Dynamic Weighted Adversarial Learning for Semi-Supervised Classification under Intersectional Class Mismatch

MINGYU LI, Nanjing University of Science and Technology, China
TAO ZHOU, Nanjing University of Science and Technology, China
ZHUO HUANG, The University of Sydney, Australia
JIAN YANG, Nanjing University of Science and Technology, China
JIE YANG, Shanghai Jiao Tong University, China
CHEN GONG, Nanjing University of Science and Technology, China

Nowadays, class-mismatch problem has drawn intensive attention in Semi-Supervised Learning (SSL), where the classes of labeled data are assumed to be only a subset of the classes of unlabeled data. However, in a more realistic scenario, the labeled data and unlabeled data often share some common classes while they also have their individual classes, which leads to an "intersectional class-mismatch" problem. As a result, existing SSL methods are often confused by these individual classes and suffer from performance degradation. To address this problem, we propose a novel Dynamic Weighted Adversarial Learning (DWAL) framework to properly utilize unlabeled data for boosting the SSL performance. Specifically, to handle the influence of the individual classes in unlabeled data (*i.e.*, Out-Of-Distribution classes), we propose an enhanced adversarial domain adaptation to dynamically assign weight for each unlabeled example from the perspectives of domain adaptation and a class-wise weighting mechanism, which consists of transferability score and prediction confidence value. Besides, to handle the influence of the individual classes in labeled data (*i.e.*, private classes), we propose a dissimilarity maximization strategy to suppress the inaccurate correlations caused by the examples of individual classes within labeled data. Therefore, our DWAL can properly make use of unlabeled data to acquire an accurate SSL classifier under intersectional class-mismatch setting, and extensive experimental results on five public datasets demonstrate the effectiveness of the proposed model over other state-of-the-art SSL methods.

CCS Concepts: • **Computing methodologies** → **Semi-supervised learning settings**; *Image representations*; Neural networks.

Additional Key Words and Phrases: Semi-supervised learning, intersectional class mismatch, adversarial domain adaptation, dissimilarity maximization

This research is supported by NSF of China (Nos: 62336003, 12371510, 62172228, 62376153), NSF of Jiangsu Province (No: BZ2021013), NSF for Distinguished Young Scholar of Jiangsu Province (No: BK20220080), the Fundamental Research Funds for the Central Universities (Nos: 30920032202, 30921013114), CAAI-Huawei MindSpore Open Fund, and "111" Program (No: B13022).

Corresponding authors: Chen Gong (chen.gong@njust.edu.cn) and Tao Zhou (taozhou.dreams@gmail.com).
Authors' addresses: Mingyu Li, Nanjing University of Science and Technology, Nanjing, Jiangsu, China; Tao Zhou, Nanjing University of Science and Technology, Nanjing, Jiangsu, China; Zhuo Huang, The University of Sydney, Sydney, Australia; Jian Yang, Nanjing University of Science and Technology, Nanjing, Jiangsu, China; Jie Yang, Shanghai Jiao Tong University, Shanghai, China; Chen Gong, Nanjing University of Science and Technology, Nanjing, Jiangsu, China.

ACM 1551-6857/2023/12-ART
https://doi.org/10.1145/3635310

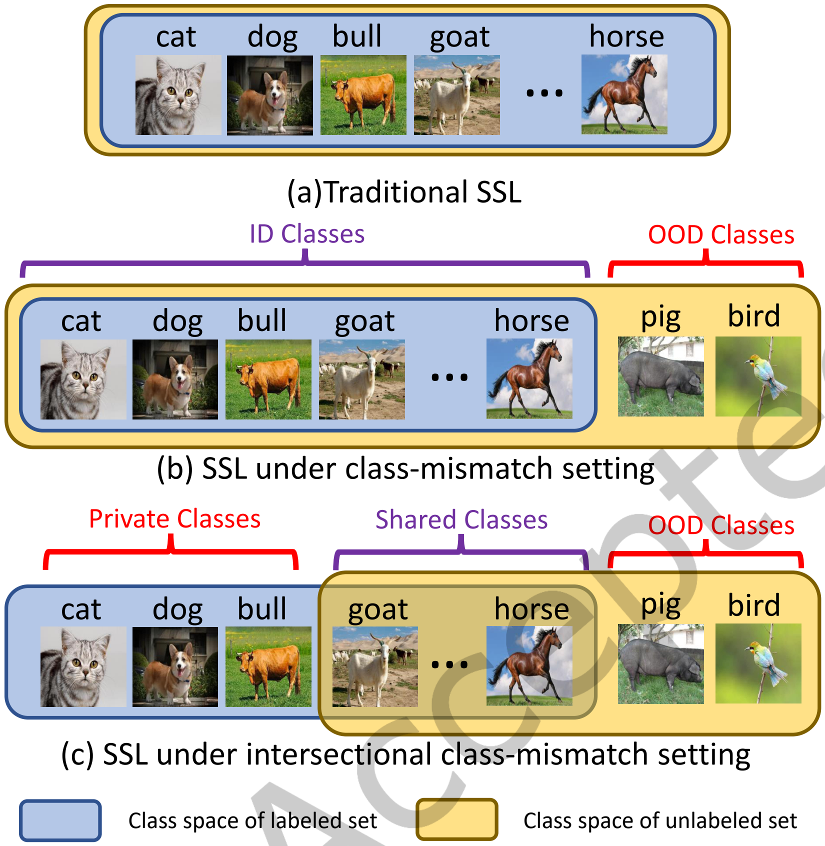

Fig. 1. Illustration of different SSL settings. (a) In traditional SSL, the labeled and unlabeled sets share the same class space. (b) In SSL with traditional class mismatch, the class space of the labeled set is a subset of that of the unlabeled set. (c) In SSL with intersectional class mismatch, both labeled and unlabeled sets contain the classes that the other does not have. Some relevant notions such as "ID Classes", "OOD Classes", "Shared Classes" and "Private Classes" are annotated in the figure.

1 INTRODUCTION

Mannually labeling lots of data for training a supervised machine learning or computer vision algorithm is often prohibitive due to the unaffordable human or monetary costs, so Semi-Supervised Learning (SSL) is proposed as a useful way to solve the labeling data shortage problem [7]. SSL aims to effectively use scarce labeled data and abundant unlabeled data to train an accurate classifier, which is capable of classifying new target examples with known classes.

Semi-Supervised Learning (SSL) has received increasing attention over the past decades as it can use massive unlabeled data to improve model performance when the labeled data is scarce [2, 3, 14, 22, 23, 26–28, 38, 49, 52, 65, 66, 68, 69]. Nowadays, the study on SSL has achieved significant progress because of the application of deep neural networks which is good at representing data and discovering data structure. There are usually three typical strategies to train deep semi-supervised learning classifiers, namely entropy minimization [22, 36], consistency regularization [35, 40, 45, 48, 50, 51], and data augmentation [18, 42, 49, 62].

However, the SSL approaches mentioned above are based on an assumption that labeled and unlabeled sets share the same class space (*i.e.*, each unlabeled example must belong to one of the known classes), as shown in Fig. 1(a). Mathematically, they assume that $C_l = C_u$, where C_l and C_u denote the class spaces of labeled and unlabeled sets, respectively. However, it is improper, as in reality, C_u is usually unknown. Under many circumstances, there exists $C_l \cap C_u = C_l$, which leads to a "class mismatch" problem [44] (see Fig. 1(b)). In this situation, existing state-of-the-art SSL algorithms often suffer performance degradation significantly.

In order to deal with the traditional class-mismatch problem of SSL, some works focus on leveraging *In-Distribution* (ID) data while trying to weaken the negative impacts caused by the *Out-Of-Distribution* (OOD) data. Here the ID data refer to the unlabeled data of which the ground-truth labels belong to the *ID Classes*, and OOD data denote the unlabeled data of which the ground-truth labels belong to the *OOD Classes* (see Fig. 1(b)). Several representative works, *e.g.*, Uncertainty-Aware Self-Distillation (UASD) [8], Safe Deep Semi-Supervised Learning (DS^{3}L) [24], Multi-part Curriculum (MTC) [61], OpenMatch (OM) [47], Trash to Treasure (T2T) [25],Class-aware Contrastive SSL (CCSSL) [58] and Out-of-distributed Semantic Pruning (OSP) [54], mainly study the case where C_l is a subset of C_u.

However, in reality, C_l may not always be a subset of C_u. For example, when we plan to take pictures of some interested animals in the wild for ecological research via using the autonomous camera, it is quite possible that some of our interested animals cannot be acquired due to the appearing occasionality of some animals. On the other hand, the camera may also unintentionally capture some species that are out of our interests because our interested species may not cover all the wild native species. For another example, to build a computer-aided medical diagnosis system, we have a limited amount of labeled data with a variety of known diseases from one hospital. Then, we need to collect a large amount of unlabeled data, which is often from another hospital and may only contain partially known diseases as well as many unseen diseases. The diseases of labeled data cannot be totally covered by unlabeled data because some of them may be rare and are not likely to appear in the collected unlabeled data.

Due to the absence of prior knowledge about C_u, the assumption of traditional class-mismatched SSL methods may be violated as well. In other words, C_u and C_l may share some common classes while also having unknown individual classes. Therefore, there is a case of $C_u \cap C_l \neq \emptyset$, $C_l - (C_l \cap C_u) \neq \emptyset$, and $C_u - (C_l \cap C_u) \neq \emptyset$, which leads to an "intersectional class-mismatch" problem (see Fig. 1(c)). Thereby, in this work, we consider such a semi-supervised learning scenario under intersectional class-mismatch setting, which is as realistic as the existing traditional class-mismatch setting [8, 24, 25, 44, 47, 61], but is more challenging than existing traditional class-mismatch setting. The goal of our work is to train an accurate SSL classifier capable of classifying new target examples of known classes under intersectional class-mismatch setting.

Under intersectional class-mismatch setting (see Fig. 1(c)), existing traditional class-mismatch SSL methods face two problems, namely: 1) The classification of ID data (*i.e.*, the unlabeled data belonging to *Shared Classes*, see Fig. 1(c)), denoted as "classification part", and the detection of OOD data (*i.e.*, the unlabeled data belonging to *OOD Classes*, see Fig. 1(c)), denoted as "detection part", often share the same network. These two parts have a conflicting optimization during feature learning, as the classification part aims to discriminate ID data from different classes while the detection part treats ID data as a whole to distinguish OOD data from ID data [25]. 2) The existence of *Private Classes* (*i.e.*, $C_l - (C_l \cap C_u)$) (see Fig. 1(c)) has been ignored and its negative influence on the detection of OOD data and classification of ID data has not been taken into consideration, which could lead to performance degradation on the test set. For SSL, making use of a large amount of unlabeled data can benefit the classifier. That is to say, to train a reliable SSL classifier under intersectional class-mismatch setting, it is important to inhibit the disturbance of OOD data and emphasize the value of ID data. As a result, three problems need to be solved, namely: 1) Harmful OOD data contain useless information for a classifier training, and it is critical to find a way to decrease their negative influence on training; 2) If OOD data detection and ID data classification share the same network, it will lead to a conflicting optimization, and increase the difficulty of training; 3) Private data could affect the normal correlations between the labeled examples and unlabeled ones, as private classes do not have corresponding examples in unlabeled set. Therefore, it is critical to detect private data, and then maximize the disagreement between private data and unlabeled data.

To this end, we propose a novel Dynamic Weighted Adversarial Learning (DWAL) approach for semi-supervised learning classification under intersectional class-mismatch setting. Specifically, to solve the first problem mentioned above, we propose to conduct enhanced adversarial domain adaptation to assign a weight for each example. The weights of unlabeled examples, which consist of transferability scores and prediction confidence value, can help highlight the ID examples that carry useful information for model training, and meanwhile suppress the adverse impact of OOD examples that may confuse the training process. Besides, the weights of labeled examples are interpreted as transferability scores, which help distinguish private data from labeled data of shared classes. Further, to tackle the second problem, we separately conduct OOD data detection and ID data classification, which makes these two parts not affect each other. To handle the third problem, we propose a dissimilarity maximization strategy to maximize the disagreement between the feature distribution of private data and that of unlabeled data, so that unlabeled data can be well adapted and assigned accurate weights. Finally, we conduct intensive experiments on benchmark datasets to demonstrate the effectiveness of our DWAL against other state-of-the-art SSL methods under intersectional class-mismatch setting.

