN BLUE WHEN COMPARING IT WITH THE RESULTS WITHOUT AN OOD FILTER, WHILE THE PERFORMANCE DEGRADATION IS HIGHLIGHTED IN RED

Dataset	CIFAR-10 (Average Accuracy, %)						CIFAR-10 (Tail Accuracy, %)					
IF	50			100			50			100		
R	0.25	0.50	0.75	0.25	0.50	0.75	0.25	0.50	0.75	0.25	0.50	0.75
MixMatch	71.84 +1.06	75.34 +7.29	72.89 +7.5	67.69 +1.84	66.61 +6.92	66.78 +6.65	56.90 +3.22	66.33 +17.03	60.75 +12.13	47.77 -2.63	42.40 +9.4	45.43 +8.53
FixMatch	76.42 +3.59	75.97 +6.31	73.37 +10.05	69.78 +1.52	70.39 +8.91	66.52 +8.71	65.52 +5.87	66.85 +13.02	63.42 +19.02	50.87 -0.13	56.05 +16.1	50.45 +16.65
CReST	74.44 +0.07	74.07 +4.38	71.91 +9.07	70.31 +4.61	69.64 +8.02	63.58 +10.65	62.70 -2.45	62.70 +9.1	60.40 +17.28	53.00 +3.37	51.77 +10.47	46.50 +22.65
DASO	77.09 +0.52	75.14 +1.57	72.89 +3.95	72.51 +1.02	69.57 +3.39	66.27 +1.38	69.10 +1.48	68.67 +6.22	64.42 +7.57	58.53 -2.47	56.23 +6.03	43.30 -4.05
MOOD	**78.35**	**78.49**	**75.17**	**74.71**	**73.74**	**69.16**	**75.28**	**72.83**	**65.65**	**65.22**	**64.25**	**59.52**

TABLE X

RESULTS ON CIFAR-10 (IF = 100) AND CIFAR-100 (IF = 50) USING WIDE RESNET WITH $\beta = 10\%$

Dataset	Method	R=0.25			R=0.50			R=0.75		
		Head	Tail	Avg.	Head	Tail	Avg.	Head	Tail	Avg.
CIFAR-10	FixMatch	**92.77**	33.65	61.56	89.80	31.55	59.05	**87.33**	27.02	**52.60**
	ABC	91.83	26.85	56.69	87.80	22.45	51.44	82.77	13.70	43.59
	CReST	92.73	33.85	60.76	**92.20**	23.50	57.55	88.57	23.15	51.95
	DASO	91.90	30.35	59.76	90.93	27.57	57.21	86.87	20.32	50.54
	MOOD	83.53	**46.20**	**62.58**	76.03	**49.83**	**61.01**	78.33	**31.35**	51.20
CIFAR-100	FixMatch	59.37	3.43	28.59	57.53	3.80	27.12	47.37	2.60	23.05
	ABC	57.83	6.11	27.12	54.43	7.43	26.08	39.03	1.83	17.96
	CReST	49.10	3.34	22.90	46.77	3.37	22.08	43.30	1.89	19.60
	DASO	**62.10**	1.91	26.29	**58.40**	1.49	24.18	**55.13**	1.17	22.96
	MOOD	52.40	**12.71**	**31.36**	48.33	**11.57**	**29.30**	47.17	**9.97**	**26.33**

poorly on the tail classes. *Experiment ID-2* incorporates the dual-biased sampling branch to increase the frequency of tail class samples, resulting in improved performance for the tail classes. *Experiment ID-3* leverages unlabeled data for the mixup, which greatly improves the tail class performance but reduces head class performance by up to 7% compared with Experiment ID-1. We argue that fusing with all unlabeled data, which contains more head class samples, causes the feature space of head classes to shrink and their performance to degrade, while the diversity of tail classes increases. These results confirm our hypothesis regarding issue (2) as described in Section III-B. *Experiment ID-4* only leverages filtered OOD samples for mixup, resulting in improved performance for tail classes. This suggests that incorporating OOD samples for mixup can enhance the diversity of tail classes. In *Experiment ID-5*, we introduce the PaP loss term into the training process, which further improves the performance of tail classes with maintaining the performance of the head class. These ablation experiments demonstrate the effectiveness of each module in our method and verify the significant contribution of the OOD mixup combined with the PaP loss term to the tail classes feature space. To further validate the advantage of using OOD data, we compare MOOD with a variant that replaces OOD mixup with standard oversampling. Compared with standard oversampling (64.96% overall/52.83% tail), MOOD achieves significantly higher accuracy (73.74%/64.25%). This is because oversampling fails to introduce semantic diversity at the feature level, while OOD-based mixup enriches tail-class representations through distributional variation. Moreover, we visualize the features extracted by the feature extractor on CIFAR-10 testing set for the ablation experiments in Fig. 6. The feature space of each class shown in Fig. 6(d) becomes more compact and tends to extend outward compared with other figures.

