Class-Wise Contrastive Prototype Learning for Semi-Supervised Classification Under Intersectional Class Mismatch

Mingyu Li, Tao Zhou, Bo Han, *Senior Member, IEEE*, Tongliang Liu, Xinkai Liang, Jiajia Zhao, and Chen Gong, *Senior Member, IEEE*

***Abstract*—Traditional Semi-Supervised Learning (SSL) classification methods focus on leveraging unlabeled data to improve the model performance under the setting where labeled set and unlabeled set share the same classes. Nevertheless, the above-mentioned setting is often inconsistent with many real-world circumstances. Practically, both the labeled set and unlabeled set often hold some individual classes, leading to an intersectional class-mismatch setting for SSL. Under this setting, existing SSL methods are often subject to performance degradation attributed to these individual classes. To solve the problem, we propose a Class-wise Contrastive Prototype Learning (CCPL) framework, which can properly utilize the unlabeled data to improve the SSL classification performance. Specifically, we employ a supervised prototype learning strategy and a class-wise contrastive separation strategy to construct a prototype for each known class. To reduce the influence of the individual classes in unlabeled set (i.e., out-of-distribution classes), each unlabeled example can be weighted reasonably based on the prototypes during classifier training, which helps to weaken the negative influence caused by out-of-distribution classes. To reduce the influence of the individual classes in labeled set (i.e., private classes), we present a private assignment suppression strategy to suppress the improper assignments of unlabeled examples to the private classes with the help of the prototypes. Experimental results on four benchmarks and one real-world dataset show that our CCPL has a clear advantage over fourteen representative SSL methods as well as two supervised learning methods under the intersectional class-mismatch setting.**

***Index Terms*—Contrastive learning, intersectional class mismatch, private assignment suppression, prototype learning, semi-supervised learning.**

I. Introduction

TRADITIONAL Semi-Supervised Learning (SSL) has achieved remarkable success over the past decades because of its ability to boost classification performance by deploying abundant unlabeled examples, where the unlabeled examples are drawn from the same data distribution as the labeled examples [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13]. In detail, traditional SSL approaches assume that labeled set and unlabeled set share the same class space, as shown in Fig. 1(a). Mathematically, we have $\mathcal{Y}_l = \mathcal{Y}_u$, where $\mathcal{Y}_l$ and $\mathcal{Y}_u$ denote the class spaces of the labeled set and unlabeled set, respectively.

Unfortunately, the class space of the unlabeled set is often different from that of the labeled set in practice, and thus violating the above-mentioned assumption. In this situation, traditional SSL methods often perform worse than only using the labeled data [14], [15]. Some works have been done to tackle this problem [16], [17], [18], [19], [20], [21], [22], [23], [24], [25]. Specifically, they focus on the problem where $\mathcal{Y}_l = \mathcal{Y}_l \cap \mathcal{Y}_u$, namely the "traditional class-mismatch" setting [14] (see Fig. 1(b)). They emphasize leveraging *In-Distribution* (ID) data while trying to decrease the negative influence of the *Out-Of-Distribution* (OOD) data. Note that, the ID data denote the unlabeled data of which the ground-truth labels are from the *ID Classes*, and OOD data refer to the unlabeled data of which the ground-truth labels are from the *OOD Classes*.

However, in reality, the assumption of traditional class-mismatch setting may be violated as well, because the labeled set and unlabeled set may not only share some common classes but also have their own individual classes. For example, while taking photos of target wildlife for ecological research with automated cameras, it is very possible that some interested animals will not be snapped due to the appearing occasionality of the species. At the same time, some species that are not our target may be caught by the camera as well. In this situation, the labeled data and unlabeled data both contain a shared class set but also hold an individual class set, respectively, *i.e.*, $\mathcal{Y}_u \cap \mathcal{Y}_l \neq \emptyset$,

Manuscript received 15 July 2023; revised 13 December 2023 and 8 February 2024; accepted 1 March 2024. Date of publication 18 March 2024; date of current version 31 July 2024. This work was supported in part by the NSF of China under Grant 62336003, Grant 12371510, Grant 62172228, and Grant 62376235, in part by the NSF of Jiangsu Province under Grant BZ2021013, in part by the NSF for Distinguished Young Scholar of Jiangsu Province under Grant BK20220080, and in part by the Fundamental Research Funds for the Central Universities under Grant 30920032202 and Grant 30921013114. The Associate Editor coordinating the review of this manuscript and approving it for publication was Dr C. Luo. *(Corresponding authors: Chen Gong; Tao Zhou.)*

Mingyu Li, Tao Zhou, and Chen Gong are with the PCA Laboratory, Key Laboratory of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education, School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: taozhou.dreams@gmail.com; chen.gong@njust.edu.cn).

Bo Han is with the School of Computer Science, Hong Kong Baptist University, Hong Kong, SAR, China (e-mail: bhanml@comp.hkbu.edu.hk).

Tongliang Liu is with the School of Computer Science, The University of Sydney, Camperdown, NSW 2008, Australia (e-mail: tongliang.liu@sydney.edu.au).

Xinkai Liang and Jiajia Zhao are with the Science and Technology on Complex System Control and Intelligent Agent Cooperation Laboratory, Beijing 100074, China (e-mail: lxk820@126.com; zhaojiajia1982@gmail.com).

This article has supplementary downloadable material available at https://doi.org/10.1109/TMM.2024.3377123, provided by the authors.

Digital Object Identifier 10.1109/TMM.2024.3377123

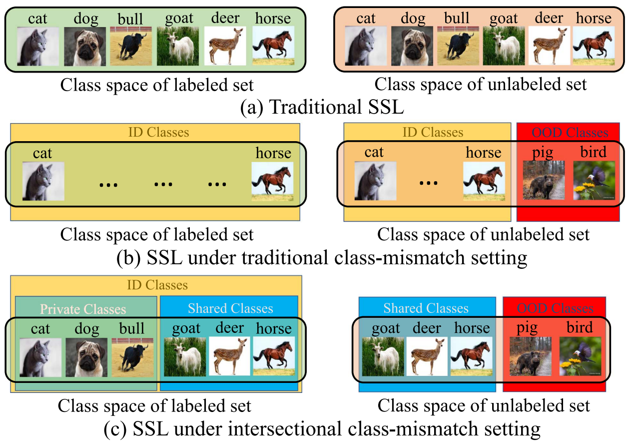


Fig. 1. (a) In traditional SSL setting, the labeled set and unlabeled set have the same class space. (b) In traditional class-mismatched SSL setting, the classes of the labeled set form a subset of the classes in the unlabeled set. The classes shared by both labeled and unlabeled sets are called "ID Classes", and the remaining classes in the unlabeled set constitute "OOD Classes". (c) In our intersectional class-mismatched SSL setting, the classes shared by both labeled and unlabeled sets are called "Shared Classes". The classes that only exist in the labeled set and unlabeled set are called "Private Classes" and "OOD Classes", respectively.

$\mathcal{Y}_l - (\mathcal{Y}_l \cap \mathcal{Y}_u) \neq \emptyset$, and $\mathcal{Y}_u - (\mathcal{Y}_l \cap \mathcal{Y}_u) \neq \emptyset$, leading to an "intersectional class-mismatch" setting (see Fig. 1(c)).

Under the intersectional class-mismatch setting, the above-mentioned SSL methods cannot work well because of two reasons, namely: 1) The pseudo label determination of unlabeled examples or the detection of OOD examples is based on the linear predictions made by a linear classifier (*i.e.*, fully connected layers trained with cross-entropy loss). As the classifier trained with cross-entropy loss is often encouraged to produce confident predictions, the predictions could be easily affected by an overconfidence issue [26]. As a result, some OOD examples could be wrongly yet confidently assigned to ID classes, which results in the deterioration of the semi-supervised classification performance [17], [22], [27], [28]. 2) The negative influence caused by private classes has not been considered. Specifically, as unlabeled set does not contain data belonging to private classes, thereby assigning unlabeled shared examples to private classes may compromise accuracy in classifying examples of ID classes. Similarly, assigning OOD examples to private classes adversely affects detection of OOD examples. Therefore, it is critical to preventing unlabeled data from being assigned to private classes.

To this end, we propose a Class-wise Contrastive Prototype Learning (CCPL) framework to deal with the intersectional class-mismatched SSL classification problem. CCPL utilizes the labeled examples and selected unlabeled examples to compute a prototype for each known class, which will be used to predict the pseudo label and evaluate the importance of each unlabeled example rather than using linear prediction. As a result, the negative influence caused by the overconfidence issue can be avoided, and the impact of each unlabeled example can be evaluated well, so Problem 1) mentioned above can be tackled. Specifically, we adopt supervised prototype learning and class-wise contrastive separation to get reliable prototypes by utilizing both labeled and unlabeled examples. With reliable prototypes, the weight of each unlabeled example can be calculated reasonably, therefore weakening the negative influence of OOD examples while training a classifier. Moreover, to decrease the assignments of unlabeled examples to private classes, we propose a private assignment suppression strategy to discourage those incorrect assignments with the help of prototypes, so the above-mentioned Problem 2) can be addressed. The advantage of our CCPL over fourteen representative SSL methods as well as two supervised learning methods under the intersectional class-mismatch setting is validated by conducting comparison experiments on four popular benchmarks and one real-world dataset.

The contributions of our work are four-fold:

1) We propose a novel Class-wise Contrastive Prototype Learning (CCPL) to deal with SSL under the intersectional class-mismatch setting where both labeled set and unlabeled set contain their own individual classes.
2) We utilize the unlabeled examples under the guidance of generated prototypes instead of previous linear predictions, which can evaluate the impact of each unlabeled example reasonably during classifier training.
3) A private assignment suppression strategy is presented to prevent unlabeled examples from being assigned to private classes, which can weaken the negative influence caused by private classes.
4) Comprehensive experiments under intersectional class mismatch on typical real-world datasets (*i.e.*, MNIST, SVHN, CIFAR-10, ImageNet-100 and Fundus) reveal that CCPL outperforms many existing state-of-the-art methods, including class-matched SSL methods (*e.g.*, Π-model [29] and FixMatch [9]) and class-mismatched SSL methods (*e.g.*, Uncertainty Aware Self-Distillation [17] and Safe Deep Semi-Supervised Learning [18]).