The remaining of the article is organized as follows. In Section 2, related works are briefly reviewed. We provide the details of our DWAL in Section 4. In Section 5, we report the experimental results of our method and other compared methods. Finally, the conclusion for this work is provided in Section 6.

2 RELATED WORK

In this section, we present a brief overview of the two types of prior works that are most related to the proposed intersectional class-mismatched semi-supervised learning approach, including traditional semi-supervised learning and traditional class-mismatched semi-supervised learning. In addition, the related works on domain adaptation are discussed as well.

2.1 Traditional Semi-Supervised Learning

Various SSL methods have been developed over the past years. These algorithms can be roughly divided into the following categories: GAN-based methods, self-training methods, ensemble-based methods, graph-based methods, and clustering-based methods.

GAN-based methods utilize unlabeled data by reforming the loss of the discriminator, or employing extra discriminators and generators. For example, Dai *et al.* [12] employ a bad generator that does not match the true data distribution but simply plays the role of a complement generator to help improve the classification ability of the discriminator. Triple-GAN [37] is proposed to achieve a good generation of realistically-looking examples conditioned on class labels, which can reduce the possible prediction error of the discriminator. MarginGAN [13] yields large-margin examples to train a classifier to increase the margin of real examples and to decrease the margin of fake examples. Kumar *et al.* [34] propose to estimate the tangent space to the data manifold using GANs, and employ it to inject invariances into the classifier. Relaxed Spatial Structural Alignment [56] calibrates the target generative models during the adaptation with a cross-domain spatial structural consistency loss, where the loss comprises of a self-correlation part and a disturbance correlation consistency part.

Self-training methods aim to learn a classifier by assigning pseudo-labels to the unlabeled data without a solid threshold. For instance, Noisy Student [57] trains a student model on the combination of labeled and pseudo-labeled examples, and then treats the student model as a teacher model to enhance the quality of pseudo-label. The work [6] applies curriculum learning to pseudo-labeling so as to train a model with clear labels before considering more complex ones.

Ensemble-based methods combine several independent models to obtain promising generalization performance. For example, the work [67] generates three subsets from the original set with Bootstrap sampling mechanism to mine more information from the data. Temporal Ensembling [35] employs Exponential Moving Average (EMA), and aggregates all the previous predictions to yield promising performance. Dual Student [31] trains two student models with different initialization simultaneously to provide targets for each other, in order to improve the quality of pseudo labels.

Graph-based methods aim to exploit connectivity relationships between labeled and unlabeled examples to classify the examples of unknown classes. For instance, Graph cut algorithms [30] conduct SSL by finding a graph partition to classify all the examples. Fick's Law Assisted Propagation (FLAP) [16] employs the theory of Fick's First Law of Diffusion to distribute eigenvalues of the iteration matrix regularly.

Clustering-based methods use a clustering algorithm to process unlabeled data with a small amount of prior information. Specifically, Transductive Support Vector Machines (TSVMs) [29] adapt traditional Support Vector Machines (SVMs) to the transductive learning setting, and incorporate the hat loss instead of the hinge loss commonly adopted by SVMs. Semi-Supervised Classification based on Class Membership (SSCCM) [53] introduces the membership vector to develop the classification ability of a semi-supervised classifier.

With the employment of deep neural network, three typical strategies are developed to train classifier of SSL, namely entropy minimization, consistency regularization and data augmentation.

Entropy minimization enforces the networks to make confident predictions on unlabeled data by minimizing the label prediction entropy. For example, Pseudo-Labeling [36] picks up the class label with the highest probability, and then trains the classifier with all data in a supervised way. Uncertainty-aware Pseudo-label Selection (UPS) [46] utilizes prediction uncertainty to improve the quality of pseudo label of every unlabeled data.

The methodologies of consistency-based methods enforce that the perturbations on unlabeled data should not significantly change their label predictions. For example, Π-model [35] encourages consistent network output between two different perturbations of the same input data. Mean Teacher [51] forms an improved teacher model from the student model by averaging model parameters instead of model predictions. After that, VAT [40] computes adversarial perturbations that can maximally change the unlabeled data to confuse the classifier. Regularization framework based on Adversarial Transformations (RAT) [50] regularizes the smoothness of output distribution by utilizing adversarial transformations. Batch Nuclear-norm Maximization [11] is proposed to achieve good discriminability and diversity under label insufficient learning situations, thanks to the utilization of Frobenius norm and the rank of batch output matrix.

Recently, data augmentation becomes popular in many traditional SSL methods. It is usually combined with other traditional SSL methods, including entropy minimization and consistency regularization. For example, FixMatch [49] encourages the predicted labels of weakly augmented images and strongly augmented images to be consistent. AlphaMatch [18] uses alpha-divergence and an optimization-based framework to get effective and stable consistency regularization based on data augmentation. FlexMatch [62] sets thresholds according to the learning status and learning difficulties of different classes with Curriculum Pseudo Labeling (CPL), to get reliable training examples. SemCo [42] leverages label semantics and co-training to improve the quality of pseudo-labeling, and then weighs the training examples reasonably. However, due to the assumption that the unlabeled data should have the same class space as the labeled data, traditional SSL methods cannot deal with the real-world class-mismatch problem.

2.2 Traditional Class-Mismatched Semi-Supervised Learning

SSL methods under traditional class-mismatch setting consider the situation when the class spaces of labeled data and unlabeled data are different. This problem is raised by [35] and [44]. Inspired by their works, some new methods have been proposed to address this problem. For example, Uncertainty Aware Self-Distillation

(UASD) [8] averages the historical predictions of a self-distillation method to find the probable OOD data. Safe Deep Semi-Supervised Learning (DS^{3}L) [24] develops an instance weighting strategy to weaken the impact of unlabeled data with unseen classes to prevent performance degradation. Besides, Multi-part Curriculum learning Framework (MTCF) [61] proposes to distinguish ID data and OOD data in an ordered sequence from simple to difficult, and employ curriculum learning [1, 17] to achieve this purpose. OpenMatch (OM) [47] employs a Soft Open-set Consistency Regularization (SOCR) loss to hit the mark. Trash to Treasure (T2T) [25] employs a self-supervised learning strategy to achieve better performance. Class-aware Contrastive Semi-Supervised Learning (CCSSL) [58] utilizes both class-wise clustering and image-wise contrastive learning [32] to distinguish OOD data from ID data. Out-of-distributed Semantic Pruning (OSP) [54] proposes an aliasing OOD matching module and a soft orthogonality regularization to detect OOD data via using semantic information. However, these methods are not suitable to solve the intersectional class-mismatch problem.

2.3 Domain Adaptation

Domain adaptation is primarily concerned with leveraging the knowledge gained from the labeled source domain for application in the unlabeled target domain. This concept can be briefly categorized into four distinct scenarios. The first category is closed-set domain adaptation, which assumes that the classes of source data correspond to those of target data. Various existing methodologies [10, 39] focus on extracting class-discriminative and domain-invariant features from both source domain and target domain. The second category is partial domain adaptation, which works when the class set of the source data contains that of the target data. Certain strategies [4, 5] advocate the use of class-level weights applied to each source data point to achieve partial distribution match. The third category is open-set domain adaptation, in which the target domain contains additional classes to source domain. Notably, Zhuo *et al.* [55] put forth the idea of harnessing word vectors to identify these open domains. The fourth category is universal domain adaptation, which operates independently of any prerequisite knowledge of the class relationship between the source domain and target domain. An approach by You *et al.* [60] employs domain knowledge coupled with entropy values to identify the data belonging to the classes shared by both source domain and target domain.

3 PRELIMINARIES

Let $\mathcal{D} = \left\{\mathbf{x}_i \in \mathcal{X} \subset \mathbb{R}^d, i = 1, 2, \cdots, n, n = n_l + n_u\right\}$ denote the training set, where the first n_l images are labeled examples and the remaining n_u ones are unlabeled examples with typically $n_l \ll n_u$. We use $\mathcal{D}_l = \left\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_{n_l}, y_{n_l})\right\}$ to denote the labeled set with $y_i \in \mathcal{Y}_l = \{1, 2, \cdots, c\}$, where c is the number of known classes. Besides, we use $\mathcal{D}_u = \left\{\mathbf{x}_{n_l+1}, \mathbf{x}_{n_l+2}, \cdots, \mathbf{x}_{n_l+n_u}\right\}$ to denote the unlabeled set whose class space is denoted as $\mathcal{Y}_u$. In reality, the assumption that $\mathcal{Y}_l = \mathcal{Y}_u$ is difficult to realize. Under the intersectional class-mismatch setting, 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 $\mathcal{Y}_{shared}$, $\mathcal{Y}_{pri}$ and $\mathcal{Y}_{ood}$ constitute a shared set $\mathcal{D}_{shared}$, a private set $\mathcal{D}_{pri}$, and an OOD set $\mathcal{D}_{ood}$, respectively. Note that $\mathcal{Y}_{pri}$, $\mathcal{Y}_{ood}$, and $\mathcal{Y}_{shared}$ are unknown before training, so are $\mathcal{D}_{pri}$, $\mathcal{D}_{ood}$, and $\mathcal{D}_{shared}$. The class space of testing set is the same as $\mathcal{Y}_l$. According to [44], it is important to weaken the negative influence caused by OOD examples. To achieve this goal, existing traditional class-mismatched SSL methods [8, 24, 25, 47, 61] choose to weigh unlabeled data by weights $\mathbf{w}$ during training to optimize parameters θ_C, thus the objective function can be defined as follows:

$$\min_{\theta_C} \ \mathcal{L}_{ce}(\mathbf{x}^l; \theta_C) + \mathcal{L}_{ssl}(\mathbf{x}^u, \mathbf{w}; \theta_C), \tag{1}$$

where $\mathbf{x}^l$ and $\mathbf{x}^u$ denote examples from $\mathcal{D}_l$ and $\mathcal{D}_u$, respectively. Note that $\mathcal{L}_{ce}$ is the classical cross-entropy loss which compares the network prediction on every labeled example with its ground-truth label, while $\mathcal{L}_{ssl}$ means

Table 1. Variables and Definitions.

Variables	Definitions
$\mathcal{X}$	feature space of training set
$\mathcal{Y}$	class space of training set
$\mathcal{D}$	training set
$\mathbf{x}_i$	the ith example in $\mathcal{D}$
y_i	label of $\mathbf{x}_i$
w_i	weight of $\mathbf{x}_i$
s_i	transferability score of $\mathbf{x}_i$
p_i	prediction confidence value of $\mathbf{x}_i^u$ (only for unlabeled data)

the loss defined on unlabeled examples, such as entropy minimization, consistency regularization, etc. Last but not least, in $\mathbf{w}$, the i-th element w_i is used to weigh the i-th unlabeled example. However, the above-mentioned class-mismatched SSL methods do not consider the existence of $\mathcal{D}_{pri}$, and they often share the same network to conduct the ID data classification and OOD data detection while calculating $\mathbf{w}$. In complete, those methods are not suitable for the intersectional class-mismatched setting. In this study, we propose a new method to weigh unlabeled data reasonably, and we will confirm its effectiveness in Section 5. Table 1 lists the main variables we use throughout the paper.

4 THE PROPOSED DWAL METHOD

This section explains our proposed DWAL algorithm in a detailed way. We first introduce the motivation, and overview of DWAL. Then, we detail the three components in DWAL, namely enhanced adversarial domain adaptation, dissimilarity maximization, and weighted semi-supervised learning.