Our PaP loss exhibits a similar structure to the conventional contrastive loss. In order to illustrate the difference between the PAP loss, the instance contrastive loss (ICL) [46], the prototype contrastive loss (PCL) [57], and the class-aware contrastive loss (CCL) [20], we conducted a feature loss difference experiment. The experimental results are shown in Table VI. From the results, it is evident that the PAP loss offers better assistance to the tail classes and significantly

improves the model's performance compared with the ICL, PCL, and CCL losses. This finding highlights the effectiveness and superiority of the PAP loss in addressing the challenges posed by tail classes in the given experiments.

E. Evaluation of Ood Filter

We conduct experiments to evaluate the performance of the OOD filter, and the results are presented in Table VII. The table displays the true negative rate (TNR) and true positive rate (TPR) of the OOD filter with a different R at a fixed IF = 100. The results demonstrate that the OOD filter is effective in recognizing ID samples and the performance of the OOD filter is a bottleneck of MOOD. To further evaluate the robustness of MOOD under different OOD filtering strategies, We evaluate the impact of an alternative OOD filter [56] under the IF = 50 setting on the CIFAR10 dataset. As shown in Table VIII, although the filter change leads to slight performance drops, our method still achieves SOTA results. In future work, we will consider improving OOD filter performance to further improve MOOD.

In addition, in order to ensure fairness in the experiments, we also evaluated the performance of other methods when utilizing an OOD filter. The results of this comparison are presented in Table IX. The overall performance of these methods is improved a lot, while the tail performance deteriorates. For example, in the case of IF = 100 and R = 0.75, the performance of CReST on the tail classes improved by 22.65%. Even compared with the other methods employing OOD filtering, our method still exhibits superior performance. After using the OOD filter, the average results of the model are improved, but for some methods like DASO, the performance of the tail class decreases. This is because some unlabeled tail samples are misclassified as OOD data and are filtered out. Whereas simply filtering out OOD samples can only produce suboptimal results for tail classes, our method offers a more effective approach for tail class optimization.

F. Influence of Different Model Backbones

The experiments compared with wide ResNet [42] as backbone can be found in Table X. The labeled data account for 10% of the entire training dataset. The results obtained by different networks as the backbone are similar, indicating that our method is insensitive to the backbone. When the proportion of OOD samples is larger, our model is more robust than other methods.

V. Conclusion

In this article, we propose MOOD, an imbalanced SSL method to deal with the challenges under more realistic scenarios. We fully exploit the potential of seemingly detrimental OOD data to improve the model robustness. The dual-biased sampling branch with OOD mixup provides an equal opportunity for all classes to learn with sample diversity. Meanwhile, we introduce the PaP loss to encourage the model to learn a more balanced and none-squeezed feature space, improving the overall performance of the model. Experimental results on SSL benchmarks demonstrate the effectiveness of the MOOD, achieving state-of-the-art performance. In future work, we plan to explore ways to further study imbalanced SSL with OOD data, considering the potential impact of domain transfer that has not been considered yet.

Data Available Statement

The datasets that support the findings of this study are available at: CIFAR-10/100: https://www.cs.toronto.edu/kriz/cifar.html, SVHN: http://ufldl.stanford.edu/housenumbers/, and Tiny ImageNet: https://www.kaggle.com/c/tiny-imagenet

References

[1] Y. Xu et al., "Dash: Semi-supervised learning with dynamic thresholding," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 11525–11536.

[2] R. He, J. Yang, and X. Qi, "Re-distributing biased pseudo labels for semi-supervised semantic segmentation: A baseline investigation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 6930–6940.