II. Related Work

A. Traditional Semi-Supervised Learning

Traditional SSL methods mainly employ three strategies, namely entropy minimization, consistency regularization, and data augmentation.

The entropy minimization methods assign labels to unlabeled examples with high confidence by minimizing the entropy of label predictions [30], [31], [32], [33], [34]. For example, Pseudo-Labeling (PL) [32] uses the classifier itself to select artificial label with the highest prediction probability as the pseudo label of each unlabeled example, then trains the classifier with all examples in a supervised manner. Uncertainty-aware Pseudo-label Selection [33] enhances the quality of the pseudo labels for unlabeled examples by exploiting the uncertainty in predictions.

The consistency regularization methods require the predictions of unlabeled data keeping unchanged under the influence of perturbations [29], [35], [36]. For instance, Π-model (PI) [29] requires the same classification prediction between two different perturbations on the same input data. Virtual Adversarial Training (VAT) [35] generates perturbations adversarially for unlabeled examples, and then requires consistent label predictions of perturbed and original versions.

Recently, data augmentation methods have become popular and gained remarkable success [9], [10], [37], [38], [39], [40]. For example, FixMatch [9] employs both weak and strong data augmentations on the same unlabeled example, and requires the two predictions of two different augmented image examples to be consistent. FlexMatch [40] tries to improve the performance of FixMatch by setting different thresholds for data selection according to the learning status of corresponding labels.

Unfortunately, the traditional SSL methods assume that the labeled and unlabeled sets share identical class spaces, making them unable to handle OOD examples and private examples. In contrast, our CCPL is specifically designed for intersectional class-mismatch settings, allowing it to effectively mitigate the negative influences of OOD examples and private examples.

B. Traditional Class-Mismatched Semi-Supervised Learning

SSL methods under the traditional class-mismatch setting focus on dealing with the problem by discarding the OOD examples from the training set, which is inspired by [14], [29]. For instance, Safe Deep Semi-Supervised Learning (DS^{3}L) [18] enhances the classification performance by developing a meta-learning scheme to reduce the weights of OOD examples while training a classifier. Multi-Task Curriculum learning Framework (MTCF) [23] utilizes curriculum learning [41], [42] to construct an ordered sequence, and weighs ID examples and OOD examples differently from easy to difficult. OpenMatch [21] proposes a new soft open-set consistency loss to weaken the negative influence of OOD examples by regularizing the predictions of different augmented examples made by the classifier. Trash to Treasure (T2T) [20] aims to detect OOD examples by employing a cross-modal matching strategy to improve feature learning without affecting classification performance. Class-aware Contrastive Semi-Supervised Learning (CCSSL) [22] utilizes both class-wise clustering and image-wise contrastive learning [43] to distinguish OOD data from ID data. SAFE-STUDENT [44] proposes an energy-discrepancy strategy based on energy [45] to assign a score to each unlabeled example, and thus weakening the negative influence caused by OOD examples. Out-of-Distributed Semantic Pruning (OSP) [46] proposes an aliasing OOD matching module and a soft orthogonality regularization to detect OOD data via semantic information.

However, the traditional class-mismatched SSL methods overlook private examples and rely on fully connected layers to detect OOD examples, leading to an overconfidence issue. In contrast, our CCPL addresses private examples through a proposed private assignment suppression strategy and mitigates the overconfidence issue by constructing prototypes.

C. Intersectional Class-Mismatched Semi-Supervised Learning

In recent years, some works have begun to explore SSL under intersectional class-mismatch setting, where labeled set contains private classes and unlabeled set contains OOD classes. For example, Uncertainty Aware Self-Distillation (UASD) [17] tries to find out the OOD data by averaging the historical prediction probability made by a self-distillation strategy. Class-shAring data detection and Feature Adaptation (CAFA) [47] employs both domain information and class information to deal with class mismatch and feature mismatch. However, for UASD, it does not take the private classes in labeled set into consideration during model designing, therefore the ability of UASD in dealing with the negative influence caused by private classes is very limited. For CAFA, it relies on linear prediction for shared example detection, which impairs its performance under intersectional class mismatch.

TABLE I
VARIABLES AND DEFINITIONS

Variables	Definitions
$\mathcal{D}$	training set
$\mathcal{Y}$	class space of training set
$\mathbf{x}_i$	the i-th labeled example
$\mathbf{y}_i$	class label of $\mathbf{x}_i$
$\mathbf{u}_i$	the i-th unlabeled example
$\mathbf{z}_i^l$	embedding of $\mathbf{x}_i$
$\mathbf{z}_i^u$	embedding of $\mathbf{u}_i$
$\mathbf{c}_i^u$	classification prediction of $\mathbf{u}_i$
$\mathbf{p}_k$	prototype for class k
$m_{i,k}$	similarity between $\mathbf{u}_i$ and $\mathbf{p}_k$
$\mathbf{Q}_k$	embedding queue for class k
$\mathbf{s}_k$	class-wise similarity for class k
$\mathbf{t}_k$	similarity ratio for class k

III. Proposed Method

A. Preliminaries

Under the intersectional class-mismatch setting, we let $\mathcal{D}_l = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_{n_l}, y_{n_l})\}$ denote the labeled set with $y_i \in \mathcal{Y}_l = \{1, 2, \ldots, K\}$, where n_l and K mean numbers of the labeled examples and the known classes, respectively. Besides, we use $\mathcal{D}_u = \{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{n_u}\}$ to denote the unlabeled set of which the class space is denoted as $\mathcal{Y}_u$, and n_u means the number of the unlabeled examples with typically $n_l \ll n_u$. Here the shared classes $\mathcal{Y}_{shared} = \mathcal{Y}_l \cap \mathcal{Y}_u \neq \emptyset$, the private classes $\mathcal{Y}_{pri} = \mathcal{Y}_l - (\mathcal{Y}_l \cap \mathcal{Y}_u) \neq \emptyset$, and the OOD classes $\mathcal{Y}_{ood} = \mathcal{Y}_u - (\mathcal{Y}_l \cap \mathcal{Y}_u) \neq \emptyset$. The examples of which the ground-truth labels fall into $\mathcal{Y}_{shared}$ and $\mathcal{Y}_{pri}$ constitute a shared set $\mathcal{D}_{shared}$ and a private set $\mathcal{D}_{pri}$, respectively. In addition, the unlabeled examples of which the true labels fall into $\mathcal{Y}_{ood}$ constitute an OOD set $\mathcal{D}_{ood}$. Note that $\mathcal{Y}_{pri}$, $\mathcal{Y}_{ood}$, and $\mathcal{Y}_{shared}$ are unknown during training, so are $\mathcal{D}_{pri}$, $\mathcal{D}_{ood}$, and $\mathcal{D}_{shared}$. During testing, the class space of presented test examples is the same as $\mathcal{Y}_l$, as the classes in $\mathcal{Y}_l$ are of our interests. Table I lists the important variables we will use throughout the paper.

B. Overview of CCPL

The overall framework of our CCPL approach is shown in Fig. 2, where E, R_{pro} and R_{cls} denote encoder, projection head, and classification head, respectively. Specifically, E is used to extract a representation $\mathbf{f}$ for an input example. R_{pro} projects the

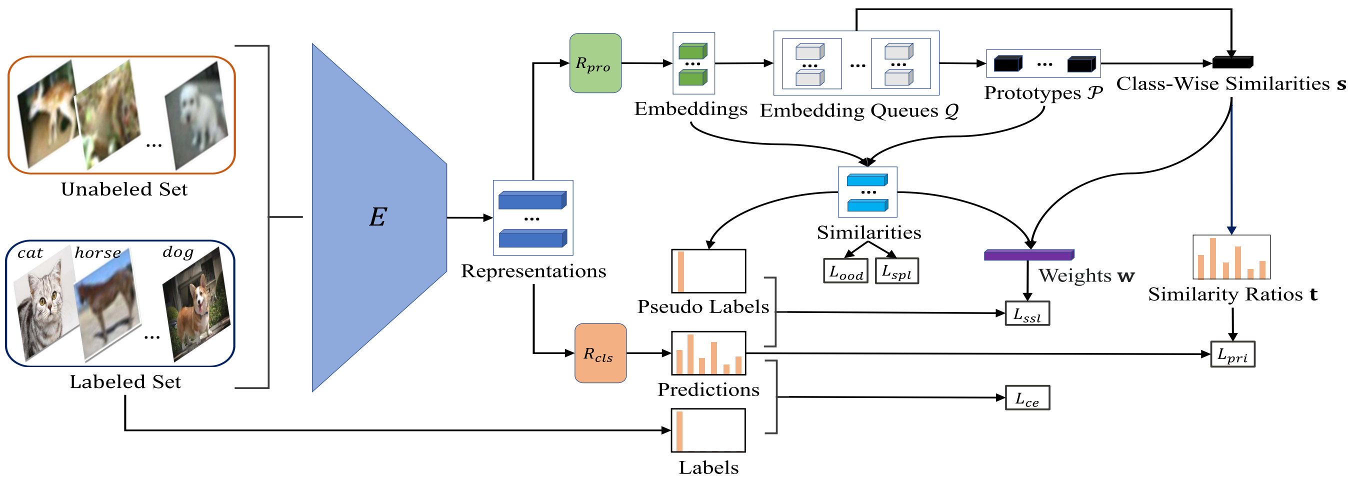


Fig. 2. Pipeline of our CCPL approach. Our CCPL will construct prototypes to deal with the negative influences caused by $\mathcal{Y}_{ood}$ and $\mathcal{Y}_{pri}$. Specifically, we firstly employ the embeddings of training examples to construct $\mathcal{P}$. Then, weights $\mathbf{w}$ and similarity ratios $\mathbf{t}$ are computed with $\mathcal{P}$ to weaken the importance of OOD examples and alleviate the wrong assignments to $\mathcal{Y}_{pri}$, respectively.

high-dimensional representation $\mathbf{f}$ to a low-dimensional embedding $\mathbf{z}$, which can be used to generate prototypes and compute similarities. R_{cls} is used to get a prediction $\mathbf{c}$ based on $\mathbf{f}$ in both training and testing stages. Note that, $R_{cls}(E(\cdot))$ denotes the classifier, which is used to validate the effectiveness of CCPL on the test set. Our CCPL includes four key components, namely: (1) *Supervised Prototype Learning* (Section III-C) employs the labeled examples to train a classifier, and constructs embedding queues $\mathcal{Q}$ to generate reasonable prototypes $\mathcal{P}$ based on the embeddings of labeled examples. (2) *Class-wise Contrastive Separation* (Section III-D) adopts a contrastive learning mechanism to enhance the quality of embeddings for unlabeled examples, and selects embeddings of unlabeled examples into $\mathcal{Q}$ to improve the reliability of $\mathcal{P}$. (3) *Private Assignment Suppression* (Section III-E) trains the classifier to suppress the improper assignments of unlabeled examples to the private classes under the guidance of similarity ratios $\mathbf{t}$. (4) *Weighted Semi-Supervised Learning* (Section III-F) trains the classifier to get consistent predictions on the unlabeled examples instructed by weights $\mathbf{w}$. Note that, both $\mathbf{t}$ and $\mathbf{w}$ are computed based on embeddings, $\mathcal{P}$ and $\mathcal{Q}$.