4.1 Motivation

The proposed DWAL aims to train an SSL classifier on the training set $\mathcal{D} = \mathcal{D}_l \cup \mathcal{D}_u$, which can precisely classify an unseen example $\mathbf{x}_i$ with unknown true label $y_i \in \mathcal{Y}_l$. As not all unlabeled examples are helpful to train an accurate classifier, it is important to weigh each unlabeled example reasonably during the classifier training. To achieve this goal, we need to deal with three problems. First, $\mathcal{D}_u$ may contain both useful and harmful examples for training an SSL classifier, so it is necessary to decrease the weights of those harmful OOD data while increasing weights of ID data. Second, since $\mathcal{Y}_{pri}$ and $\mathcal{Y}_u$ are different, it is also critical to detect $\mathcal{D}_{pri}$ from $\mathcal{D}_l$, and then maximize the disagreement between the feature distribution of private data and that of unlabeled data, in case unlabeled examples are incorrectly associated to $\mathcal{D}_{pri}$. Third, the detection part aims to seek invariant features of ID data for distinguishing OOD data from ID data, while the classification part focuses on finding variant features of ID data to discriminate themselves. In this case, the optimization of detection part is against the optimization of classification part, therefore we need to carry out the two parts separately to avoid a conflicting optimization.

4.2 Overview of DWAL

The overall framework of our DWAL approach is shown in Fig. 2, in which G, D, R, C and C' denote the feature extractor, discriminator, image restorer, classifier, and pre-trained classifier, respectively. Besides, θ_G, θ_D, θ_R and θ_C are the parameters of corresponding networks, respectively. Our DWAL includes three key components: enhanced adversarial domain adaptation, dissimilarity maximization, and weighted semi-supervised learning.

In enhanced adversarial domain adaptation, given $\mathcal{D}_l$ and $\mathcal{D}_u$, we use the feature extractor G to obtain feature representations $G(\mathbf{x})$ for $\mathbf{x} \in \mathcal{D}_l \cup \mathcal{D}_u$. Then, a discriminator D is imposed on $G(\mathbf{x})$ to get the outputs $D(G(\mathbf{x}))$ quantifying the similarity of $\mathbf{x}$ to the unlabeled set. According to the outputs of D, we define transferability scores $\mathbf{s}$ for training images, where the i-th element s_i denote the transferability score of $\mathbf{x}_i$. With the guidance of $\mathcal{L}_{adv}$,

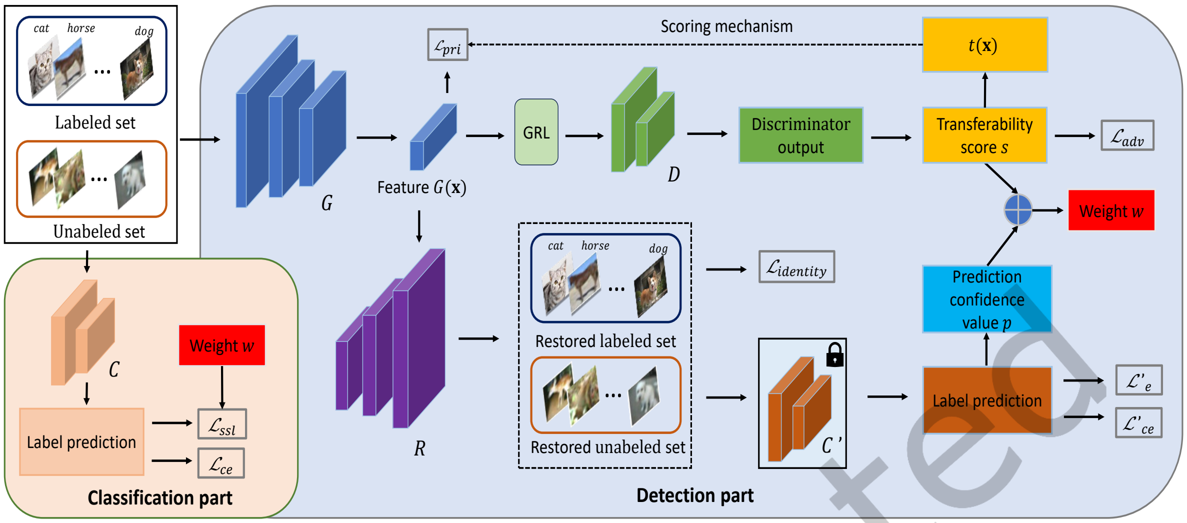

Fig. 2. This is the framework of our proposed DWAL, in which G, D, R, C and C' denote feature extractor, discriminator, image restorer, classifier and pre-trained classifier, respectively. A lock sign indicates "no optimization" in the framework. "GRL" indicates Gradient Reverse Layer [15] for adversarial domain adaptation. Through enhanced adversarial domain adaptation, we get weight for each example. According to the transferability scores $\mathbf{s}$ of examples in $\mathcal{D}_l$, dissimilarity maximization may help G maximize the disagreement between $\mathcal{D}_{pri}$ and $\mathcal{D}_u$, therefore improving the accuracy of weight of each example.

G is employed to confuse the discriminator D, while D should try to distinguish the presented data as labeled or unlabeled data. Through enhanced adversarial domain adaptation, the distribution gap between $\mathcal{Y}_l$ and $\mathcal{Y}_u$ can be reduced and the transferability score of each example can be harvested. At the same time, $G(\mathbf{x})$ will be forwarded into R to obtain the restored images $\mathbf{x}'$ where the size of each restored image $\mathbf{x}'_i$ is the same as the size of $\mathbf{x}_i$. We employ $\mathcal{L}_{identity}$ to encourage the restored image to be similar to its original version. We introduce a pre-trained classifier C' which is only trained on the labeled set without sharing parameters with classifier C, and C' outputs the label prediction of every restored image $\mathbf{x}'_i$. On one hand, $\mathcal{L}'_{ce}$ is cross-entropy minimization loss, which encourages R to restore features of labeled data to easy-to-classify images. On the other hand, $\mathcal{L}'_e$ is entropy maximization loss, which encourages R to restore features of unlabeled data to difficult-to-classify images. As a result, R can be trained to capture the features important for classifying data from $\mathcal{Y}_l$ and reduce the confidence of assigning OOD data to label of $\mathcal{Y}_l$ in the label prediction. The prediction confidence values $\mathbf{p}$ are calculated from the label predictions of images, which are restored from the features of unlabeled data. In $\mathbf{p}$, the i-th element p_i denote the prediction confidence value of $\mathbf{x}^u_i$.

In dissimilarity maximization, with the help of transferability scores of labeled examples, we can decide $\mathcal{D}_{pri}$, which are sent back to minimize $\mathcal{L}_{pri}$. As a result, G can be trained to maximize the disagreement between feature distribution of private data and that of unlabeled data, so that it can avoid the improper adaptation from unlabeled data to private data during adversarial domain adaptation, helping to obtain accurate transferability scores $\mathbf{s}$ and prediction confidence values $\mathbf{p}$.

In weighted semi-supervised learning, $\mathcal{D}_l$ and $\mathcal{D}_u$ will be fed to the classifier C to achieve their label predictions. The predictions of $\mathcal{D}_l$ will be used to compute $\mathcal{L}_{ce}$, and those of $\mathcal{D}_u$ will be used to compute $\mathcal{L}_{ssl}$ with their weights $\mathbf{w}$, made up of transferability scores and prediction confidence values.

It is worth noting that enhanced adversarial domain adaptation and dissimilarity maximization constitute detection part, and weighted semi-supervised learning is the classification part in Fig. 2. The weights utilized in the classification part are totally computed in detection part and they do not share any network, so that the conflicting optimization brought by the combination of detection part and classification part in other traditional class-mismatched SSL methods [8, 24, 47, 61] can be avoided, and the classification performance on the test set of the classifier can be improved. Therefore, the overall objective function of proposed DWAL can be formulated as:

$$\begin{aligned}\min_{\theta_G,\theta_R,\theta_C}\max_{\theta_D}\ & \mathcal{L}_{adv}(\mathbf{x};\theta_D,\theta_G)+\mathcal{L}_{pri}(\mathbf{x};\theta_G) \\ & +\mathcal{L}_{ce}(\mathbf{x}^l;\theta_C)+\mathcal{L}_{ssl}(\mathbf{x}^u,\mathbf{w};\theta_C) \\ & +\mathcal{L}_{identity}(\mathbf{x};\theta_R)+\mathcal{L}'_{ce}(\mathbf{x}^l;\theta_R)+\mathcal{L}'_{e}(\mathbf{x}^u;\theta_R).\end{aligned} \tag{2}$$

We have $\mathbf{x}^l$ mean labeled examples, and $\mathbf{x}^u$ mean unlabeled examples. We have $\mathbf{w}$ denote the weights of training examples. The weights of $\mathbf{x}^l$ are the transferability scores of $\mathbf{x}^l$. The weights of $\mathbf{x}^u$ consist of transferability scores of $\mathbf{x}^u$ and prediction confidence values of $\mathbf{x}^u$. We will detail each component below.

4.3 Enhanced Adversarial Domain Adaptation

To train an accurate SSL classifier under intersectional class-mismatch setting, it is critical to utilize useful ID data and weaken the influences of harmful OOD data. To achieve this goal, we propose an enhanced adversarial domain adaptation to assign weights $\mathbf{w}$ for examples from the perspectives of domain adaptation and a class-wise weighting mechanism, which can effectively detect private data in $\mathcal{D}_l$ and weaken the influences of OOD data in $\mathcal{D}_u$. The weights $\mathbf{w}$ should satisfy the following inequalities

$$\begin{aligned}\mathbb{E}_{\mathbf{x}^u\in\mathcal{D}_{shared}}\mathbf{w} > \mathbb{E}_{\mathbf{x}^u\in\mathcal{D}_{ood}}\mathbf{w}, \\ \mathbb{E}_{\mathbf{x}^l\in\mathcal{D}_{shared}}\mathbf{w} > \mathbb{E}_{\mathbf{x}^l\in\mathcal{D}_{pri}}\mathbf{w},\end{aligned} \tag{3}$$

where $\mathbf{x}^l$ denotes labeled examples and $\mathbf{x}^u$ denotes unlabeled examples. Specifically, first, $\mathcal{D}_u$ may contain both useful and harmful examples for training an SSL classifier, so it is critical to decrease the weights of those harmful OOD data, while increasing weights of ID data that are shared examples in $\mathcal{D}_u$. As a result, the expectation of weights $\mathbf{w}$ of ID data should be larger than that of OOD data, so we have $\mathbb{E}_{\mathbf{x}^u\in\mathcal{D}_{shared}}\mathbf{w} > \mathbb{E}_{\mathbf{x}^u\in\mathcal{D}_{ood}}\mathbf{w}$. Second, as the class space of private data is different from that of unlabeled data, it is important to detect private data for avoiding the improper adaptation from unlabeled data to private data. Therefore, we have $\mathbb{E}_{\mathbf{x}^l\in\mathcal{D}_{shared}}\mathbf{w} > \mathbb{E}_{\mathbf{x}^l\in\mathcal{D}_{pri}}\mathbf{w}$. Consequently, we can distinguish private data from labeled shared data based on their weights $\mathbf{w}$, and the labeled examples with small weights are considered as private data. As a result, the above inequities should hold in a large margin for better classification performance instructed by $\mathbf{w}$, and we achieve this through producing reliable transferability scores and prediction confidence values.