[3] H. Pham, Z. Dai, Q. Xie, and Q. V. Le, "Meta pseudo labels," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 11557–11568.

[4] K. Huang, J. Geng, W. Jiang, X. Deng, and Z. Xu, "Pseudo-loss confidence metric for semi-supervised few-shot learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2021, pp. 8671–8680.

[5] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1195–1204.

[6] Z. Ke, D. Wang, Q. Yan, J. Ren, and R. Lau, "Dual student: Breaking the limits of the teacher in semi-supervised learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6728–6736.

[7] Q. Xie, Z. Dai, E. Hovy, M.-T. Luong, and Q. V. Le, "Unsupervised data augmentation for consistency training," in *Proc. NIPS*, vol. 33, 2020, pp. 6256–6268.

[8] C. Wei, H. Wang, W. Shen, and A. Yuille, "CO_2: Consistent contrast for unsupervised visual representation learning," in *Proc. Int. Conf. Learn. Represent.*, Jan. 2020, pp. 1–13.

[9] D. Lee, S. Kim, I. Kim, Y. Cheon, M. Cho, and W.-S. Han, "Contrastive regularization for semi-supervised learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2022, pp. 3911–3920.

[10] L. Yang, H. Jiang, Q. Song, and J. Guo, "A survey on long-tailed visual recognition," *Int. J. Comput. Vis.*, vol. 136, pp. 1837–1872, May 2022.

[11] Y. Jin, M. Li, Y. Lu, Y.-M. Cheung, and H. Wang, "Long-tailed visual recognition via self-heterogeneous integration with knowledge excavation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 23695–23704.

[12] Z. Lai, C. Wang, S.-c. Cheung, and C.-N. Chuah, "SAR: Self-adaptive refinement on pseudo labels for multiclass-imbalanced semi-supervised learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2022, pp. 4090–4099.

[13] C. Wei, K. Sohn, and C. Mellina, "CReST: A class-rebalancing self-training framework for imbalanced semi-supervised learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 10857–10866.

[14] Y. Oh, D.-J. Kim, and I. S. Kweon, "DASO: Distribution-aware semantics-oriented pseudo-label for imbalanced semi-supervised learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2022, pp. 9786–9796.

[15] J. Kim, Y. Hur, S. Park, E. Yang, S. J. Hwang, and J. Shin, "Distribution aligning refinery of pseudo-label for imbalanced semi-supervised learning," in *Proc. NIPS*, vol. 33, 2020, pp. 14567–14579.

[16] K. Li et al., "CODA: A real-world road corner case dataset for object detection in autonomous driving," in *Proc. Eur. Conf. Comput. Vis.*, Oct. 2022, pp. 406–423.

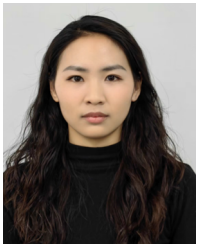
[17] L.-Z. Guo, Z.-Y. Zhang, Y. Jiang, Y.-F. Li, and Z.-H. Zhou, "Safe deep semi-supervised learning for unseen-class unlabeled data," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 3897–3906.

[18] K. Saito, D. Kim, and K. Saenko, "OpenMatch: Open-set consistency regularization for semi-supervised learning with outliers," 2021, *arXiv:2105.14148*.