C. Supervised Prototype Learning

Leveraging labeled examples is critical to improving classification ability and generating reasonable prototypes. Firstly, we conduct supervised learning on the labeled examples with $\mathcal{L}_{ce}$ which is the classical cross-entropy loss. Then, we construct an embedding queue set $\mathcal{Q} = \{\mathbf{Q}_k\}_{k=1}^{K}$ based on the low-dimensional embedding of each labeled example $\mathbf{x}_i$, denoted by $\mathbf{z}_i^l = R_{pro}(E(\mathbf{x}_i))$. In detail, we let $\mathbf{Q}_k$ be the embedding queue for class k, and $\mathbf{z}_i^l$ will be put into $\mathbf{Q}_k$ when its ground-truth label y_i is k. The length of $\mathbf{Q}_k$ ($k \in \{1, 2, \ldots, K\}$) is set to L. Then, we obtain a prototype set $\mathcal{P} = \{\mathbf{p}_k\}_{k=1}^{K}$ based on $\mathcal{Q}$. Specifically, the prototype $\mathbf{p}_k$ is calculated by averaging the embeddings in $\mathbf{Q}_k$ at each iteration, and $\mathbf{Q}_k$ is updated at each iteration by pushing new embeddings of labeled examples into $\mathbf{Q}_k$, while discarding the earliest ones if $\mathbf{Q}_k$ is full. Note that, the prototypes are computed by a model trained using cross-entropy loss, implying that the prototypes are not randomly initialized.

To get reliable prototypes, we define $\mathcal{L}_{spl}$ to cluster the embeddings of labeled examples. Particularly, we encourage the embedding of each labeled example to be similar to the prototype of its true label, and meanwhile reduce the similarities between the embedding and other prototypes. The similarity between two embeddings $\mathbf{z}_i$ and $\mathbf{z}_j$ is calculated by $\exp(\mathbf{z}_i \cdot \mathbf{z}_j/\tau)$, where τ denotes a hyper-parameter. Therefore, the objective function of $\mathcal{L}_{spl}$ is

$$\mathcal{L}_{spl} = -\frac{1}{n_l}\sum_{i=1}^{n_l} \ln\left(\frac{\exp\left(\mathbf{z}_i^l \cdot \mathbf{p}_{y_i}/\tau\right)}{\sum_{k\neq y_i}^{K} \exp\left(\mathbf{z}_i^l \cdot \mathbf{p}_k/\tau\right)}\right). \tag{1}$$

As a result, we can utilize the labeled examples to train a classifier and get prototypes of all known classes.

D. Class-Wise Contrastive Separation

To further improve the reliability of $\mathcal{P}$, we select safe embeddings of unlabeled examples into $\mathcal{Q}$ at each iteration. We firstly calculate class-wise similarities $\mathbf{s}$, where the k-th element s_k denotes the class-wise similarity of the k-th class. The class-wise similarity of the k-th class is the average of the similarity between $\mathbf{p}_k$ and each embedding in $\mathbf{Q}_k$. For each unlabeled example $\mathbf{u}_i$, we define the embedding of its weakly augmented version as $\mathbf{z}_i^{u,weak} = R_{pro}(E(Aug_{weak}(\mathbf{u}_i)))$, where $Aug_{weak}(\cdot)$ denotes the weak data augmentation used in FixMatch [9]. For convenience, we let $\mathbf{m}_i$ denote the similarities between $\mathbf{z}_i^{u,weak}$ and all prototypes, where the k-th element $m_{i,k} = \exp(\mathbf{z}_i^{u,weak} \cdot \mathbf{p}_k/\tau)$. If the maximum value $m_{i,\widehat{k}_i}$ in $\mathbf{m}_i$ is larger than $s_{\widehat{k}_i}$, it means that $\mathbf{z}_i^{u,weak}$ is more similar to $\mathbf{p}_{\widehat{k}_i}$ than existing embeddings in $\mathbf{Q}_{\widehat{k}_i}$. It indicates that $\mathbf{z}_i^{u,weak}$ contains more valuable information for improving the reliability of $\mathbf{p}_{\widehat{k}_i}$ when compared with other embeddings in $\mathbf{Q}_{\widehat{k}_i}$. In this case, $\mathbf{z}_i^{u,weak}$ is considered safe and is selected into $\mathbf{Q}_{\widehat{k}_i}$.

Therefore, we select the safe embeddings by computing

$$v(\mathbf{u}_i) = \begin{cases} 1, m_{i,\widehat{k}_i} > s_{\widehat{k}_i}, \\ 0, m_{i,\widehat{k}_i} \le s_{\widehat{k}_i}, \end{cases} \quad \widehat{k}_i = \underset{k}{\operatorname{argmax}}\, m_{i,k}. \tag{2}$$

Specifically, if $v(\mathbf{u}_i) = 1$, $\mathbf{z}_i^{u,weak}$ will be put into $\mathbf{Q}_{\widehat{k}_i}$ to update $\mathbf{p}_{\widehat{k}_i}$, otherwise, $\mathbf{z}_i^{u,weak}$ should be discarded.

To get reliable prototypes, it is also important to enhance the quality of embeddings for unlabeled examples. Particularly, in contrast to unlabeled shared examples, OOD examples do not belong to any known class, so we need to push the embeddings of OOD examples far from prototypes and meanwhile exploit useful information of unlabeled shared examples. Traditional contrastive learning [48] can utilize the unlabeled examples effectively and isolate the OOD examples from others. However, it will affect the classification performance on shared classes, because traditional contrastive learning pushes embeddings of different unlabeled examples far from the others even if they belong to the same shared class. To solve the problem, we propose $\mathcal{L}_{ood}$ to prevent OOD examples from influencing prototypes and meanwhile preserve classification ability on shared classes, which is formulated as:

$$\mathcal{L}_{ood} = -\frac{1}{n_u}\sum_{i=1}^{n_u} \ln\left(\frac{\exp\left(\mathbf{z}_i^{u,weak}\cdot \mathbf{z}_i^{u,strong}/\tau\right)}{\sum_{k\neq \widehat{k}_i}^{K} m_{i,k}}\right), \tag{3}$$

where $\mathbf{z}_i^{u,strong} = R_{pro}(E(Aug_{strong}(\mathbf{u}_i)))$ with $Aug_{strong}(\cdot)$ denoting the strong data augmentation used in FixMatch [9]. In this regard, prototypes of which the indices are not $\widehat{k}_i$ will be considered as negative embeddings for $\mathbf{z}_i^{u,weak}$. The minimization of $\mathcal{L}_{ood}$ will encourage $\mathbf{z}_i^{u,weak}$ to be far from the negative embeddings, and meanwhile increase the similarity between $\mathbf{z}_i^{u,weak}$ and $\mathbf{z}_i^{u,strong}$. It is also worth noting that $\mathcal{Y}_{shared}$ is a part of $\mathcal{Y}_l$, so the minimization of $\mathcal{L}_{spl}$ in Section III-C may also encourage embeddings of shared examples to gather around corresponding prototypes narrowly. To conclude, the minimization of $\mathcal{L}_{spl}$ and $\mathcal{L}_{ood}$ encourages that the embeddings of unlabeled shared examples will be close to the corresponding prototypes, by projecting examples of the same shared class to similar embeddings. As a result, if $\mathbf{u}_i$ belongs to $\mathcal{Y}_{shared}$, the similarity between $\mathbf{z}_i^{u,weak}$ and the prototype of its ground-truth label can be enlarged by minimizing $\mathcal{L}_{spl}$ and $\mathcal{L}_{ood}$. Nevertheless, if $\mathbf{u}_i$ belongs to $\mathcal{Y}_{ood}$, the similarity between $\mathbf{z}_i^{u,weak}$ and any prototype will not be enlarged through the above-mentioned optimization, as $\mathcal{Y}_{ood} \cap \mathcal{Y}_l = \emptyset$. The embeddings of OOD examples will only be pushed away from prototypes by minimizing $\mathcal{L}_{ood}$. Consequently, the embeddings of the unlabeled shared examples will be generally more similar to the prototypes of their ground-truth labels than those of the OOD examples. Therefore, the quality of embeddings for unlabeled examples is enhanced to improve the reliability of prototypes. In summary, the supervised prototype learning strategy employs cross-entropy loss and clustering technique to construct prototypes based on labeled examples. The class-wise contrastive separation strategy focuses on selecting safe unlabeled examples to enhance the reliability of prototypes and preventing OOD examples from influencing the construction of prototypes. The two strategies can not only contribute to prototype construction, but also mitigate the occurrence of imprecise prototypes.