4.3.1 Transferability Score. We treat $\mathcal{D}_l$ and $\mathcal{D}_u$ as the source domain and target domain, respectively. To highlight shared data in both $\mathcal{D}_l$ and $\mathcal{D}_u$, we propose an adversarial domain adaptation strategy adopted from [19]. Here the discriminator D is trained to determine whether an example belongs to $\mathcal{D}_u$ or $\mathcal{D}_l$, and the feature extractor G is used to fool D. In adversarial domain adaptation, if an example's domain is hard to determine, the example can be considered transferable and it is likely to appear in both source domain and target domain. Similarly, if some examples belong to $\mathcal{D}_{shared}$, it is difficult for D to determine their domains, because both $\mathcal{D}_u$ and $\mathcal{D}_l$ contain examples belonging to $\mathcal{D}_{shared}$. As a result, the classes of transferable examples are likely to belong to $\mathcal{Y}_{shared}$. Consequently, we propose to use the transferability score s_i to weigh each example, and the process can be formulated as:

$$\begin{aligned}\min_{\theta_G}\max_{\theta_D}\ \mathcal{L}_{adv} = &-\frac{1}{n_l}\sum_{i=1}^{n_l} s_i \ln(D(G(\mathbf{x}_i))) \\ &-\frac{1}{n_u}\sum_{i=n_l+1}^{n_l+n_u} s_i \ln(1-D(G(\mathbf{x}_i))),\end{aligned} \tag{4}$$

where s_i is the transferability score of $\mathbf{x}_i$. The parameters θ_D are learned to distinguish the labeled data from the unlabeled data by maximizing a cross-entropy loss. The feature extractor's parameters θ_G are learned to deceive D by minimizing a cross-entropy loss. In this case, the output $D(G(\mathbf{x}_i))$ is in the range of $(0, 1)$. If an example $\mathbf{x}_i$ belongs to $\mathcal{D}_{shared}$, $D(G(\mathbf{x}_i))$ should be close to 0.5, because it is difficult for D to distinguish its domain. Consequently, the transferability score s_i of an example $\mathbf{x}_i$ is computed by

$$s_i = 1 - 2 \cdot |D(G(\mathbf{x}_i)) - 0.5| \,. \tag{5}$$

To conclude, for $\mathbf{x} \in \mathcal{D}_l$, $\mathbb{E}_{\mathbf{x}\in\mathcal{D}_{shared}}\mathbf{s} > \mathbb{E}_{\mathbf{x}\in\mathcal{D}_{pri}}\mathbf{s}$; for $\mathbf{x} \in \mathcal{D}_u$, $\mathbb{E}_{\mathbf{x}\in\mathcal{D}_{shared}}\mathbf{s} > \mathbb{E}_{\mathbf{x}\in\mathcal{D}_{ood}}\mathbf{s}$. It is worth noting that we can obtain the transferability scores of all examples. Hence, our transferability score can help highlight $\mathcal{D}_{shared}$ in both $\mathcal{D}_l$ and $\mathcal{D}_u$.

4.3.2 Prediction Confidence Value. To further enhance the detection performance, we calculate a prediction confidence value for each unlabeled example via a class-wise weighting mechanism. Typically, the prediction confidence value is generated by the SSL classifier, which predicts the classification probability of each example. However, when dealing with a class mismatch scenario, the outcome of an OOD example using the aforementioned process tends to be similar to that of a shared example, incurring the "overconfidence" issue. This issue is inconsistent with the expectations of DWAL regarding weights of unlabeled data (*i.e.*, Eq. (3)). The issue arises due to the distributional difference between labeled data and OOD data [20, 21, 43, 63]. To address the challenge, we first construct an image restorer R to restore the generated feature $G(\mathbf{x}_i)$ to $R(G(\mathbf{x}_i))$, which is denoted as $\mathbf{x}'_i$. Thanks to the adversarial domain adaptation between G and D, the feature $G(\mathbf{x}_i)$ can reliably describe the information of $\mathbf{x}_i$, and mitigate the distribution difference between labeled data and OOD data. $\mathcal{L}_{identity}$ calculates the ℓ_2 distance between $\mathbf{x}_i$ and $\mathbf{x}'_i$ as an instruction of the generation of the restored image, so that the incorrect restoration can be avoided. The objective function $\mathcal{L}_{identity}$ can be formulated as

$$\min_{\theta_R} \mathcal{L}_{identity} = \frac{1}{n_l+n_u}\sum\nolimits_{i=1}^{n_l+n_u} \|\mathbf{x}_i - \mathbf{x}'_i\|_2, \tag{6}$$

As a result, the restored image $\mathbf{x}'_i$ can get useful information from $\mathbf{x}_i$, and mitigate the distribution difference between labeled data and OOD data. Therefore, we can replace $\mathbf{x}_i$ with $\mathbf{x}'_i$ to calculate reliable prediction confidence value.

Then, a pre-trained classifier C' is imposed on the restored images, where the pre-trained classifier has been trained on the labeled data (*i.e.*, $\mathcal{D}_l$) and keeps fixed in this stage. Note that, the classifier C' is only trained on $\mathcal{D}_l$, so that the label prediction of C' could avoid negative influences caused by incorrect information, which inevitably appears in SSL classifier [8, 59]. The loss $\mathcal{L}'_{ce}$ is the classical cross-entropy loss for entropy minimization on $\mathbf{x}'_i$ when $\mathbf{x}_i$ is labeled data, so that we can encourage R to emphasize the features that are easy-to-classify in labeled data. By denoting $H(\cdot)$ as the cross-entropy loss, the objective function $\mathcal{L}'_{ce}$ is defined by

$$\min_{\theta_R} \mathcal{L}'_{ce} = \frac{1}{n_l}\sum\nolimits_{i=1}^{n_l} H(y_i, C'(\mathbf{x}'_i)). \tag{7}$$

We employ $\mathcal{L}_e'$ to enhance the distinction ability between the prediction confidence values of shared data and those of OOD data. Specifically, the minimization of $\mathcal{L}_e'$ tries to prevent restored OOD data from being mistakenly assigned to known classes with a high probability, therefore reducing the prediction confidence values of OOD data. As such, we can further reduce the negative influence caused by overconfidence issue. The induced objective is

$$\min_{\theta_R} \mathcal{L}_e' = \frac{1}{n_u} \sum\nolimits_{i=n_l+1}^{n_l+n_u} \vartheta(C'(\mathbf{x}_i')) \cdot \ln(\vartheta(C'(\mathbf{x}_i'))), \tag{8}$$

where $\vartheta(\cdot)$ means the softmax function.

At this point, we introduce a class-wise weighting mechanism guaranteed by margin theory to calculate a prediction confidence value. The margin between features and the classification surface can reflect the confidence of label prediction, which contributes to a generalizable SSL classifier [33, 59, 64]. According to [33, 59, 64], given an unlabeled example $\mathbf{x}_i^u$, its prediction confidence value p_i can be computed by

$$p_i = C'(R(G(\mathbf{x}_i^u)), \widehat{y}) - \max_{y \neq \widehat{y}} C'(R(G(\mathbf{x}_i^u)), y), \tag{9}$$

where

$$\widehat{y} = \operatorname*{argmax}_{y \in \mathcal{Y}_l} C'(R(G(\mathbf{x}_i^u)), y). \tag{10}$$

Here $C'(R(G(\mathbf{x}_i^u)), y)$ means the classification probability of the restored image $R(G(\mathbf{x}_i^u))$ belonging to class y in $\mathcal{Y}_l$.

Obviously, the minimization of $\mathcal{L}_{ce}'$ will enlarge the prediction confidence values for labeled data. As shared data may appear in both $\mathcal{D}_l$ and $\mathcal{D}_u$, the minimization of $\mathcal{L}_{ce}'$ will enlarge the prediction confidence values for shared data in the unlabeled set. However, OOD data only exist in $\mathcal{D}_u$, so the prediction confidence values for OOD data cannot be enlarged through the minimization of $\mathcal{L}_{ce}'$, and the prediction confidence value of OOD data will only be reduced by the minimization of $\mathcal{L}_e'$. As a result, we have $\mathbb{E}_{\mathbf{x}_i^u \in \mathcal{D}_{shared}} \mathbf{p} > \mathbb{E}_{\mathbf{x}_i^u \in \mathcal{D}_{ood}} \mathbf{p}$.

Finally, with the help of transferability score s_i and prediction confidence value p_i, we can obtain the weight w_i of each example $\mathbf{x}_i$ by

$$w_i = \begin{cases} s_i, \ \mathbf{x}_i \in \mathcal{D}_l, \\ s_i + p_i, \ \mathbf{x}_i \in \mathcal{D}_u. \end{cases} \tag{11}$$

Through Eq. (11), Eq. (3) can be perfectly satisfied. For labeled examples, their weights can help detect private data, which will be detailed in Section 4.4. For unlabeled data, we can use their weights to weigh the importance of all unlabeled examples during the classifier training, which will be detailed in Section 4.5. To conclude, $\mathbf{w}$ can weigh the labeled data and unlabeled data accurately.

4.4 Dissimilarity Maximization

The difference between intersectional and traditional class-mismatch setting lies in the existence of $\mathcal{Y}_{pri}$. As examples of $\mathcal{Y}_{pri}$ will appear in testing set, private examples can provide important supervision information during training. However, they will affect the adaptation of unlabeled examples. This is because that $\mathcal{Y}_{pri}$ is totally different from $\mathcal{Y}_u$, so if an unlabeled example is adapted to $\mathcal{D}_{pri}$, it will cause confusion and lead to imprecise transferability score. Consequently, it is important to find out these examples of $\mathcal{D}_{pri}$ to prevent adaptation to them. As such private data have the labels of $\mathcal{Y}_{pri}$ which only exist in $\mathcal{D}_l$, their transferability scores could be lower than those of other labeled data. Therefore, we propose to detect the examples of $\mathcal{D}_{pri}$ from $\mathcal{D}_l$ by computing

$$t(\mathbf{x}_i) = \begin{cases} 0, s_i \geq z, \\ 1, s_i < z, \end{cases} \quad z = \frac{1}{n_l} \cdot \sum\nolimits_{i=1}^{n_l} s_i. \tag{12}$$

Since the averaged transferability score of all labeled examples keeps updating, z is updated in each training iteration as well. By using Eq. (12), we can update $\mathcal{D}_{pri}$ in each iteration via the following rules: if $t(\mathbf{x}_i) = 1$, $\mathbf{x}_i$ should be put into $\mathcal{D}_{pri}$, otherwise $\mathbf{x}_i$ should not be put into $\mathcal{D}_{pri}$. After obtaining $\mathcal{D}_{pri}$, we prevent it from confusing the normal adaptation of unlabeled data by enforcing the examples in $\mathcal{D}_{pri}$ to be far from the examples in $\mathcal{D}_u$. Specifically, we calculate the dissimilarity between the examples in $\mathcal{D}_{pri}$ and $\mathcal{D}_u$. Let $\mathbf{x}_j$ and $\mathbf{x}_k$ denote the j-th example in $\mathcal{D}_{pri}$ and the k-th example in $\mathcal{D}_u$, respectively, then the dissimilarity between $\mathbf{x}_j$ and $\mathbf{x}_k$ is computed by

$$m_{j,k} = \frac{1}{1 + e^{-q_{j,k}}}, \tag{13}$$

where $q_{j,k}$ is the ℓ_2 distance between $G(\mathbf{x}_j)$ and $G(\mathbf{x}_k)$, and $m_{j,k}$ is used to map $q_{j,k}$ from $(0, +\infty)$ to $(0.5, 1)$. To maximize the disagreement between $\mathbf{x}_j$ and $\mathbf{x}_k$, the optimization problem with objective function $\mathcal{L}_{pri}(\cdot)$ can be formulated as:

$$\min_{\theta_G} \mathcal{L}_{pri} = -\frac{1}{n_u} \frac{1}{n_{pri}} \sum_{j=1}^{n_{pri}} \sum_{k=n_l+1}^{n_l+n_u} \ln(m_{j,k}), \tag{14}$$

where n_{pri} is the number of labeled examples which are attributed to $\mathcal{D}_{pri}$.

To conclude, we can make the features of examples in $\mathcal{D}_{pri}$ be far from the features of examples in $\mathcal{D}_u$ by conducting dissimilarity maximization, which avoids the improper adaptation from unlabeled data to private data in domain adaptation.

4.5 Weighted Semi-Supervised Learning

Through the enhanced adversarial domain adaptation and dissimilarity maximization, we can adjust the effect of each unlabeled example according to its weight. Then, we can implement semi-supervised learning under the setting of intersectional class mismatch. Specifically, for $\mathcal{D}_l$, we use the cross-entropy loss $\mathcal{L}_{ce}$ on the labeled example. For $\mathcal{D}_u$, we use a consistency regularization with the sum of transferability score and prediction confidence value controlling the weight of each unlabeled example for training the classifier C. Therefore, this process can be formulated as:

$$\min_{\theta_C} \mathcal{L}_{ssl} = -\frac{1}{n_u} \sum_{i=n_l+1}^{n_l+n_u} w_i \| C(perturb(\mathbf{x}_i)) - C(\mathbf{x}_i) \|_2^2, \tag{15}$$

where $perturb(\cdot)$ means a perturbation on $\mathbf{x}_i$, which is widely used in consistency regularization of SSL works such as [24, 35, 48]. It is worth noting that w_i is calculated in the detection part where the classifier C is not involved. As a result, the detection part and the classification part do not share any network, so that we can obtain accurate weights to boost the classification performance.