[19] Y. Chen, X. Zhu, W. Li, and S. Gong, "Semi-supervised learning under class distribution mismatch," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, 2020, pp. 3569–3576.
[20] F. Yang et al., "Class-aware contrastive semi-supervised learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 14421–14430.
[21] Q. Yu, D. Ikami, G. Irie, and K. Aizawa, "Multi-task curriculum framework for open-set semi-supervised learning," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 438–454.
[22] J. Huang et al., "Trash to treasure: Harvesting OOD data with cross-modal matching for open-set semi-supervised learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 8290–8299.
[23] Z. Huang, J. Yang, and C. Gong, "They are not completely useless: Towards recycling transferable unlabeled data for class-mismatched semi-supervised learning," *IEEE Trans. Multimedia*, vol. 25, pp. 1844–1857, 2022.
[24] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "AutoAugment: Learning augmentation strategies from data," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 113–123.
[25] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "RandAugment: Practical automated data augmentation with a reduced search space," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 18613–18624.
[26] M. Assran et al., "Semi-supervised learning of visual features by non-parametrically predicting view assignments with support samples," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Jun. 2021, pp. 8443–8452.
[27] Z. Hu, Z. Yang, X. Hu, and R. Nevatia, "SimPLE: Similar pseudo label exploitation for semi-supervised classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 15099–15108.
[28] K. Sohn et al., "FixMatch: Simplifying semi-supervised learning with consistency and confidence," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2020, pp. 596–608.
[29] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel, "MixMatch: A holistic approach to semi-supervised learning," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 5050–5060.
[30] D. Berthelot et al., "ReMixMatch: Semi-supervised learning with distribution alignment and augmentation anchoring," 2019, *arXiv:1911.09785*.
[31] J. Li, C. Xiong, and S. C. H. Hoi, "CoMatch: Semi-supervised learning with contrastive graph regularization," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9475–9484.
[32] M. Zheng, S. You, L. Huang, F. Wang, C. Qian, and C. Xu, "SimMatch: Semi-supervised learning with similarity matching," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 14471–14481.
[33] Y. Fan, D. Dai, A. Kukleva, and B. Schiele, "CoSSL: Co-learning of representation and classifier for imbalanced semi-supervised learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 14554–14564.
[34] H. Lee, S. Shin, and H. Kim, "ABC: Auxiliary balanced classifier for class-imbalanced semi-supervised learning," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2021, pp. 7082–7094.
[35] H. Lee and H. Kim, "CDMAD: Class-distribution-mismatch-aware debiasing for class-imbalanced semi-supervised learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 23891–23900.
[36] Z. Li, L. Qi, Y. Shi, and Y. Gao, "IOMatch: Simplifying open-set semi-supervised learning with joint inliers and outliers utilization," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 15824–15833.
[37] A. Banitalebi-Dehkordi, P. Gujjar, and Y. Zhang, "AuxMix: Semi-supervised learning with unconstrained unlabeled data," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2022, pp. 3998–4005.
[38] T. Wei and K. Gan, "Towards realistic long-tailed semi-supervised learning: Consistency is all you need," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 3469–3478.
[39] H. Kong, S.-J. Kim, G. Jung, and S.-W. Lee, "Diversify and conquer: Open-set disagreement for robust semi-supervised learning with outliers," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Mar. 28, 2025, doi: 10.1109/TNNLS.2025.3547801.
[40] J. Ren et al., "Likelihood ratios for out-of-distribution detection," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 14707–14718.
[41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[42] S. Zagoruyko and N. Komodakis, "Wide residual networks," 2016, *arXiv:1605.07146*.
[43] C. Zhang et al., "MosaicOS: A simple and effective use of object-centric images for long-tailed object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 407–417.
[44] J. Cai, Y. Wang, and J. Hwang, "ACE: Ally complementary experts for solving long-tailed recognition in one-shot," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 112–121.
[45] G. Ghiasi et al., "Simple copy-paste is a strong data augmentation method for instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 2918–2928.
[46] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2018, *arXiv:1807.03748*.
[47] H. Zhang, M. Cissé, Y. Dauphin, and D. López-Paz, "Mixup: Beyond empirical risk minimization," in *Proc. Int. Conf. Learn. Represent.*, Jan. 2017, pp. 1–13.
[48] B. Zhou, Q. Cui, X.-S. Wei, and Z.-M. Chen, "BBN: Bilateral-branch network with cumulative learning for long-tailed visual recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9719–9728.
[49] Y. Sun, Y. Ming, X. Zhu, and Y. Li, "Out-of-distribution detection with deep nearest neighbors," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 20827–20840.
[50] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9729–9738.
[51] H. Wei, L. Tao, R. Xie, L. Feng, and B. An, "Open-sampling: Exploring out-of-distribution data for re-balancing long-tailed datasets," in *Proc. Int. Conf. Mach. Learn.*, Jan. 2022, pp. 23615–23630.
[52] Z. Zhong, L. Zhu, Z. Luo, S. Li, Y. Yang, and N. Sebe, "OpenMix: Reviving known knowledge for discovering novel visual categories in an open world," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 9462–9470.
[53] A. Krizhevsky, "Learning multiple layers of features from tiny images," Univ. Toronto, Tech. Rep., 2009.
[54] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," in *Proc. NeurIPS Workshop Deep Learn. Unsupervised Feature Learn.*, Jan. 2011, pp. 1–12.
[55] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.
[56] S. Saha, J. Gawlikowski, J. Nandy, and X. X. Zhu, "Compact feature representation for unsupervised ood detection," in *Proc. IGARSS - IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2022, pp. 3143–3146.
[57] J. Li, P. Zhou, C. Xiong, and S. C. Hoi, "Prototypical contrastive learning of unsupervised representations," in *Proc. Int. Conf. Learn. Represent.*, 2021, pp. 1–21.