E. Private Assignment Suppression

Under the intersectional class-mismatch setting, there exist private examples in $\mathcal{D}_l$. As a part of labeled set, private examples can provide useful supervision information. However, as $\mathcal{Y}_{pri} \cap \mathcal{Y}_u = \emptyset$, assigning unlabeled examples to $\mathcal{Y}_{pri}$ during classifier training will lead to mistakes, and thus inhibiting the classification performance. Therefore, it is important to prevent unlabeled examples from being assigned to the private classes.

To tackle the problem, we employ class-wise similarities $\mathbf{s}$, which are computed based on prototypes in Section III-D, to alleviate the influence of private classes. As mentioned in Section III-D, safe embeddings of unlabeled examples will be selected into $\mathcal{Q}$ to improve the reliability of prototypes. Particularly, $m_{i,\widehat{k}_i}$ of the selected embedding $\mathbf{z}_i^{u,weak}$ is larger than the corresponding class-wise similarity $s_{\widehat{k}_i}$, so the involvement of the safe embedding in $\mathbf{Q}_{\widehat{k}_i}$ will help increase the class-wise similarity of the corresponding class. Nevertheless, as $\mathcal{Y}_{pri}$ does not have examples in $\mathcal{D}_u$, the class-wise similarities of $\mathcal{Y}_{pri}$ will not be increased in the above-mentioned process. As a result, the class-wise similarities of $\mathcal{Y}_{pri}$ should be smaller than those of $\mathcal{Y}_{shared}$. At the same time, the class-wise similarity of a shared class will be close to those of other shared classes when compared with those of $\mathcal{Y}_{pri}$. Accordingly, the lower the class-wise similarity is, the more the corresponding class is likely to fall into $\mathcal{Y}_{pri}$. Consequently, we define the similarity ratios $\mathbf{t}$ for all known classes to show their tendency to fall into $\mathcal{Y}_{pri}$, where the k-th element t_k for class k is formulated as

$$t_k = 1 - s_k/s_{k'}, \quad k' = \underset{k}{\operatorname{argmax}}\, s_k. \tag{4}$$

With the help of similarity ratios, we propose $\mathcal{L}_{pri}$ to alleviate the wrong assignments of unlabeled examples to private classes, which is formulated as:

$$\mathcal{L}_{pri} = -\frac{1}{n_u}\sum_{i=1}^{n_u} t_{k_i} \ln\left(1 - c_{i,k_i}^{u,weak}\right), \tag{5}$$

with

$$k_i = \underset{k}{\operatorname{argmax}}\, c_{i,k}^{u,weak}, \tag{6}$$

where $c_{i,k}^{u,weak} = R_{cls,k}(E(Aug_{weak}(\mathbf{u}_i)))$ denotes the classification probability of $Aug_{weak}(\mathbf{u}_i)$ belonging to the k-th class. Particularly, if k_i of $\mathbf{u}_i$ belongs to $\mathcal{Y}_{pri}$, the corresponding class-wise similarity s_{k_i} should be clearly lower than $s_{k'}$ and leads to a high similarity ratio t_{k_i}, so that the classifier will be discouraged from assigning $\mathbf{u}_i$ to the private class k_i. When k_i belongs to $\mathcal{Y}_{shared}$, s_{k_i} will be close to $s_{k'}$ and the corresponding similarity ratio t_{k_i} will be close to 0, which will weaken the influence on the assignment of $\mathbf{u}_i$. To conclude, the similarity ratios of private classes are clearly larger than those of shared classes, so the assignments of unlabeled examples to private classes will be alleviated by minimizing $\mathcal{L}_{pri}$. Meanwhile, the assignments of unlabeled examples to shared classes can be preserved due to the tiny similarity ratios of shared classes.

F. Weighted Semi-Supervised Learning

With reliable $\mathcal{P}$ and $\mathbf{s}$, we can properly utilize the unlabeled examples to train the classifier. Specifically, the pseudo label $\widehat{k}_i$ of each unlabeled example $\mathbf{u}_i$ can be computed through (2). As there exist OOD examples in $\mathcal{D}_u$, it is important to decrease the weights of OOD examples during training and focus on utilizing the unlabeled shared examples [17], [18], [21], [22], [23], [44]. As discussed in Section III-D, the embeddings of the unlabeled shared examples will be closer to the prototypes of their ground-truth labels than those of OOD examples, so the unlabeled examples can be reasonably weighted based on the similarities between their embeddings and prototypes. Besides, considering the difference between the clustering tendency of different known classes, the class-wise similarities should be used to calibrate the weights of the unlabeled examples. Thereby, we define weights $\mathbf{w}$ to evaluate all unlabeled examples, where the i-th element w_i for each unlabeled example $\mathbf{u}_i$ is defined as

$$w_i = \frac{m_{i,\widehat{k}_i}}{s_{\widehat{k}_i}}. \tag{7}$$

As a result, weights of the unlabeled shared examples will be generally larger than those of the OOD examples. Thereby, we may conduct semi-supervised learning under the guidance of $\mathbf{w}$. The objective function is formulated as

$$\mathcal{L}_{ssl} = \frac{1}{n_u}\sum\nolimits_{i=1}^{n_u} w_i H\left(\widehat{k}_i, \mathbf{c}_i^{u,strong}\right), \tag{8}$$

where $H(\cdot)$ denotes the cross-entropy loss and $\mathbf{c}_i^{u,strong} = R_{cls}(E(Aug_{strong}(\mathbf{u}_i)))$. Note that, w_i is calculated without using a linear prediction, and thus avoiding the overconfidence issue. Therefore, we can get accurate w_i for each unlabeled example to boost the classification performance. Finally, the overall loss of CCPL is stated as:

$$\mathcal{L} = \mathcal{L}_{ce} + \mathcal{L}_{spl} + \mathcal{L}_{ood} + \mathcal{L}_{pri} + \mathcal{L}_{ssl}. \tag{9}$$

IV. Experiments

In this section, we carry out experiments to show the effectiveness of CCPL. We first introduce the experimental settings (Section IV-A). Then, we provide the performance comparison (Section IV-B), ablation study (Section IV-C), and performance verification (Section IV-D).

A. Experimental Settings

1) Datasets: We evaluate the effectiveness of CCPL on the following five different datasets, namely: (1) **MNIST** [49] contains 60,000 training images and 10,000 testing images with the size of 28×28, belonging to 10 classes: "0"$\sim$"9". (2) **SVHN** [50] consists of 73,257 training images and 26,032 testing images with the size of 32×32, belonging to 10 classes: "0"$\sim$"9". (3) **CIFAR-10** [51] includes 60,000 training images and 10,000 testing images with the size of 32×32. The dataset contains 10 classes, which consist of six animal classes: "0"$\sim$"5", and four transportation tool classes: "6"$\sim$"9". (4) **ImageNet-100** is a subset of ImageNet [52] and has 133,116 images from ImageNet of which the size is 32×32, belonging to 100 classes: "0"$\sim$"99" [16]. (5) **Fundus** is a real-world dataset for eye disease detection based on retinal photographs [53]. Its labeled set and unlabeled set are from two public datasets, respectively, *i.e.*, TAOP[1] and ODIR.[2] Note that the eye disease images of TAOP and ODIR are collected from two different hospitals, so the types of eye diseases are different between labeled set and unlabeled set, which naturally follows the intersectional class-mismatch setting. In this dataset, the labeled set has five classes: "0"$\sim$"4", and the unlabeled set has eight classes: "1"$\sim$"8".

2) Intersectional Class-Mismatch Setting: To investigate the capability of our CCPL method in tackling the intersectional class-mismatched SSL problem, we define the intersectional class-mismatch setting for each dataset. Specifically, we fix $\mathcal{Y}_l$ and $\mathcal{Y}_{ood}$, and change the number of classes in $\mathcal{Y}_{shared}$ to evaluate the effectiveness of CCPL under different circumstances. As $\mathcal{Y}_{pri} = \mathcal{Y}_l - \mathcal{Y}_{shared}$ and $\mathcal{Y}_u = \mathcal{Y}_{ood} \cup \mathcal{Y}_{shared}$, if we define $\mathcal{Y}_{shared}$, then $\mathcal{Y}_u$ and $\mathcal{Y}_{pri}$ are known as well. For MNIST, SVHN, and CIFAR-10, they share the same setting because they all contain 10 classes. By following [18], [21], their $\mathcal{Y}_l$ and $\mathcal{Y}_{ood}$ are set as $\{0,1,2,3,4,5\}$ and $\{6,7,8,9\}$, respectively. Then, we define four cases of $\mathcal{Y}_{shared}$, namely 1) $Case\,1$: $\mathcal{Y}_{shared} = \{2,3,4,5\}$; 2) $Case\,2$: $\mathcal{Y}_{shared} = \{3,4,5\}$; 3) $Case\,3$: $\mathcal{Y}_{shared} = \{4,5\}$; and 4) $Case\,4$: $\mathcal{Y}_{shared} = \{5\}$. For ImageNet-100, as it includes 100 classes, we set its $\mathcal{Y}_l$ and $\mathcal{Y}_{ood}$ as $\{0,1,\ldots,59\}$ and $\{60,61,\ldots,99\}$, respectively. By following [16], its $\mathcal{Y}_{shared}$ is defined as: 1) $Way\,1$: $\mathcal{Y}_{shared} = \{15,16,\ldots,59\}$; 2) $Way\,2$: $\mathcal{Y}_{shared} = \{30,31,\ldots,59\}$. For Fundus, its original setting has already been made consistent with our assumption of the intersectional class-mismatch setting. By following [18]: (1) For MNIST, each class has 10 examples in $\mathcal{D}_l$, and 3,332 examples in $\mathcal{D}_u$. (2) For SVHN, each class has 100 examples in $\mathcal{D}_l$, and 3,332 examples in $\mathcal{D}_u$. (3) For CIFAR-10, each class has 400 examples in $\mathcal{D}_l$, and 3,332 examples in $\mathcal{D}_u$. (4) For ImageNet-100, each class has 100 examples in $\mathcal{D}_l$, and 1,208 examples in $\mathcal{D}_u$. (5) For Fundus, the labeled set containing 2,472 examples of five classes is from TAOP, and the unlabeled set containing 5,814 examples of eight classes is from ODIR.