Overall, our DWAL approach can effectively utilize unlabeled examples to aid SSL training, and weaken the negative impacts caused by private data and OOD data. The detailed process of our DWAL approach is summarized in **Algorithm** 1.

Algorithm 1 Training process for our DWAL method.

Input: Labeled set $\mathcal{D}_l = \{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_{n_l}, y_{n_l})\}$, unlabeled set $\mathcal{D}_u = \{\mathbf{x}_{n_l+1}, \cdots, \mathbf{x}_{n_l+n_u}\}$.

for $i = 1$ to MaxIter **do**
  Optimize θ_G and θ_D according to Eq. (4);
  Compute s_i for each example $\mathbf{x}_i$ according to Eq. (5)
  Optimize θ_R according to Eq. (6), Eq. (7), and Eq. (8);
  Compute p_i for each unlabeled example $\mathbf{x}_i^u$ according to Eq. (9)
  Compute w_i for each example $\mathbf{x}_i$ according to Eq. (11)
  Find out the data in $\mathcal{D}_{pri}$ according to Eq. (12);
  Optimize θ_G according to Eq. (14);
  Optimize θ_C by minimizing $\mathcal{L}_{ce}$;
  Optimize θ_C according to Eq. (15);
end for

Output: θ_C.

5 EXPERIMENTS

In this section, we carry out experiments to show the effectiveness of DWAL. We first introduce the experimental settings and implementation details. Then, we provide the performance comparison, ablation study, and performance verification.

5.1 Experimental Settings

We will introduce the experimental datasets and the formulation of intersectional class-mismatch setting.

5.1.1 Datasets. We adopt five image classification datasets in the comparison experiments, and the details are provided below. (1) **MNIST** includes 60,000 training images and 10,000 test images with the size of 28×28, belonging to 10 classes: "0"∼"9". (2) **SVHN** is collected from house numbers with 73,257 training images and 26,032 test images with the size of 32×32, belonging to 10 classes: "0"∼"9". (3) **CIFAR-10** includes 60,000 training images and 10,000 test images with the size of 32×32. This dataset contains 10 classes, which consist of six animal classes (*i.e.*, "bird", "cat", "deer", "dog", "frog", and "horse", of which the class IDs are denoted as "0"∼"5", respectively) and four transportation tool classes (*i.e.*, "airplane", "automobile", "ship", and "truck", of which the class IDs are "6"∼"9", respectively). (4) **ImageNet-100** is a subset of ImageNet [9] and has 100 classes, which contains 133,116 images with a size of 32×32 from ImageNet. In this dataset, the class IDs are denoted as "0"∼"99", respectively. (5) **Fundus** is a real-world dataset. Its labeled set and unlabeled set are from two public datasets, respectively, *i.e.*, TAOP[1] and ODIR[2]. Note that the fundus disease images of TAOP and ODIR are collected from patients of different hospitals, respectively, so the types of eye diseases are different between labeled set and unlabeled set. This is consistent with the intersectional class-mismatch setting. In this dataset, the labeled set has five classes: "0"∼"4", and the unlabeled set has eight classes: "1"∼"8".

5.1.2 Intersectional Class-Mismatch Setting. To evaluate the capability of our DWAL method in handling SSL tasks with intersectional class mismatch, 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 DWAL 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.

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

For MNIST, SVHN, and CIFAR-10, they share the same setting because they all contain 10 classes. By following [24], 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: $\mathcal{Y}_{shared} = \{5\}$. Thereby, the number of classes contained in $\mathcal{Y}_{shared}$ will decrease from *Case* 1 to *Case* 4. By following [24], the detailed settings of labeled and unlabeled data in MNIST, SVHN and CIFAR-10 are provided below: (1) For MNIST, we select 10 images from each class in $\mathcal{Y}_l$ to construct $\mathcal{D}_l$, and choose $3,332$ images per class from $\mathcal{Y}_u$ to form $\mathcal{D}_u$. (2) For SVHN, we select 100 images from each class in $\mathcal{Y}_l$ to construct $\mathcal{D}_l$, and choose $3,332$ images per class from $\mathcal{Y}_u$ to form $\mathcal{D}_u$. (3) For CIFAR-10, we select 400 images from each class in $\mathcal{Y}_l$ to construct $\mathcal{D}_l$, and choose $3,332$ images per class from $\mathcal{Y}_u$ to form $\mathcal{D}_u$. For ImageNet-100, as it includes 100 classes, we set its $\mathcal{Y}_l$ and $\mathcal{Y}_{ood}$ as $\{0, 1, ..., 59\}$ and $\{60, 61, ..., 99\}$, respectively, by following [28]. Then we define two ways of $\mathcal{Y}_{shared}$ for ImageNet-100, namely 1) *Way* 1: $\mathcal{Y}_{shared} = \{15, 16, ..., 99\}$; 2) *Way* 2: $\mathcal{Y}_{shared} = \{30, 31, ..., 99\}$. For ImageNet-100, we choose 100 images from each class in $\mathcal{Y}_l$ to construct the $\mathcal{D}_l$, and then we select $1,208$ images from each class in $\mathcal{Y}_u$ to form $\mathcal{D}_u$. For Fundus, we have made the setting consistent with the assumption of our intersectional class-mismatch setting. Its labeled set containing $2,472$ examples of five classes is from TAOP, and its unlabeled set containing $5,814$ examples of eight classes is from ODIR. For each dataset, under different settings, we investigate the accuracy of each method on test sets, of which the class space is the same as $\mathcal{Y}_l$, and then report the mean classification accuracy over five trials.

5.2 Implementation Details

We will introduce the compared methods and the algorithmic settings.

5.2.1 Compared Methods. To validate the effectiveness of DWAL, we compare it with the following deep SSL methods: Pseudo Labeling (PL) [36], Π-model (PI) [35], VAT [40], Mean Teacher (MT) [51], MTCF [61], UASD [8], DS^{3}L [24], OM [47], T2T [25], CCSSL [58], FixMatch [49], FlexMatch [62], SemCo [42], and OSP [54]. Note that MTCF, UASD, DS^{3}L, OM, T2T, CCSSL, and OSP are designed for SSL under the traditional class-mismatch setting. In addition, we compare DWAL with a supervised baseline method, which simply trains a deep neural network on $\mathcal{D}_l$ based on the cross-entropy loss. Note that FlexMatch, SemCo, OM, T2T, CCSSL, and OSP largely depend on the framework of FixMatch, which uses weak and strong data augmentations to process training data and achieve consistency regularization. Different from the above methods, the DWAL and the rest of compared SSL methods are based on the framework of Pseudo-Labeling (PL), which only processes training data through Global Contrast Normalization (GCN) and Zero-phase Component Analysis Whitening (ZCA) strategy. Owing to the fact that the aforementioned SSL methods are implemented across diverse frameworks (*i.e.*, PL and FixMatch), we also implement our DWAL under the framework of FixMatch which is based on Data Augmentation (DA), namely "DWAL+DA". Specifically, in the classification part of DWAL+DA, the labeled data are weakly augmented in $\mathcal{L}_{ce}$, and the predictions of weakly augmented unlabeled data are compared with those of their strongly augmented versions in $\mathcal{L}_{ssl}$. The weak data augmentation and strong data augmentation employed in DWAL+DA are the same as those in FixMatch. As a result, for fairness of comparison, DWAL should be compared with the SSL methods following PL framework, including PL, PI, VAT, MT, MTCF, UASD, and DS^{3}L. DWAL+DA should be compared with the SSL methods based on FixMatch framework, including FixMatch, FlexMatch, SemCo, OM, T2T, CCSSL, and OSP. For all compared methods, their classifier architectures are the same with DWAL, and use well-tuned hyperparameters for each dataset. The data preprocessing method of each compared method is consistent with its original work. DWAL and DWAL+DA share the same network architectures, but their optimizers are not exactly identical, which will be shown in **Appendix**. The detailed $\mathcal{L}_{ssl}$ of DWAL+DA can be found in **Appendix** as well.

Table 2. Classification accuracies on MNIST.

Settings	**Methods**	*Case* 1	*Case* 2	*Case* 3	*Case* 4
Class-matched Methods	Supervised	87.92 ± 0.11 * *	87.92 ± 0.11 * *	87.92 ± 0.11 * *	87.92 ± 0.11 * *
	VAT [40]	94.02 ± 0.20 * *	88.75 ± 0.29 * *	82.17 ± 0.24 * *	78.98 ± 0.34 * *
	MT [51]	94.37 ± 0.16 * *	87.09 ± 0.28 * *	81.00 ± 0.27 * *	78.95 ± 0.32 * *
	PI [35]	94.09 ± 0.19 * *	86.66 ± 0.27 * *	78.12 ± 0.26 * *	76.62 ± 0.29 * *
	PL [36]	92.76 ± 0.23 * *	86.81 ± 0.31 * *	80.54 ± 0.33 * *	77.77 ± 0.36 * *
	FixMatch [49]	97.02 ± 0.16 *	96.15 ± 0.19 *	95.25 ± 0.21 *	92.34 ± 0.22 *
	FlexMatch [62]	96.30 ± 0.29 *	95.04 ± 0.35 *	93.57 ± 0.36 *	91.73 ± 0.40 *
	SemCo [42]	97.75 ± 0.15 *	96.67 ± 0.20 *	95.10 ± 0.24 *	93.28 ± 0.26 *
Class-mismatched Methods	MTCF [61]	94.48 ± 0.17 * *	89.12 ± 0.21 * *	82.53 ± 0.27 * *	80.83 ± 0.26 * *
	UASD [8]	95.18 ± 0.21 * *	90.73 ± 0.23 * *	85.71 ± 0.29 * *	83.48 ± 0.27 * *
	DS^{3}L [24]	95.22 ± 0.19 * *	89.37 ± 0.21 * *	82.62 ± 0.22 * *	81.23 ± 0.26 * *
	OM [47]	97.52 ± 0.21 *	96.22 ± 0.24 *	95.29 ± 0.30 *	93.37 ± 0.32 *
	T2T [25]	97.94 ± 0.19 *	96.87 ± 0.22 *	95.63 ± 0.27 *	93.60 ± 0.33 *
	CCSSSL [58]	97.16 ± 0.27 *	96.10 ± 0.27 *	94.68 ± 0.29 *	92.95 ± 0.34 *
	OSP [54]	97.98 ± 0.17 *	96.91 ± 0.22 *	95.66 ± 0.24 *	93.74 ± 0.35 *
Our Methods	DWAL	96.29 ± 0.16	93.54 ± 0.15	91.69 ± 0.14	91.31 ± 0.20
	DWAL+DA	**98.36 ± 0.15**	**97.27 ± 0.17**	**95.91 ± 0.17**	**94.02 ± 0.18**

5.2.2 Algorithmic Settings. The batch size for both labeled and unlabeled sets is set to 100. We report the mean classification accuracy of all test examples with the labeled classes over five trials. The proposed DWAL method is implemented with PyTorch, which is trained using two NVIDIA TITAN GPUs.

5.3 Performance Comparison

This part we will show the classification accuracies with averaged results of five runs for last several iterations on each dataset. To evaluate statistical significance, we perform the paired t-test [41] with 95% confidence level on the classification performances of our method and other compared methods. In Tables 2-6, the black "*" denotes that DWAL is significantly better than the compared method. Similarly, the blue "*" denotes that DWAL+DA is significantly better than the compared method. Experimental results confirm the advantage of our methods over other competitors.