Yang Lu (Senior Member, IEEE) received the Ph.D. degree in computer science from Hong Kong Baptist University, Hong Kong, China, in 2019.

He is currently an Assistant Professor with the Department of Computer Science and Technology, School of Informatics, Xiamen University, Xiamen, China. His current research interest is open-world robust deep learning, such as long-tail learning, federated learning, label-noise learning, and continual learning.



Xiaolin Huang received the B.S. degree in computer science from China University of Geosciences, Wuhan, China, in 2021. She is currently pursuing the M.S. degree in computer science with the Department of Computer Science and Technology, School of Informatics, Xiamen University, Xiamen, China.

Her current research interests include semi-supervised learning, contrastive learning, and long-tail learning.

Yizhou Chen received the B.S. degree in computer science and technology from Xiamen University, Xiamen, China, in 2023, where he is currently pursuing the M.S. degree in artificial intelligence with the Institute of Artificial Intelligence.

His current research interests include semi-supervised learning and long-tail learning.

Mengke Li (Member, IEEE) received the Ph.D. degree from Hong Kong Baptist University, Hong Kong, SAR, China, in 2022.

She is currently an Assistant Professor with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. Her current research interests include imbalanced data learning, long-tail learning, and computer vision.

Yan Yan (Senior Member, IEEE) received the Ph.D. degree in information and communication engineering from Tsinghua University, Beijing, China, in 2009.

He worked as a Research Engineer with Japan Research and Development Center, Nokia, Tokyo, Japan, from 2009 to 2010. He worked as a Project Leader with the Panasonic Singapore Laboratory, Singapore, in 2011. He is currently a Full Professor with the School of Informatics, Xiamen University, Xiamen, China. He has published around 100 papers in international journals and conferences, including IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IJCV, IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, CVPR, ICCV, ECCV, AAAI, and ACM MM. His research interests include computer vision and pattern recognition.

Chen Gong (Senior Member, IEEE) received the dual Ph.D. degree from Shanghai Jiao Tong University (SJTU), Shanghai, China, and the University of Technology Sydney (UTS), Sydney, Australia, in 2016.

He is currently a Full Professor with the School of Automation and Intelligent Sensing, SJTU. He has published more than 130 technical papers at prominent journals and conferences such as JMLR, IJCV, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE TRANSACTIONS ON IMAGE PROCESSING, ICML, NeurIPS, ICLR, CVPR, ICCV, ECCV, AAAI, and IJCAI. His research interests mainly include machine learning, data mining, and learning-based vision problems.

Dr. Chen won the "Excellent Doctorial Dissertation Award" of Chinese Association for Artificial Intelligence, the "Young Elite Scientists Sponsorship Program" of China Association for Science and Technology, and the "Wu Wen-Jun AI Excellent Youth Scholar Award." He was also selected as the "Global Top Chinese Young Scholars in AI" released by Baidu and "World's Top 2% Scientists" released by Stanford University. He serves as the Area Chair or a Senior PC Member for several top-tier conferences, such as ICML, ICLR, AAAI, IJCAI, ECML-PKDD, AISTATS, ICDM, and ACM MM; and an Associate Editor for IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, *Neural Networks*, and NePL.

Hanzi Wang (Senior Member, IEEE) received the Ph.D. degree in computer vision from Monash University, Melbourne, VIC, Australia, in 2004.

He was a Distinguished Professor of Minjiang Scholars in Fujian province and is currently the Founding Director of the Center for Pattern Analysis and Machine Intelligence, Xiamen University, Xiamen, China. His research interests are concentrated on computer vision and pattern recognition including visual tracking, object detection, person reidentification, pedestrian attribute recognition, action recognition, robust model fitting, and related fields.