3) Implementation Details: We implement the proposed framework in PyTorch [54] and train on 2 NVIDIA TITAN GPUs. For all compared methods, their classifier architectures are the same as CCPL, and we use well-tuned hyperparameters for each dataset. By following [55], embedding queue length L is set to $5\times$ size of per class labeled examples. The hyper-parameter τ is set to 0.95. More details can be found in **Supplementary Material**.

B. Performance Comparison

We compare CCPL with fourteen representative SSL methods, including PL [32], PI [29], VAT [35], Mean Teacher (MT) [36], FixMatch [9], FlexMatch [40], MTCF [23], UASD [17], DS^{3}L [18], OpenMatch [21], T2T [20], CAFA [47],

[1][Online]. Available: https://contest.taop.qq.com
[2][Online]. Available: https://odir2019.grand-challenge.org

TABLE II
CLASSIFICATION ACCURACY (%) ON MNIST

Settings	Methods	*Case* 1	*Case* 2	*Case* 3	*Case* 4
Class-matched Methods	SupCe	87.92 ± 0.11	87.92 ± 0.11	87.92 ± 0.11	87.92 ± 0.11
	SupCon [43]	88.24 ± 0.13	88.24 ± 0.13	88.24 ± 0.13	88.24 ± 0.13
	VAT [35]	94.02 ± 0.20	88.75 ± 0.29	82.17 ± 0.24	78.98 ± 0.34
	MT [36]	94.37 ± 0.16	87.09 ± 0.28	81.00 ± 0.27	78.95 ± 0.32
	PI [29]	94.09 ± 0.19	86.66 ± 0.27	78.12 ± 0.26	76.62 ± 0.29
	PL [32]	92.76 ± 0.23	86.81 ± 0.31	80.54 ± 0.33	77.77 ± 0.36
	FixMatch [9]	97.02 ± 0.16	96.15 ± 0.19	95.25 ± 0.21	92.34 ± 0.22
	FlexMatch [40]	96.30 ± 0.29	95.04 ± 0.35	93.57 ± 0.36	91.73 ± 0.40
Class-mismatched Methods	MTCF [23]	94.48 ± 0.17	89.12 ± 0.21	82.53 ± 0.27	80.83 ± 0.26
	UASD [17]	95.18 ± 0.21	90.73 ± 0.23	85.71 ± 0.29	83.48 ± 0.27
	DS^{3}L [18]	95.22 ± 0.19	89.37 ± 0.21	82.62 ± 0.22	81.23 ± 0.26
	OpenMatch [21]	97.52 ± 0.21	96.22 ± 0.24	95.29 ± 0.30	93.37 ± 0.32
	T2T [20]	97.94 ± 0.19	96.87 ± 0.22	95.63 ± 0.27	93.60 ± 0.33
	CAFA [47]	97.71 ± 0.16	96.77 ± 0.25	95.45 ± 0.26	93.39 ± 0.31
	CCSSSL [22]	97.16 ± 0.27	96.10 ± 0.27	94.68 ± 0.29	92.95 ± 0.34
	OSP [46]	97.98 ± 0.17	96.91 ± 0.22	95.72 ± 0.24	93.84 ± 0.35
Our Method	CCPL	**98.13 ± 0.18**	**97.11 ± 0.20**	**96.04 ± 0.24**	**94.25 ± 0.27**

The bold value means the best performance in comparison.

TABLE III
CLASSIFICATION ACCURACY (%) ON SVHN

Settings	Methods	*Case* 1	*Case* 2	*Case* 3	*Case* 4
Class-matched Methods	SupCe	82.93 ± 0.20	82.93 ± 0.20	82.93 ± 0.20	82.93 ± 0.20
	SupCon [43]	83.15 ± 0.19	83.15 ± 0.19	83.15 ± 0.19	83.15 ± 0.19
	VAT [35]	83.49 ± 0.37	82.38 ± 0.42	80.28 ± 0.46	79.56 ± 0.52
	MT [36]	83.12 ± 0.32	81.95 ± 0.36	81.69 ± 0.47	79.41 ± 0.49
	PI [29]	81.84 ± 0.41	79.95 ± 0.50	79.69 ± 0.49	78.95 ± 0.53
	PL [32]	82.94 ± 0.36	82.27 ± 0.43	81.18 ± 0.48	79.15 ± 0.52
	FixMatch [9]	94.98 ± 0.19	93.02 ± 0.20	91.89 ± 0.20	90.72 ± 0.22
	FlexMatch [40]	94.53 ± 0.36	92.75 ± 0.40	90.81 ± 0.41	89.05 ± 0.44
Class-mismatched Methods	MTCF [23]	85.89 ± 0.32	85.35 ± 0.42	84.73 ± 0.43	83.36 ± 0.46
	UASD [17]	85.78 ± 0.35	85.07 ± 0.36	84.42 ± 0.43	83.74 ± 0.45
	DS^{3}L [18]	85.79 ± 0.29	83.82 ± 0.38	82.89 ± 0.45	81.95 ± 0.44
	OpenMatch [21]	94.44 ± 0.27	92.88 ± 0.30	90.81 ± 0.33	89.70 ± 0.36
	T2T [20]	94.36 ± 0.30	93.13 ± 0.33	91.01 ± 0.35	89.89 ± 0.38
	CAFA [47]	94.82 ± 0.33	93.57 ± 0.35	91.25 ± 0.39	90.39 ± 0.41
	CCSSL [22]	93.72 ± 0.32	92.90 ± 0.34	90.47 ± 0.37	89.58 ± 0.41
	OSP [46]	94.54 ± 0.33	93.32 ± 0.37	91.19 ± 0.40	90.05 ± 0.46
Our Method	CCPL	**96.44 ± 0.28**	**95.19 ± 0.29**	**92.81 ± 0.28**	**92.35 ± 0.30**

The bold value means the best performance in comparison.

TABLE IV
CLASSIFICATION ACCURACY (%) ON CIFAR-10

Settings	Methods	*Case* 1	*Case* 2	*Case* 3	*Case* 4
Class-matched Methods	SupCe	69.83 ± 0.35	69.83 ± 0.35	69.83 ± 0.35	69.83 ± 0.35
	SupCon [43]	70.08 ± 0.39	70.08 ± 0.39	70.08 ± 0.39	70.08 ± 0.39
	VAT [35]	71.46 ± 0.61	70.10 ± 0.67	68.06 ± 0.58	66.10 ± 0.62
	MT [36]	70.46 ± 0.56	69.86 ± 0.69	68.42 ± 0.67	66.32 ± 0.71
	PI [29]	70.83 ± 0.45	69.42 ± 0.64	68.75 ± 0.59	66.21 ± 0.58
	PL [32]	70.66 ± 0.49	69.03 ± 0.55	68.33 ± 0.63	66.65 ± 0.66
	FixMatch [9]	86.31 ± 0.25	84.17 ± 0.25	81.91 ± 0.27	80.77 ± 0.30
	FlexMatch [40]	87.47 ± 0.20	85.37 ± 0.21	83.09 ± 0.24	81.95 ± 0.24
Class-mismatched Methods	MTCF [23]	75.16 ± 0.54	74.27 ± 0.51	73.21 ± 0.47	71.06 ± 0.48
	UASD [17]	75.01 ± 0.51	74.60 ± 0.53	74.01 ± 0.49	71.82 ± 0.57
	DS^{3}L [18]	74.76 ± 0.47	72.67 ± 0.48	71.52 ± 0.55	70.18 ± 0.56
	OpenMatch [21]	88.37 ± 0.35	86.06 ± 0.40	83.92 ± 0.40	82.41 ± 0.42
	T2T [20]	87.84 ± 0.33	85.75 ± 0.36	83.49 ± 0.37	82.28 ± 0.43
	CAFA [47]	89.52 ± 0.48	87.61 ± 0.51	85.97 ± 0.53	84.71 ± 0.58
	CCSSL [22]	87.42 ± 0.35	85.58 ± 0.38	83.46 ± 0.41	82.44 ± 0.46
	OSP [46]	86.92 ± 0.34	84.98 ± 0.39	82.72 ± 0.45	81.07 ± 0.49
Our Method	CCPL	**90.09 ± 0.36**	**88.35 ± 0.37**	**86.87 ± 0.40**	**85.82 ± 0.42**

The bold value means the best performance in comparison.

CCSSL [22] and OSP [46], in which MTCF, UASD, DS^{3}L, OpenMatch, T2T, CAFA, CCSSL and OSP are designed for SSL under the class-mismatch setting. Besides, we compare CCPL with two supervised learning baseline methods which only use $\mathcal{D}_l$. The first baseline method, namely SupCe, involves training a neural network with cross-entropy loss. The second baseline method, namely SupCon, employs supervised contrastive learning [43]. All the compared methods are trained under the consistent intersectional class-mismatch setting, and their classification accuracies on test examples belonging to $\mathcal{Y}_l$ are reported with averaged results of five runs.

The comparison results are shown in Tables II–VI. It can be found that our CCPL outperforms other compared methods on the five datasets under different settings. On four benchmarks (*i.e.*, MNIST, SVHN, CIFAR-10, and ImageNet-100), from $Case\,1$ to $Case\,4$ and $Way\,1$ to $Way\,2$, the number of classes from $\mathcal{Y}_{shared}$ decreases, and thus inducing the significant performance drop of most SSL methods. As the number of classes from $\mathcal{Y}_{ood}$ increases, several compared methods such as PI and UASD are even surpassed by SupCe and SupCon. This suggests that these methods struggle to address the negative influence of OOD examples while leveraging unlabeled data. However, our CCPL still yields promising performance, indicating that the weights of CCPL can utilize unlabeled examples effectively and mitigate the impact of OOD examples. Furthermore, as the number of private classes increases, CCPL maintains its advantage over competitors, suggesting that the proposed private assignment suppression strategy effectively mitigates the influence of private classes. In comparison to class-matched methods, our CCPL, designed for intersectional class-mismatch settings, effectively addresses OOD examples and private examples. In contrast to class-mismatched methods, CCPL avoids the negative influence of overconfidence and private classes by constructing prototypes and proposing the private assignment suppression strategy. On the real-world Fundus dataset, the accuracy of our CCPL is higher than those of others as well. The results can validate the advantage of our CCPL method over other competitors in dealing with the intersectional class-mismatched SSL problem.