5.3.1 Results on MNIST. Table 2 shows classification accuracies of different methods on MNIST. It can be seen that all methods perform better than the supervised baseline in *Case* 1. However, from *Case* 1 to *Case* 4, the performance of existing SSL methods degrades. Specifically, as for *Case* 3, it means that half of the classes of $\mathcal{Y}_l$ is unseen in $\mathcal{Y}_u$ and two-thirds of the classes of $\mathcal{Y}_u$ is unseen in $\mathcal{Y}_l$. In this case, it can be observed that many existing deep SSL methods perform worse than the supervised baseline method, while our DWAL method achieves good performance and shows advantage over PL-based methods (*i.e.*, VAT, MT, PL, PI, MTCF, UASD, and DS^{3}L). More importantly, DWAL+DA outperforms all the compared methods as well. Therefore, the advantage of our methods over other methods in tackling the intersectional class-mismatch problem is verified.

5.3.2 Results on SVHN. The results on SVHN are shown in Table 3. It can be observed that DWAL performs better than other compared methods without data augmentation in different cases. In these cases, most of the existing SSL methods perform worse than the supervised learning competitor, while DWAL achieves very encouraging results and outperforms PL-based methods. Moreover, DWAL+DA performs better than the FixMatch-based methods (*i.e.*, FixMatch, FlexMatch, SemCo, OM, T2T, CCSSL, and OSP).

5.3.3 Results on CIFAR-10. The comparison results on CIFAR-10 are shown in Table 4. We can see that DWAL performs better than most other compared methods in all cases. From *Case* 1 to *Case* 4, the number of classes contained in $\mathcal{Y}_{shared}$ decreases, which leads to the significant performance degradation of most SSL methods.

Table 3. Classification accuracies on SVHN.

Settings	**Methods**	*Case* 1	*Case* 2	*Case* 3	*Case* 4
Class-matched Methods	Supervised	82.93 ± 0.20 * *	82.93 ± 0.20 * *	82.93 ± 0.20 * *	82.93 ± 0.20 * *
	VAT [40]	83.49 ± 0.37 * *	82.38 ± 0.42 * *	80.28 ± 0.46 * *	79.56 ± 0.52 * *
	MT [51]	83.12 ± 0.32 * *	81.95 ± 0.36 * *	81.69 ± 0.47 * *	79.41 ± 0.49 * *
	PI [35]	81.84 ± 0.41 * *	79.95 ± 0.50 * *	79.69 ± 0.49 * *	78.95 ± 0.53 * *
	PL [36]	82.94 ± 0.36 * *	82.27 ± 0.43 * *	81.18 ± 0.48 * *	79.15 ± 0.52 * *
	FixMatch [49]	94.98 ± 0.19 *	93.02 ± 0.20 *	91.89 ± 0.20 *	90.72 ± 0.22 *
	FlexMatch [62]	94.53 ± 0.36 *	92.75 ± 0.40 *	90.81 ± 0.41 *	89.05 ± 0.44 *
	SemCo [42]	95.44 ± 0.17 *	94.27 ± 0.21 *	92.11 ± 0.21 *	90.98 ± 0.25 *
Class-mismatched Methods	MTCF [61]	85.89 ± 0.32 * *	85.35 ± 0.42 * *	84.73 ± 0.43 * *	83.36 ± 0.46 * *
	UASD [8]	85.78 ± 0.35 * *	85.07 ± 0.36 * *	84.42 ± 0.43 * *	83.74 ± 0.45 * *
	DS^{3}L [24]	85.79 ± 0.29 * *	83.82 ± 0.38 * *	82.89 ± 0.45 * *	81.95 ± 0.44 * *
	OM [47]	94.44 ± 0.27 *	92.88 ± 0.30 *	90.81 ± 0.33 *	89.70 ± 0.36 *
	T2T [25]	94.36 ± 0.30 *	93.13 ± 0.33 *	91.01 ± 0.35 *	89.89 ± 0.38 *
	CCSSL [58]	93.72 ± 0.32 *	92.90 ± 0.34 *	90.47 ± 0.37 *	89.58 ± 0.41 *
	OSP [54]	94.54 ± 0.33 *	93.32 ± 0.37 *	91.19 ± 0.40 *	90.05 ± 0.46 *
Our Methods	DWAL	86.12 ± 0.34	85.76 ± 0.37	85.21 ± 0.26	84.98 ± 0.33
	DWAL+DA	**96.58 ± 0.26**	**95.31 ± 0.27**	**93.09 ± 0.26**	**92.40 ± 0.31**

Table 4. Classification accuracies on CIFAR-10.

Settings	**Methods**	*Case* 1	*Case* 2	*Case* 3	*Case* 4
Class-matched Methods	Supervised	69.83 ± 0.35 * *	69.83 ± 0.35 * *	69.83 ± 0.35 * *	69.83 ± 0.35 * *
	VAT [40]	71.46 ± 0.61 * *	70.10 ± 0.67 * *	68.06 ± 0.58 * *	66.10 ± 0.62 * *
	MT [51]	70.46 ± 0.56 * *	69.86 ± 0.69 * *	68.42 ± 0.67 * *	66.32 ± 0.71 * *
	PI [35]	70.83 ± 0.45 * *	69.42 ± 0.64 * *	68.75 ± 0.59 * *	66.21 ± 0.58 * *
	PL [36]	70.66 ± 0.49 * *	69.03 ± 0.55 * *	68.33 ± 0.63 * *	66.65 ± 0.66 * *
	FixMatch [49]	86.31 ± 0.25 *	84.17 ± 0.25 *	81.91 ± 0.27 *	80.77 ± 0.30 *
	FlexMatch [62]	87.47 ± 0.20 *	85.37 ± 0.21 *	83.09 ± 0.24 *	81.95 ± 0.24 *
	SemCo [42]	87.84 ± 0.27 *	85.81 ± 0.29 *	83.58 ± 0.31 *	82.52 ± 0.34 *
Class-mismatched Methods	MTCF [61]	75.16 ± 0.54 * *	74.27 ± 0.51 * *	73.21 ± 0.47 * *	71.06 ± 0.48 * *
	UASD [8]	75.01 ± 0.51 * *	74.60 ± 0.53 * *	74.01 ± 0.49 * *	71.82 ± 0.57 * *
	DS^{3}L [24]	74.76 ± 0.47 * *	72.67 ± 0.48 * *	71.52 ± 0.55 * *	70.18 ± 0.56 * *
	OM [47]	88.37 ± 0.35 *	86.06 ± 0.40 *	83.92 ± 0.40 *	82.41 ± 0.42 *
	T2T [25]	87.84 ± 0.33 *	85.75 ± 0.36 *	83.49 ± 0.37 *	82.28 ± 0.43 *
	CCSSL [58]	87.42 ± 0.35 *	85.58 ± 0.38 *	83.46 ± 0.41 *	82.44 ± 0.46 *
	OSP [54]	86.92 ± 0.34 *	84.98 ± 0.39 *	82.72 ± 0.45 *	81.07 ± 0.49 *
Our Methods	DWAL	76.18 ± 0.57	75.57 ± 0.53	74.53 ± 0.58	72.35 ± 0.66
	DWAL+DA	**90.58 ± 0.29**	**88.51 ± 0.31**	**86.48 ± 0.34**	**85.62 ± 0.38**

Our DWAL still performs better than those PL-based methods, and DWAL+DA shows its advantage over other FixMatch-based methods. This indicates that our methods can effectively cope with the intersectional class-mismatch problem.

5.3.4 Results on ImageNet-100. The results are shown in Table 5. ImageNet-100 has more classes than MNIST, SVHN and CIFAR-10 which means that its complexity is higher than other datasets. Under the complex situation, it can be observed that our methods perform better than SSL methods within the same framework.

5.3.5 Results on Fundus. Fundus is a real-world clinic dataset, and the comparison results on Fundus are shown in Table 6. From the results, it can be observed that DWAL performs better than PL-based methods. Moreover, by

Table 5. Classification accuracies on ImageNet-100.

Settings	Methods	*Way* 1	*Way* 2
Class-matched Methods	Supervised	40.03 ± 0.21 * *	40.03 ± 0.21 * *
	VAT [40]	39.42 ± 0.21 * *	37.39 ± 0.25 * *
	MT [51]	38.66 ± 0.14 * *	37.30 ± 0.21 * *
	PI [35]	39.57 ± 0.19 * *	38.33 ± 0.27 * *
	PL [36]	39.17 ± 0.15 * *	37.83 ± 0.19 * *
	FixMatch [49]	52.55 ± 0.17 *	50.15 ± 0.23 *
	FlexMatch [62]	54.23 ± 0.18 *	53.26 ± 0.29 *
	SemCo [42]	54.94 ± 0.33 *	53.88 ± 0.42 *
Class-mismatched Methods	MTCF [61]	40.86 ± 0.17 * *	38.57 ± 0.26 * *
	UASD [8]	40.20 ± 0.19 * *	37.97 ± 0.28 * *
	DS^{3}L [24]	39.91 ± 0.16 * *	37.52 ± 0.24 * *
	OM [47]	55.52 ± 0.34 *	54.01 ± 0.39 *
	T2T [25]	54.75 ± 0.29 *	53.45 ± 0.33 *
	CCSSL [58]	55.16 ± 0.27 *	53.73 ± 0.40 *
	OSP [54]	55.82 ± 0.34 *	54.61 ± 0.37 *
Our Methods	DWAL	42.12 ± 0.16	41.23 ± 0.22
	DWAL+DA	**58.42 ± 0.24**	**56.17 ± 0.32**

Table 6. Classification accuracies on Fundus.

Settings	Methods	Accuracy
Class-matched Methods	Supervised	38.07 ± 0.26 * *
	VAT [40]	40.03 ± 0.61 * *
	MT [51]	39.25 ± 0.64 * *
	PI [35]	39.95 ± 0.52 * *
	PL [36]	40.51 ± 0.59 * *
	FixMatch [49]	78.39 ± 0.41 *
	FlexMatch [62]	80.27 ± 0.37 *
	SemCo [42]	80.87 ± 0.42 *
Class-mismatched Methods	MTCF [61]	42.76 ± 0.51 * *
	UASD [8]	42.48 ± 0.48 * *
	DS^{3}L [24]	41.74 ± 0.57 * *
	OM [47]	80.02 ± 0.55 *
	T2T [25]	78.63 ± 0.48 *
	CCSSL [58]	80.75 ± 0.56 *
	OSP [54]	79.89 ± 0.58 *
Our Methods	DWAL	46.57 ± 0.79
	DWAL+DA	**83.42 ± 0.56**

employing the same data augmentation strategy, our DWAL+DA surpasses all other FixMatch-based methods, indicating that our method can handle the intersectional class-mismatched problem.

5.4 Ablation Study

To investigate the effectiveness of different key components in our DWAL, we conduct the following ablative experiments, including: 1) we remove dissimilarity maximization term while keeping others fixed, denoted as "w/o dissimilarity maximization"; 2) we drop transferability scores **s** and only use prediction confidence values **p** to weigh unlabeled data, denoted as "w/o transferability score"; 3) we drop prediction confidence values **p** and only use transferability scores **s** to weigh unlabeled data, denoted as "w/o prediction confidence"; 4) we drop weights of unlabeled examples, *i.e.*, both prediction confidence values **p** and transferability scores **s**, in

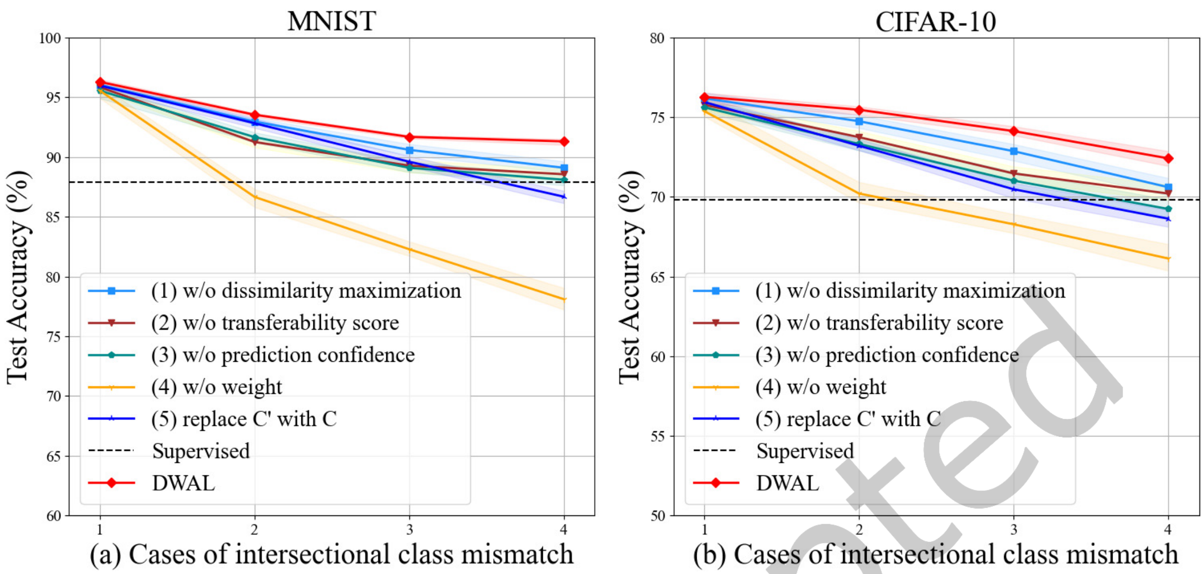

Fig. 3. Ablation study on MNIST and CIFAR-10. Shaded regions indicate a standard deviation over five trails.

$\mathcal{L}_{ssl}$, denoted as "w/o weight"; and 5) we replace pre-trained classifier C' with classifier C to calculate prediction confidence value, denoted as "replace C' with C".

Fig. 3 shows the ablative results on MNIST and CIFAR-10. It can be clearly seen how DWAL improves classification performance. Without transferability score or prediction confidence value, (see line (2) and (3)), the performance will decrease. When transferability score and prediction confidence value are both dropped, the whole DWAL method is equal to the traditional SSL method and we observe that the line (4) suffers the largest performance drop when compared with other lines, indicating that the transferability score and prediction confidence value yielded by DWAL can provide rich information beyond label supervision, which can instruct the classifier to learn unlabeled data well. In addition, by observing line (1), we can draw several interesting observations. In *Case* 1, its performance is very close to that of DWAL method (see Fig. 3). This is mainly because, in *Case* 1, the real proportion of examples of $\mathcal{D}_{pri}$ in $\mathcal{D}_l$ is low and the negative influence caused by private data is small. From *Case* 1 to *Case* 4, the proportion improves, so does the contribution of dissimilarity maximization. Though dissimilarity maximization cannot get weight directly, it can weaken the negative influence caused by private data and improve the accuracy of weight calculated by DWAL. Last but not least, line (5) deserves our attention as well. When we replace C' with C, we involve C in the calculation of weight in fact. We can find that line (5) suffers a large performance drop as well, which confirms the disastrous outcome when combining the detection part and classification part together, and our DWAL method can solve this problem.

5.5 Performance Verification

We will show visualization of transferability score, visualization of prediction confidence value, and effects of z.

5.5.1 Visualization of Transferability Scores. To confirm the reasonability of transferability scores, we visualize the transferability scores of labeled data and unlabeled data on three different datasets in Fig. 4. The results show that, the distribution of transferability score of shared data is different from that of other data, and the averaged transferability score of shared data is more likely to be close to 1 than that of other data, in both labeled and unlabeled sets. The transferability score provides valuable information in distinguishing shared data from

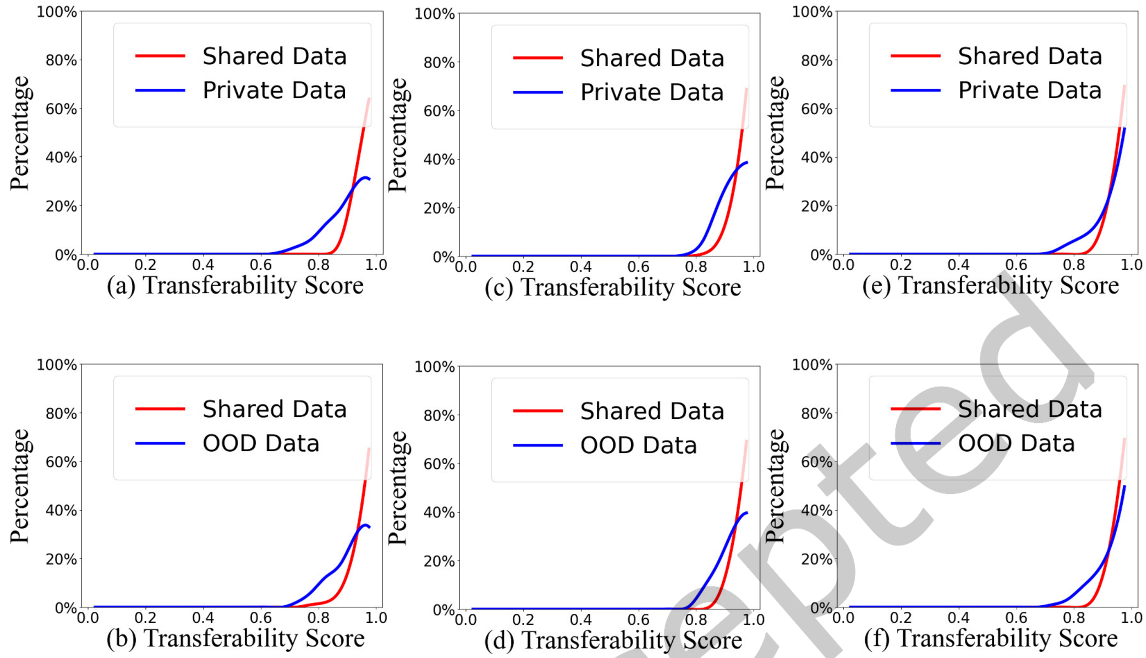

Fig. 4. Transferability score of labeled data and unlabeled data in three datasets: (a): Labeled data in MNIST; (b) Unlabeled data in MNIST; (c): Labeled data in SVHN; (d) Unlabeled data in SVHN; and (e): Labeled data in CIFAR-10; and (f) Unlabeled data in CIFAR-10.

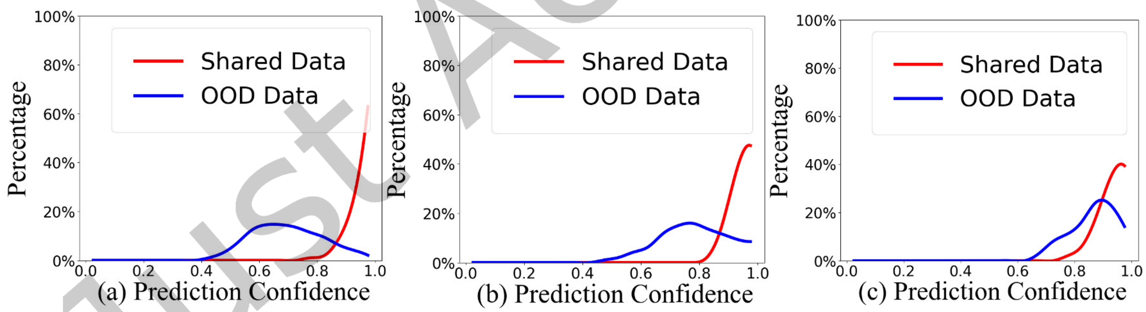

Fig. 5. Visualization of prediction confidence value on three datasets: (a) MNIST, (b) SVHN; and (c) CIFAR-10.

private data in $\mathcal{D}_l$, and shared data from OOD data in $\mathcal{D}_u$. This explains how transferability score improves the performance under intersectional class-mismatch setting.

5.5.2 Visualization of Prediction Confidence Value. To show the reliability of prediction confidence value, we visualize the prediction confidence value of unlabeled data on three different datasets in Fig. 5. The results show that, the distribution of prediction confidence value of shared data is different from that of OOD data, and the expectation of prediction confidence value of shared data is larger than that of OOD data. This confirms that the way we calculate prediction confidence value is effective and explains why prediction confidence value can instruct classifier to learn from unlabeled data.

Table 7. Classification accuracies of z with different coefficients on SVHN.

Value of z	*Case* 1	*Case* 2	*Case* 3	*Case* 4
$0.5 * z$	86.19 ± 0.21	85.30 ± 0.17	83.86 ± 0.29	82.81 ± 0.26
$0.9 * z$	86.16 ± 0.27	85.81 ± 0.24	85.15 ± 0.32	84.94 ± 0.37
$1.0 * z$	86.12 ± 0.34	85.76 ± 0.37	85.21 ± 0.26	84.98 ± 0.33
$1.1 * z$	86.10 ± 0.35	85.56 ± 0.42	85.19 ± 0.39	85.01 ± 0.35
$1.5 * z$	86.06 ± 0.35	85.03 ± 0.36	84.09 ± 0.37	83.15 ± 0.46

5.5.3 Effects of z. According to Eq. (12), we can detect private data by comparing the transferability scores of labeled examples with z, where z is the average transferability score of labeled examples. To investigate the effects brought by value of z, we set different values of z by letting the averaged transferability score multiply the range of coefficients (*i.e.*, 0.5, 0.9, 1.0, 1.1 and 1.5), and then we have conducted an experiment on SVHN from *Case* 1 to *Case* 4. The accuracies are shown in Table 7. From Table 7, it can be seen that when z fluctuates slightly between 0.9 and 1.1, the results change mildly, but when z fluctuates critically to 0.5 or 1.5, the accuracy suffers degradation. This indicates dissimilarity maximization is not sensitive to the value of z when it is close to the averaged transferability score of labeled data.

6 CONCLUSION

In this paper, we propose a novel DWAL method to tackle SSL under intersectional class mismatch. Specifically, we conduct enhanced adversarial domain adaptation to reliably calculate weight from the perspectives of domain adaptation and a class-wise weighting mechanism, so that we can weigh each example. After detecting the private data from labeled data, dissimilarity maximization is proposed to maximize the disagreement between the feature distribution of private data and that of unlabeled data, reducing the inaccurate adaptation caused by private data. More importantly, we separately conduct the detection part and classification part to avoid a conflicting optimization. Experimental results show that our DWAL performs better than other state-of-the-art SSL methods under intersectional class-mismatch setting.

In our method, the private examples in labeled set are decided according to the threshold z in Eq. (12), where z is simply the average number of transferbility scores of labeled data. In the future, we plan to find a more advanced way to adaptively decide this threshold, so that the private examples can be reliably detected.

REFERENCES

[1] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *International Conference on Machine Learning*. 41–48.

[2] David Berthelot, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. 2019. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. In *International Conference on Learning Representations.*

[3] Kaidi Cao, Maria Brbic, and Jure Leskovec. 2022. Open-World Semi-Supervised Learning. In *International Conference on Learning Representations.*

[4] Zhangjie Cao, Lijia Ma, Mingsheng Long, and Jianmin Wang. 2018. Partial adversarial domain adaptation. In *European Conference on Computer Vision.* 135–150.

[5] Zhangjie Cao, Kaichao You, Mingsheng Long, Jianmin Wang, and Qiang Yang. 2019. Learning to transfer examples for partial domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition.* 2985–2994.

[6] Paola Cascante-Bonilla, Fuwen Tan, Yanjun Qi, and Vicente Ordonez. 2021. Curriculum labeling: Revisiting pseudo-labeling for semi-supervised learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 6912–6920.

[7] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. 2009. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks and Learning Systems* 20, 3 (2009), 542–542.

[8] Yanbei Chen, Xiatian Zhu, Wei Li, and Shaogang Gong. 2020. Semi-supervised learning under class distribution mismatch. In *International Conference on Machine Learning*, Vol. 34. 3569–3576.

[9] Patryk Chrabaszcz, Ilya Loshchilov, and Frank Hutter. 2017. A downsampled variant of imagenet as an alternative to the cifar datasets. *arXiv preprint arXiv:1707.08819* (2017).