C. Ablation Study

We investigate the effectiveness of three main techniques in our CCPL, namely $\mathcal{L}_{spl}$, $\mathcal{L}_{ood}$ and $\mathcal{L}_{pri}$. Fig. 3 shows the ablative results on CIFAR-10. Specifically, when the supervised prototype learning strategy or the class-wise contrastive separation strategy is not complete (see lines (2), (3), (4) and (5)), it can be seen that the accuracy will decrease, because embeddings generated through R_{pro} are not safe enough to get reliable prototypes when $\mathcal{L}_{spl}$ or $\mathcal{L}_{ood}$ is absent, and thus leading to inaccurate weights. When we discard $\mathbf{w}$ from the CCPL method and keep others fixed, the algorithm degrades to a traditional SSL method, so we can observe that line (6) suffers a large performance drop when compared with other lines. This validates the benefits brought by safe weights $\mathbf{w}$ during training, which can encourage the classifier to learn unlabeled examples well. Line (1) deserves our attention as well. It can be found that its accuracy is very close to that of CCPL in $Case\,1$ and $Case\,2$.

TABLE V
CLASSIFICATION ACCURACY (%) ON IMAGENET-100

Settings	**Methods**	$Way\ 1$	$Way\ 2$
Class-matched Methods	SupCe	40.03 ± 0.21	40.03 ± 0.21
	SupCon [43]	40.14 ± 0.25	40.14 ± 0.25
	VAT [35]	39.42 ± 0.21	37.39 ± 0.25
	MT [36]	38.66 ± 0.14	37.30 ± 0.21
	PI [29]	39.57 ± 0.19	38.33 ± 0.27
	PL [32]	39.17 ± 0.15	37.83 ± 0.19
	FixMatch [9]	52.55 ± 0.17	50.15 ± 0.23
	FlexMatch [40]	54.23 ± 0.18	53.26 ± 0.29
Class-mismatched Methods	MTCF [23]	40.86 ± 0.17	38.57 ± 0.26
	UASD [17]	40.20 ± 0.19	37.97 ± 0.28
	DS^{3}L [18]	39.91 ± 0.16	37.52 ± 0.24
	OpenMatch [21]	55.52 ± 0.34	54.01 ± 0.39
	T2T [20]	54.75 ± 0.29	53.45 ± 0.33
	CAFA [47]	57.65 ± 0.35	56.07 ± 0.42
	CCSSL [22]	55.16 ± 0.27	53.73 ± 0.40
	OSP [46]	55.82 ± 0.34	54.61 ± 0.37
Our Method	CCPL	**58.17 ± 0.32**	**56.92 ± 0.39**

The bold value means the best performance in comparison.

TABLE VI
CLASSIFICATION ACCURACY (%) ON FUNDUS

Settings	**Methods**	**Accuracy**
Class-matched Methods	SupCe	38.07 ± 0.26
	SupCon [43]	37.96 ± 0.27
	VAT [35]	40.03 ± 0.61
	MT [36]	39.25 ± 0.64
	PI [29]	39.95 ± 0.52
	PL [32]	40.51 ± 0.59
	FixMatch [9]	78.39 ± 0.41
	FlexMatch [40]	80.27 ± 0.37
Class-mismatched Methods	MTCF [23]	42.76 ± 0.51
	UASD [17]	42.48 ± 0.48
	DS^{3}L [18]	41.74 ± 0.57
	OpenMatch [21]	80.02 ± 0.55
	T2T [20]	78.63 ± 0.48
	CAFA [47]	82.43 ± 0.59
	CCSSL [22]	80.75 ± 0.56
	OSP [46]	79.89 ± 0.58
Our Method	CCPL	**83.03 ± 0.51**

The bold value means the best performance in comparison.

This is mainly due to the low proportion of private classes in $\mathcal{Y}_l$ in $Case\,1$ and $Case\,2$, leading to the small negative influence of private classes. As the private assignment suppression strategy can weaken the negative influence caused by private classes, its contribution will increase with the improving proportion of private classes in $\mathcal{Y}_l$ from $Case\,1$ to $Case\,4$, which can be confirmed by the obvious gap between the performance of line (1) and that of CCPL in $Case\,3$ and $Case\,4$.

D. Performance Verification

1) Weights $\mathbf{w}$*:* In our CCPL, we employ $\mathcal{L}_{spl}$ and $\mathcal{L}_{ood}$ to improve the reliability of generated embeddings and prototypes, which are closely connected to the safety of the weights $\mathbf{w}$. To validate the effectiveness of $\mathcal{L}_{spl}$ and $\mathcal{L}_{ood}$ in getting a safe weight w_i for each unlabeled example, we compare our CCPL on CIFAR-10 with two degraded versions: 1) we remove $\mathcal{L}_{ood}$ and keep others fixed, denoted as "w/o $\mathcal{L}_{ood}$"; and 2) we remove $\mathcal{L}_{spl}$ and keep others fixed, denoted as "w/o $\mathcal{L}_{spl}$".

Table VII shows the comparison results, containing the classification accuracy, an average weight of unlabeled shared examples w_{shared} and that of OOD examples w_{ood}. The results show

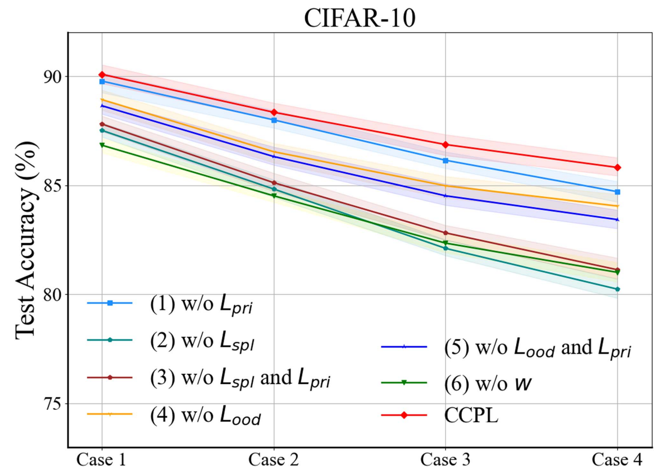


Fig. 3. Ablation study on CIFAR-10. Shaded regions indicate a standard deviation over five trails.

that the w_{shared} of CCPL is obviously larger than the w_{ood}, which indicates that unlabeled shared examples can get more attention than OOD examples, and thus leading to the highest accuracy among all competitors. Unfortunately, the w_{shared} and w_{ood} of other competitors are not satisfactory. Without $\mathcal{L}_{ood}$, the difference between w_{shared} and w_{ood} is reduced because many OOD examples are mistakenly selected into $\mathcal{Q}$, and thus affecting the reliability of $\mathcal{P}$. Without $\mathcal{L}_{spl}$, the situation becomes worse than before as w_{shared} is even smaller than w_{ood} in $Case\,3$ and $Case\,4$, because the supervision information provided by labeled examples is not exploited, inducing the terrible prototypes which mislead the class-wise contrastive separation. As a result, the accuracies of the two competitors are lower than that of CCPL. Therefore, both $\mathcal{L}_{spl}$ and $\mathcal{L}_{ood}$ are critical for obtaining the safe weights, and the effectiveness of weights $\mathbf{w}$ made by CCPL can be confirmed as well.

Fig. 4 shows the probability density curves for weights $\mathbf{w}$ of CCPL in two different training stages from $Case\,1$ to $Case\,4$. It can be seen that the distribution difference between the weights for unlabeled shared examples and those for OOD examples enlarges when the iteration proceeds in each case. Especially, in Fig. 4(b), (d), (f), and (h), we can see that the weight distribution is clearly separable between unlabeled shared examples and OOD examples, and thus indicating the ability of CCPL to obtain safe weights $\mathbf{w}$ under the intersectional class-mismatch setting.

2) Similarity Ratios $\mathbf{t}$*:* Similarity ratios $\mathbf{t}$ play an important role in (5). To investigate its effects, we plot the averaged similarity ratio t_k with respect to each class k in every case in Fig. 5. We can find that t_k is large when k belongs to $\mathcal{Y}_{pri}$, and small when k is in $\mathcal{Y}_{shared}$. Thereby, our private assignment suppression can help mitigate the negative influence caused by $\mathcal{Y}_{pri}$ under the intersectional class-mismatch setting.