[10] Shuhao Cui, Xuan Jin, Shuhui Wang, Yuan He, and Qingming Huang. 2020. Heuristic domain adaptation. In *Advances in Neural Information Processing Systems*, Vol. 33. 7571–7583.

[11] Shuhao Cui, Shuhui Wang, Junbao Zhuo, Liang Li, Qingming Huang, and Qi Tian. 2020. Towards discriminability and diversity: Batch nuclear-norm maximization under label insufficient situations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3941–3950.

[12] Zihang Dai, Zhilin Yang, Fan Yang, William W Cohen, and Russ R Salakhutdinov. 2017. Good semi-supervised learning that requires a bad gan. In *Advances in Neural Information Processing Systems*, Vol. 30.

[13] Jinhao Dong and Tong Lin. 2019. Margingan: Adversarial training in semi-supervised learning. In *Advances in Neural Information Processing Systems*, Vol. 32.

[14] Sichao Fu, Weifeng Liu, Weili Guan, Yicong Zhou, Dapeng Tao, and Changsheng Xu. 2021. Dynamic graph learning convolutional networks for semi-supervised classification. *ACM Transactions on Multimedia Computing, Communications, and Applications* 17, 1s (2021), 1–13.

[15] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research* 17, 1 (2016), 2096–2030.

[16] Chen Gong, Dacheng Tao, Keren Fu, and Jie Yang. 2014. Fick's law assisted propagation for semisupervised learning. *IEEE Transactions on Neural Networks and Learning Systems* 26, 9 (2014), 2148–2162.

[17] Chen Gong, Dacheng Tao, Stephen J Maybank, Wei Liu, Guoliang Kang, and Jie Yang. 2016. Multi-modal curriculum learning for semi-supervised image classification. *IEEE Transactions on Image Process* 25, 7 (2016), 3249–3260.

[18] Chengyue Gong, Dilin Wang, and Qiang Liu. 2021. AlphaMatch: Improving Consistency for Semi-supervised Learning with Alpha-divergence. In *IEEE Conference on Computer Vision and Pattern Recognition*. 13683–13692.

[19] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, Vol. 27.

[20] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2020. Generative adversarial networks. *Commun. ACM* 63, 11 (2020), 139–144.

[21] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* (2014).

[22] Yves Grandvalet, Yoshua Bengio, et al. 2005. Semi-supervised learning by entropy minimization.. In *Advances in Neural Information Processing Systems*, Vol. 367. 281–296.

[23] Lan-Zhe Guo, Yi-Ge Zhang, Zhi-Fan Wu, Jie-Jing Shao, and Yu-Feng Li. 2022. Robust Semi-Supervised Learning when Not All Classes have Labels. In *Advances in Neural Information Processing Systems*, Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (Eds.).

[24] Lan-Zhe Guo, Zhen-Yu Zhang, Yuan Jiang, Yu-Feng Li, and Zhi-Hua Zhou. 2020. Safe deep semi-supervised learning for unseen-class unlabeled data. In *International Conference on Machine Learning*. 3897–3906.

[25] Junkai Huang, Chaowei Fang, Weikai Chen, Zhenhua Chai, Xiaolin Wei, Pengxu Wei, Liang Lin, and Guanbin Li. 2021. Trash to Treasure: Harvesting OOD Data with Cross-Modal Matching for Open-Set Semi-Supervised Learning. In *IEEE International Conference on Computer Vision*. 8310–8319.

[26] Zhuo Huang, Li Shen, Jun Yu, Bo Han, and Tongliang Liu. 2023. FlatMatch: Bridging Labeled Data and Unlabeled Data with Cross-Sharpness for Semi-Supervised Learning. In *Advances in Neural Information Processing Systems*.

[27] Zhuo Huang, Xiaobo Xia, Li Shen, Bo Han, Mingming Gong, Chen Gong, and Tongliang Liu. 2023. Harnessing out-of-distribution examples via augmenting content and style. In *ICLR*.

[28] Zhuo Huang, Jian Yang, and Chen Gong. 2022. They are Not Completely Useless: Towards Recycling Transferable Unlabeled Data for Class-Mismatched Semi-Supervised Learning. *IEEE Transactions on Multimedia* (2022).

[29] T. Joachims. 1999. Transductive Inference for Text Classification using Support Vector Machines. In *International Conference on Machine Learning*. Bled, Slowenien, 200–209.

[30] T. Joachims. 2003. Transductive Learning via Spectral Graph Partitioning. In *International Conference on Machine Learning*. 290–297.

[31] Zhanghan Ke, Daoye Wang, Qiong Yan, Jimmy Ren, and Rynson WH Lau. 2019. Dual student: Breaking the limits of the teacher in semi-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 6728–6736.

[32] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. In *Conference and Workshop on Neural Information Processing Systems*, Vol. 33. 18661–18673.

[33] Vladimir Koltchinskii and Dmitry Panchenko. 2002. Empirical margin distributions and bounding the generalization error of combined classifiers. *Annals of Statistics* 30, 1 (2002), 1–50.

[34] Abhishek Kumar, Prasanna Sattigeri, and Tom Fletcher. 2017. Semi-supervised learning with gans: Manifold invariance with improved inference. In *Advances in Neural Information Processing Systems*, Vol. 30.

[35] Samuli Laine and Timo Aila. 2016. Temporal ensembling for semi-supervised learning. In *International Conference on Learning Representations*.
[36] Dong-Hyun Lee et al. 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *International Conference on Machine Learning*, Vol. 3. 896.
[37] Chongxuan Li, Taufik Xu, Jun Zhu, and Bo Zhang. 2017. Triple generative adversarial nets. In *Advances in Neural Information Processing Systems*, Vol. 30.
[38] Yang Li, Zhiqun Zhao, Hao Sun, Yigang Cen, and Zhihai He. 2020. Snowball: Iterative model evolution and confident sample discovery for semi-supervised learning on very small labeled datasets. *IEEE Transactions on Multimedia* 23 (2020), 1354–1366.
[39] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. 2018. Conditional adversarial domain adaptation. In *Advances in Neural Information Processing Systems*, Vol. 31.
[40] Takeru Miyato, Shin-Ichi Maeda, Masanori Koyama, and Shin Ishii. 2018. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41, 8 (2018), 1979–1993.
[41] Bernice D Mowery. 2011. The paired t-test. *Pediatric Nursing* 37, 6 (2011), 320–322.
[42] Islam Nassar, Samitha Herath, Ehsan Abbasnejad, Wray Buntine, and Gholamreza Haffari. 2021. All labels are not created equal: Enhancing semi-supervision via label grouping and co-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7241–7250.
[43] Youngtaek Oh, Dong-Jin Kim, and In So Kweon. 2022. DASO: Distribution-Aware Semantics-Oriented Pseudo-Label for Imbalanced Semi-Supervised Learning. In *IEEE Conference on Computer Vision and Pattern Recognition*. 9786–9796.
[44] Avital Oliver, Augustus Odena, Colin Raffel, Ekin D Cubuk, and Ian J Goodfellow. 2018. Realistic evaluation of deep semi-supervised learning algorithms. In *Advances in Neural Information Processing Systems*. 3239–3250.
[45] Sungrae Park, JunKeon Park, Su-Jin Shin, and Il-Chul Moon. 2018. Adversarial dropout for supervised and semi-supervised learning. In *Association for the Advance of Artificial Intelligence*.
[46] Mamshad Nayeem Rizve, Kevin Duarte, Yogesh S Rawat, and Mubarak Shah. 2021. In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning. In *International Conference on Learning Representations*.
[47] Kuniaki Saito, Donghyun Kim, and Kate Saenko. 2021. OpenMatch: Open-set Consistency Regularization for Semi-supervised Learning with Outliers. In *Advances in Neural Information Processing Systems*. 25956–25967.
[48] Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. 2016. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In *Advances in Neural Information Processing Systems*, Vol. 29. 1163–1171.
[49] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. 2020. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *Advances in Neural Information Processing Systems*. 596–608.
[50] Teppei Suzuki and Ikuro Sato. 2020. Adversarial Transformations for Semi-Supervised Learning. In *International Conference on Machine Learning*, Vol. 34. 5916–5923.
[51] Antti Tarvainen and Harri Valpola. 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *International Conference on Learning Representations*.
[52] Wenjing Wang, Lilang Lin, Zejia Fan, and Jiaying Liu. 2023. Semi-Supervised Learning for Mars Imagery Classification and Segmentation. *ACM Transactions on Multimedia Computing, Communications and Applications* 19, 4 (2023), 1–23.
[53] Yunyun Wang, Songcan Chen, and Zhi-Hua Zhou. 2012. New semi-supervised classification method based on modified cluster assumption. *IEEE Transactions on Neural Networks and Learning Systems* 23, 5 (2012), 689–702.
[54] Yu Wang, Pengchong Qiao, Chang Liu, Guoli Song, Xiawu Zheng, and Jie Chen. 2023. Out-of-Distributed Semantic Pruning for Robust Semi-Supervised Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 23849–23858.
[55] Yingda Xia, Fengze Liu, Dong Yang, Jinzheng Cai, Lequan Yu, Zhuotun Zhu, Daguang Xu, Alan Yuille, and Holger Roth. 2020. 3d semi-supervised learning with uncertainty-aware multi-view co-training. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 3646–3655.
[56] Jiayu Xiao, Liang Li, Chaofei Wang, Zheng-Jun Zha, and Qingming Huang. 2022. Few shot generative model adaption via relaxed spatial structural alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 11204–11213.
[57] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. 2020. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10687–10698.
[58] Fan Yang, Kai Wu, Shuyi Zhang, Guannan Jiang, Yong Liu, Feng Zheng, Wei Zhang, Chengjie Wang, and Long Zeng. 2022. Class-Aware Contrastive Semi-Supervised Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 14421–14430.
[59] Yueming Yin, Zhen Yang, Haifeng Hu, and Xiaofu Wu. 2020. Universal Multi-Source Domain Adaptation. *Pattern Recognition* (2020).
[60] Kaichao You, Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. 2019. Universal domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2720–2729.

[61] Qing Yu, Daiki Ikami, Go Irie, and Kiyoharu Aizawa. 2020. Multi-task curriculum framework for open-set semi-supervised learning. In *European Conference on Computer Vision*. Springer, 438–454.

[62] Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. 2021. FlexMatch: Boosting semi-supervised learning with curriculum pseudo labeling. In *Advances in Neural Information Processing Systems*, Vol. 34. 18408–18419.

[63] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. 2017. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*.

[64] Yuchen Zhang, Tianle Liu, Mingsheng Long, and Michael Jordan. 2019. Bridging theory and algorithm for domain adaptation. In *International Conference on Machine Learning*. 7404–7413.

[65] Yuhang Zhang, Xiaopeng Zhang, Jie Li, Robert Qiu, Haohang Xu, and Qi Tian. 2022. Semi-supervised contrastive learning with similarity co-calibration. *IEEE Transactions on Multimedia* (2022).

[66] Jian Zhao, Xianhui Liu, and Weidong Zhao. 2022. Balanced and Accurate Pseudo-labels for Semi-supervised Image Classification. *ACM Transactions on Multimedia Computing, Communications and Applications* 18, 3s (2022), 1–18.

[67] Zhi-Hua Zhou and Ming Li. 2005. Tri-training: Exploiting unlabeled data using three classifiers. *IEEE Transactions on Knowledge and Data Engineering* 17, 11 (2005), 1529–1541.

[68] Bowei Zhu and Yong Liu. 2021. General approximate cross validation for model selection: Supervised, semi-supervised and pairwise learning. In *ACM International Conference on Multimedia*. 5281–5289.

[69] Hui Zhu, Yongchun Lu, Hongbin Wang, Xunyi Zhou, Qin Ma, Yanhong Liu, Ning Jiang, Xin Wei, Linchengxi Zeng, and Xiaofang Zhao. 2022. Enhancing Semi-Supervised Learning with Cross-Modal Knowledge. In *ACM International Conference on Multimedia*. 4456–4465.