3) Feature Visualization: Fig. 6 visualizes the image features made by classifiers trained in $Case\,4$ of CIFAR-10 through FixMatch and CCPL, respectively. It can be observed that examples of $\mathcal{Y}_l$ can be clustered better by CCPL than FixMatch, and the features of $\mathcal{Y}_{shared}$ are separated well from those of other known

TABLE VII
EFFECTS OF $\mathbf{w}$ ON CLASSIFICATION ACCURACY (%) ON CIFAR-10

Methods	$Case$ 1			$Case$ 2			$Case$ 3			$Case$ 4		
	Accuracy	w_{shared}	w_{ood}	Accuracy	w_{shared}	w_{ood}	Accuracy	w_{shared}	w_{ood}	Accuracy	w_{shared}	w_{ood}
CCPL	90.09 ± 0.36	99.01	68.33	88.35 ± 0.37	97.62	68.41	86.87 ± 0.40	96.83	68.91	85.82 ± 0.42	95.48	69.47
w/o $\mathcal{L}_{ood}$	88.89 ± 0.39	98.53	82.04	86.55 ± 0.43	96.15	83.26	84.98 ± 0.45	93.37	83.59	83.95 ± 0.50	91.53	84.67
w/o $\mathcal{L}_{spl}$	87.71 ± 0.40	90.41	85.03	84.83 ± 0.45	87.76	86.25	82.11 ± 0.46	84.40	87.12	80.41 ± 0.48	82.28	87.95

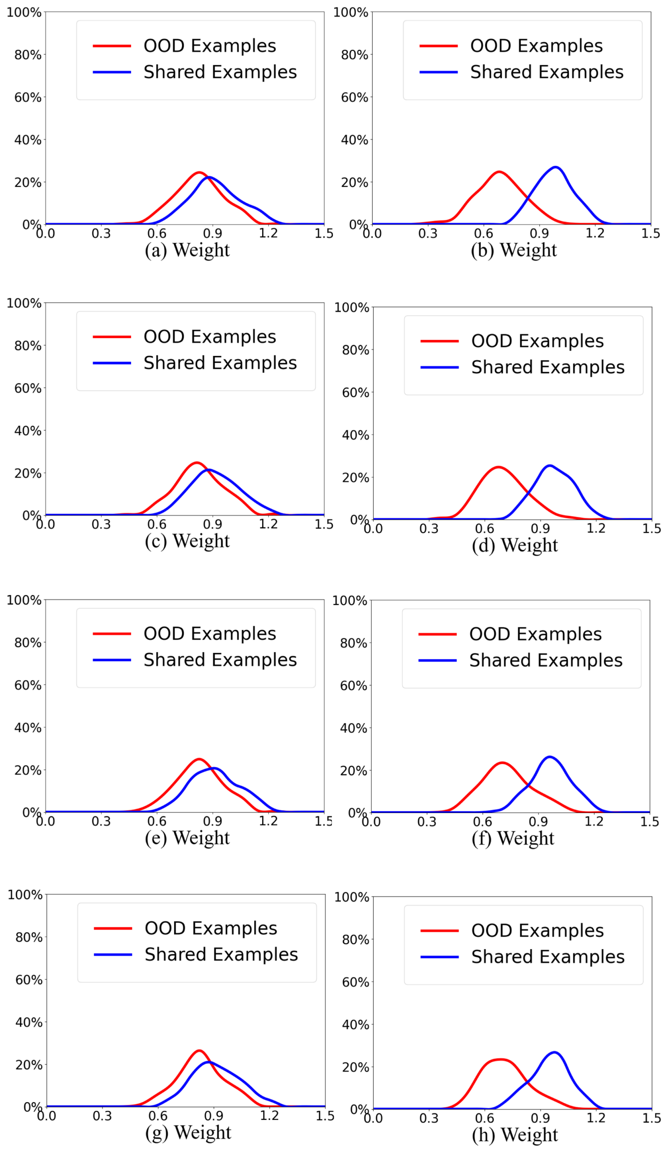


Fig. 4. Probability density curves of weights $\mathbf{w}$ on CIFAR-10 from $Case$ 1 to $Case$ 4 in two stages: (a) The $5,000^{th}$ iteration in $Case$ 1; (b) The $15,000^{th}$ iteration in $Case$ 1; (c) The $5,000^{th}$ iteration in $Case$ 2; (d) The $15,000^{th}$ iteration in $Case$ 2; (e) The $5,000^{th}$ iteration in $Case$ 3; (f) The $15,000^{th}$ iteration in $Case$ 3; (g) The $5,000^{th}$ iteration in $Case$ 4; (h) The $15,000^{th}$ iteration in $Case$ 4.

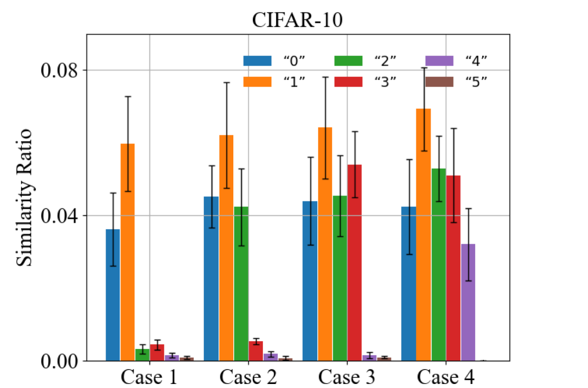


Fig. 5. Similarity ratio of each known class from $Case$ 1 to $Case$ 4 on CIFAR-10.

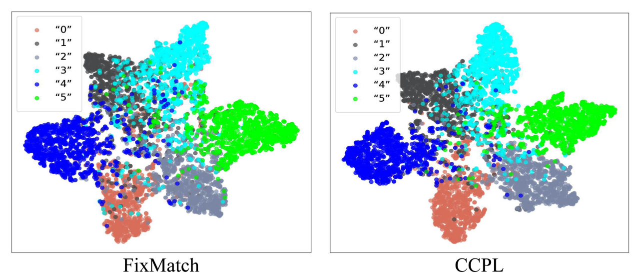


Fig. 6. Feature visualization of t-SNE [56] for FixMatch (left) and CCPL (right) with classifiers trained in $Case$ 4 on CIFAR-10.

classes by CCPL, indicating that CCPL can exploit both labeled and unlabeled examples to train an effective classifier under the intersectional class-mismatch setting.

4) Efficiency Evaluation: We compare the efficiency of our CCPL with those of other competitors using two metrics, namely: 1) the amount of parameters in neural networks, denoted as "#Param", and 2) the average training time of each iteration, denoted as "Training Time". The proposed CCPL is implemented with PyTorch, and is trained on two NVIDIA 2080Ti GPUs. WideResNet 28-2 [57] is chosen as the backbone of classifier for all compared methods. Besides, the batch size is set to 100. As reported in Table VIII, the number of training parameters required by CCPL is comparable to that of most existing SSL methods while it needs a relatively longer training time. However, the classification accuracy of our CCPL is significantly superior to that of existing methods as revealed by the results in Tables II–VI.

V. CONCLUSION

In this article, we present a novel approach dubbed Class-wise Contrastive Prototype Learning (CCPL) to solve the intersectional class-mismatched SSL problem. Particularly, we construct reliable prototypes by utilizing training examples. Then we employ private assignment suppression to reduce the improper classification caused by the private classes. Finally, we conduct semi-supervised training to learn consistent predictions

TABLE VIII
EFFICIENCY STUDY OF ALL COMPARED METHODS ON CIFAR-10

Methods	#Param (M)	Training Time (s)
VAT [35]	1.47	0.154
MT [36]	2.94	0.119
PI [29]	1.47	0.105
PL [32]	1.47	0.108
FixMatch [9]	1.47	0.157
FlexMatch [40]	1.47	0.149
MTCF [23]	1.47	0.158
UASD [17]	1.47	0.196
DS^3L [18]	2.93	0.424
OpenMatch [21]	1.50	0.376
T2T [20]	1.50	0.892
CAFA [47]	1.47	0.804
CCSSL [22]	1.49	0.762
OSP [46]	1.50	0.947
CCPL(ours)	1.49	0.776

between different augmented versions of each unlabeled example under the guidance of weights, where the weights are computed based on the prototypes, so the impacts of OOD examples can be weakened. Experimental results show that our CCPL outperforms fourteen representative SSL methods as well as two supervised learning methods under the intersectional class-mismatch setting.

Our CCPL is proposed to reduce the influences caused by OOD classes and private classes under intersectional class-mismatch setting. However, its advantage diminishes in a class-matched setting when there are no OOD classes and private classes. Therefore, future efforts will focus on enhancing the performance of CCPL in such settings to broaden its applicability.

REFERENCES

[1] Y. Grandvalet and Y. Bengio, "Semi-supervised learning by entropy minimization," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2005, pp. 281–296.

[2] H. Zhu et al., "Enhancing semi-supervised learning with cross-modal knowledge," in *Proc. 30th ACM Int. Conf. Multimedia*, 2022, pp. 4456–4465.

[3] W. Wang, L. Lin, Z. Fan, and J. Liu, "Semi-supervised learning for mars imagery classification and segmentation," *ACM Trans. Multimedia Comput., Commun. Appl.*, vol. 19, no. 4, pp. 1–23, 2023.

[4] Y. Zhang et al., "Semi-supervised contrastive learning with similarity co-calibration," *IEEE Trans. Multimedia*, vol. 25, pp. 1749–1759, 2023.

[5] B. Zhu and Y. Liu, "General approximate cross validation for model selection: Supervised, semi-supervised and pairwise learning," in *Proc. 29th ACM Int. Conf. Multimedia*, 2021, pp. 5281–5289.

[6] J. Zhao, X. Liu, and W. Zhao, "Balanced and accurate pseudo-labels for semi-supervised image classification," *ACM Trans. Multimedia Comput., Commun. Appl.*, vol. 18, no. 3s, pp. 1–18, 2022.

[7] S. Fu et al., "Dynamic graph learning convolutional networks for semi-supervised classification," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 17, no. 1s, pp. 1–13, 2021.

[8] Y. Li, Z. Zhao, H. Sun, Y. Cen, and Z. He, "Snowball: Iterative model evolution and confident sample discovery for semi-supervised learning on very small labeled datasets," *IEEE Trans. Multimedia*, vol. 23, pp. 1354–1366, 2021.

[9] K. Sohn et al., "Fixmatch: Simplifying semi-supervised learning with consistency and confidence," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2020, pp. 596–608.

[10] D. Berthelot et al., "Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring," in *Proc. Int. Conf. Learn. Representations*, 2019.

[11] T. Sosea and C. Caragea, "MarginMatch: Improving semi-supervised learning with pseudo-margins," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 15773–15782.

[12] I. Nassar, M. Hayat, E. Abbasnejad, H. Rezatofighi, and G. Haffari, "ProtoCon: Pseudo-label refinement via online clustering and prototypical consistency for efficient semi-supervised learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 11641–11650.

[13] Y. Chen et al., "Boosting semi-supervised learning by exploiting all unlabeled data," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 7548–7557.

[14] A. Oliver, A. Odena, C. Raffel, E. D. Cubuk, and I. J. Goodfellow, "Realistic evaluation of deep semi-supervised learning algorithms," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2018, pp. 3239–3250.

[15] J. Yang, K. Zhou, Y. Li, and Z. Liu, "Generalized out-of-distribution detection: A survey," 2021, *arXiv:2110.11334*.

[16] 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, 2023.

[17] Y. Chen, X. Zhu, W. Li, and S. Gong, "Semi-supervised learning under class distribution mismatch," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 3569–3576.

[18] 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.

[19] R. He, Z. Han, Y. Yang, and Y. Yin, "Not all parameters should be treated equally: Deep safe semi-supervised learning under class distribution mismatch," in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 6874–6883.

[20] J. Huang et al., "Trash to treasure: Harvesting OOD data with cross-modal matching for open-set semi-supervised learning," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 8290–8299.

[21] K. Saito, D. Kim, and K. Saenko, "Openmatch: Open-set consistency regularization for semi-supervised learning with outliers," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2021, pp. 25956–25967.

[22] F. Yang et al., "Class-aware contrastive semi-supervised learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 14401–14410.

[23] Q. Yu, D. Ikami, G. Irie, and K. Aizawa, "Multi-task curriculum framework for open-set semi-supervised learning," in *Proc. 16th Eur. Conf. Comput. Vis.*, 2020, pp. 438–454.

[24] K. Cao, M. Brbic, and J. Leskovec, "Open-world semi-supervised learning," in *Proc. Int. Conf. Learn. Representations*, 2022.

[25] L. Z. Guo, Y. G. Zhang, Z. F. Wu, J. J. Shao, and Y. F. Li, "Robust semi-supervised learning when not all classes have labels," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2022, pp. 3305–3317.

[26] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 427–436.

[27] D. Amodei et al., "Concrete problems in AI safety," 2016, *arXiv:1606.06565*.

[28] 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.*, 2022, pp. 9776–9786.

[29] S. Laine and T. Aila, "Temporal ensembling for semi-supervised learning," in *Proc. Int. Conf. Learn. Representations*, 2016.

[30] P. Cascante-Bonilla, F. Tan, Y. Qi, and V. Ordonez, "Curriculum labeling: Revisiting pseudo-labeling for semi-supervised learning," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 6912–6920.

[31] A. Iscen, G. Tolias, Y. Avrithis, and O. Chum, "Label propagation for deep semi-supervised learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5065–5074.

[32] D.-H. Lee, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 896–901.

[33] M. N. Rizve, K. Duarte, Y. S. Rawat, and M. Shah, "In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning," in *Proc. Int. Conf. Learn. Representations*, 2021.

[34] I. Triguero, S. García, and F. Herrera, "Self-labeled techniques for semi-supervised learning: Taxonomy, software and empirical study," *Knowl. Inf. Syst.*, vol. 42, pp. 245–284, 2015.

[35] T. Miyato, S.-I. Maeda, M. Koyama, and S. Ishii, "Virtual adversarial training: A regularization method for supervised and semi-supervised learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 8, pp. 1979–1993, Aug. 2019.

[36] 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. Learn. Representations*, 2017.

[37] D. Berthelot et al., "MixMatch: A holistic approach to semi-supervised learning," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2019, pp. 5050–5060.

[38] J. Li, C. Xiong, and S. C. Hoi, "CoMatch: Semi-supervised learning with contrastive graph regularization," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 9455–9464.

[39] Q. Xie, Z. Dai, E. Hovy, T. Luong, and Q. Le, "Unsupervised data augmentation for consistency training," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2020, pp. 6256–6268.

[40] B. Zhang et al., "FlexMatch: Boosting semi-supervised learning with curriculum pseudo labeling," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2021, pp. 18408–18419.

[41] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proc. Int. Conf. Mach. Learn.*, 2009, pp. 41–48.

[42] C. Gong et al., "Multi-modal curriculum learning for semi-supervised image classification," *IEEE Trans. Image Process.*, vol. 25, no. 7, pp. 3249–3260, Jul. 2016.

[43] P. Khosla et al., "Supervised contrastive learning," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2020, pp. 18661–18673.

[44] R. He, Z. Han, X. Lu, and Y. Yin, "Safe-student for safe deep semi-supervised learning with unseen-class unlabeled data," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 14565–14574.

[45] W. Liu, X. Wang, J. Owens, and Y. Li, "Energy-based out-of-distribution detection," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2020, pp. 21464–21475.

[46] Y. Wang et al., "Out-of-distributed semantic pruning for robust semi-supervised learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 23849–23858.

[47] Z. Huang, C. Xue, B. Han, J. Yang, and C. Gong, "Universal semi-supervised learning," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2021, pp. 26714–26725.

[48] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1597–1607.

[49] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[50] Y. Netzer et al., "Reading digits in natural images with unsupervised feature learning," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2011, pp. 7–15.

[51] A. Krizhevsky et al., "Learning multiple layers of features from tiny images," 2009.

[52] P. Chrabaszcz, I. Loshchilov, and F. Hutter, "A downsampled variant of imagenet as an alternative to the cifar datasets," 2017, *arXiv:1707.08819*.

[53] Y. Zhou, S. Bai, T. Zhou, Y. Zhang, and H. Fu, "Delving into local features for open-set domain adaptation in fundus image analysis," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Interv.*, 2022, pp. 682–692.

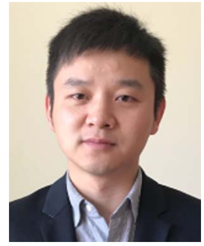
[54] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2019, pp. 8026–8037.

[55] T. Han, J. Gao, Y. Yuan, and Q. Wang, "Unsupervised semantic aggregation and deformable template matching for semi-supervised learning," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2020, pp. 9972–9982.

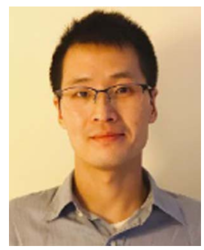
[56] L. V. d. Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2579–2605, 2008.

[57] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *Proc. Brit. Mach. Vis. Conf.*, 2016.

Mingyu Li will receive the master's degree from the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China, in 2024. His research interests include machine learning and computer vision.



Tao Zhou received the Ph.D. degree from Shanghai Jiao Tong University, Shanghai, China, in 2016. He is currently a Professor with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China. He was a Postdoctoral Fellow with UNC-CH and a Research Scientist with IIAI. He is also an Associate Editor for IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS. His research interests include machine learning, computer vision, AI in healthcare, and medical image analysis.



Bo Han (Senior Member, IEEE) is currently an Assistant Professor of machine learning with Hong Kong Baptist University, Hong Kong, and a BAIHO Visiting Scientist with RIKEN AIP, where his research focuses on machine learning, deep learning, foundation models, and their applications. He was a Visiting Faculty Researcher with Microsoft Research and a Postdoc Fellow with RIKEN AIP. He has coauthored two machine learning monographs by MIT Press and Springer Nature. He was Area Chairs of NeurIPS, ICML, ICLR, UAI, and AISTATS. He was also Action Editors and Editorial Board Members of *Journal of Machine Learning Research*, *Madras Law Journal*, *Transactions on Machine Learning Research*, Journal of Artificial Intelligence Research, and IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS. He was the recipient of the Outstanding Paper Award at NeurIPS, Notable Area Chair at NeurIPS, Outstanding Area Chair at ICLR, and Outstanding Associate Editor at IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS.

Tongliang Liu is currently an Associate Professor with the School of Computer Science and The Director of Sydney AI Centre, University of Sydney, Camperdown, NSW, Australia. He has authored and coauthored more than 200 research articles, including ICML, NeurIPS, ICLR, CVPR, ICCV, ECCV, AAAI, IJCAI, JMLR, and TPAMI. His research interests include trustworthy machine learning and its interdisciplinary applications, with a particular emphasis on learning with noisy labels, adversarial learning, causal representation learning, transfer learning, unsupervised learning, and statistical deep learning theory. He is/was a (senior-) meta Reviewer of many conferences, such as ICML, NeurIPS, ICLR, UAI, AAAI, IJCAI, and KDD, and was a notable AC for NeurIPS and ICLR. He is a co-Editor-in-Chief for Neural Networks, an Associate Editor for *Transactions on Machine Learning Research* and *ACM Computing Surveys*, and is on the Editorial Boards of JMLR and MLJ. He was the recipient of the CORE Award for Outstanding Research Contribution in 2024, IEEE AI's 10 to Watch Award in 2023, Future Fellowship Award from Australian Research Council (ARC) in 2022, Top-40 Early Achievers by The Australian in 2020, and Discovery Early Career Researcher Award (DECRA) from ARC in 2018.

Xinkai Liang is currently working toward the Ph.D. degree with the Department of Astronautics, Harbin Institute of Technology, Harbin, China, under Dr. Bin Zhou. He is currently a Senior Engineer with Science and Technology on Complex System Control and Intelligent Agent Cooperation Laboratory, China. His research interests include deep learning, image processing, and SLAM.

Jiajia Zhao received the Ph.D. degree from the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China, in 2011. He is currently the Principal investigator with Science and Technology on Complex System Control and Intelligent Agent Cooperation Laboratory, Beijing, China. His research interests include deep learning, image processing, and SLAM.

Chen Gong (Senior Member, IEEE) received the dual doctoral degree from Shanghai Jiao Tong University, Shanghai, China, and University of Technology Sydney, Ultimo, NSW, Australia. He is currently a Full Professor with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China. He has authored or coauthored more than 130 technical papers at prominent journals and conferences, such as *Journal of Machine Learning Research*, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON CYBERNETICS, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, IEEE TRANSACTIONS ON MULTIMEDIA, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, *ACM Transactions on Intelligent Systems and Technology*, ICML, NeurIPS, ICLR, CVPR, ICCV, AAAI, IJCAI, and ICDM. His research interests mainly include machine learning, data mining, and learning-based vision problems. He is an Associate Editor for IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, *Neural Networks*, and NePL, and also the Area Chair or Senior PC member of several top-tier conferences, such as AAAI, IJCAI, ECML, AISTATS, and ACM MM. He won the “Excellent Doctorial Dissertation Award” of Chinese Association for Artificial Intelligence, “Young Elite Scientists Sponsorship Program” of China Association for Science and Technology, “Wu Wen-Jun AI Excellent Youth Scholar Award”, and the Scientific Fund for Distinguished Young Scholars of Jiangsu Province. 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.