Bidirectional Adaptation for Robust Semi-Supervised Learning with Inconsistent Data Distributions

Lin-Han Jia [1] **Lan-Zhe Guo** [1] **Zhi Zhou** [1] **Jie-Jing Shao** [1] **Yu-Ke Xiang** [2] **Yu-Feng Li** [1]

Abstract

Semi-supervised learning (SSL) suffers from severe performance degradation when labeled and unlabeled data come from inconsistent data distributions. However, there is still a lack of sufficient theoretical guidance on how to alleviate this problem. In this paper, we propose a general theoretical framework that demonstrates how distribution discrepancies caused by pseudo-label predictions and target predictions can lead to severe generalization errors. Through theoretical analysis, we identify three main reasons why previous SSL algorithms cannot perform well with inconsistent distributions: coupling between the pseudo-label predictor and the target predictor, biased pseudo labels, and restricted sample weights. To address these challenges, we introduce a practical framework called Bidirectional Adaptation that can adapt to the distribution of unlabeled data for debiased pseudo-label prediction and to the target distribution for debiased target prediction, thereby mitigating these shortcomings. Extensive experimental results demonstrate the effectiveness of our proposed framework.

1. Introduction

Semi-supervised learning (SSL) is a promising learning paradigm that seeks to overcome the scarcity of labeled data by leveraging an abundance of unlabeled data (Chapelle et al., 2006; Oliver et al., 2018). In recent years, SSL research has received extensive attention and made significant progress. Thanks to its ability to handle both labeled and unlabeled data, SSL has found successful applications in various tasks, such as image classification (Sohn et al., 2020), object detection (Jeong et al., 2019), semantic segmentation (Souly et al., 2017), and text classification (Miyato et al., 2017), among others. It is expected that, when labeled data are limited, the use of unlabeled data will help improve the performance. However, it has been found that SSL algorithms perform even worse than using only labeled data (Ben-David et al., 2008; Li & Zhou, 2014; Li et al., 2021). Especially when unlabeled data are sampled from a different distribution than labeled data and target data, SSL algorithms will suffer from severe performance degradation (Oliver et al., 2018; Guo et al., 2020; Yu et al., 2020; Huang et al., 2021). Unfortunately, inconsistent distributions are a common challenge in real-world applications. For example, labeled and unlabeled data are collected from different sources; labeled data are real samples while unlabeled data are synthetic ones (Peng et al., 2018); labeled data are obtained through prioritized manual labeling, possibly from the parts of unlabeled data that are more likely to contribute to the learning objective (Shao et al., 2022b;a).The scope and value of SSL will be greatly enhanced if unlabeled data from other distributions can be effectively utilized. In this paper, we focus on SSL where the distributions of labeled and unlabeled data are inconsistent. Due to the lack of labels, we can only observe covariate shift $p_L(x) \neq p_U(x)$. In fact, when the learning target $p(y|x)$ remains constant, covariate shift is equivalent to the combination of label distribution shift $p_L(y) \neq p_U(y)$ and intra-class feature distribution shift $p_L(x|y) \neq p_U(x|y)$ which cannot be observed independently (as shown in Figure 1).

It is evident that neither classical SSL nor Domain Adaptation (DA) can tackle this problem well. Classical SSL typically assumes that all samples are drawn from the same distribution. From a theoretical point of view, various works formalize the idea of using unlabeled data and subsequently investigate situations where unlabeled data cannot help or where it can (Leskes, 2005; Ben-David et al., 2008; Balcan & Blum, 2010; Peters et al., 2017), but they are limited in scope to provide practical guidance for designing algorithms to address the problem of inconsistent distributions. From a practical point of view, there have been some works related to the problem of inconsistent distributions (Oliver et al., 2018; Guo et al., 2020; Chen et al., 2020; Huang et al., 2021), but most of them only focus on label distributions

[1]National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China [2]Consumer BG, Huawei Technologies, Shenzhen, China. Correspondence to: Yu-Feng Li <liyf@lamda.nju.edu.cn>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

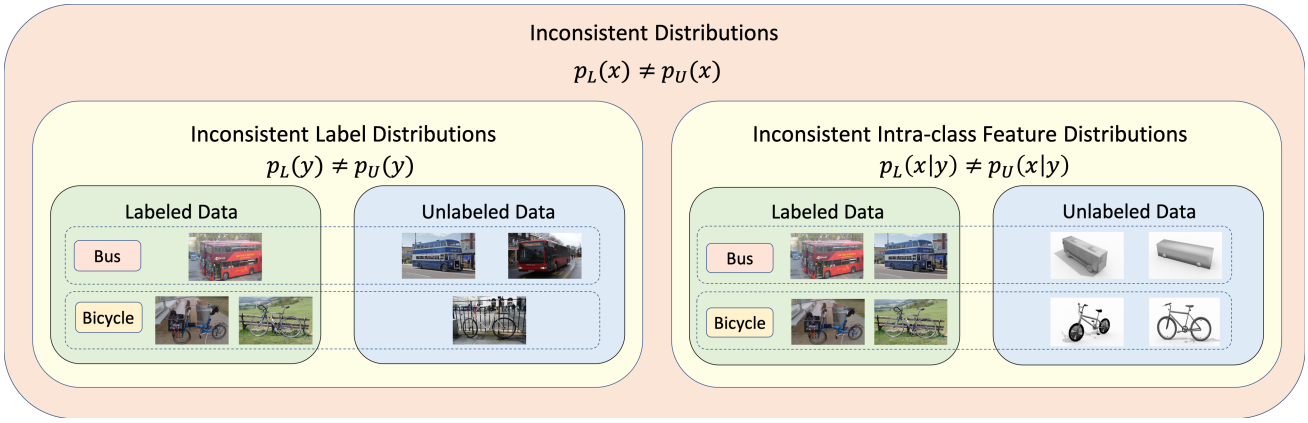

Figure 1. In SSL, when the learning target $p(y|x)$ remains constant, covariate shift $p_L(x) \neq p_U(x)$ is equivalent to a combination of unobserved label distribution shift $p_L(y) \neq p_U(y)$ and intra-class feature distribution shift $p_L(x|y) \neq p_U(x|y)$.

and lack theoretical guidance. Many works in the field of DA focus on the problem of the distribution drift between two domains (Glorot et al., 2011; Long et al., 2015; Ganin & Lempitsky, 2015; Zhu et al., 2020), and they pay more attention to the model's performance on the unlabeled target domain which is similar to transductive SSL, and the model after adaptation usually performs worse on the source domain than before adaptation, so they cannot be used for inductive SSL directly.

In this paper, our focus is on using unlabeled data from distributions that differ from that of labeled data to enhance the performance of SSL algorithms. However, previous theoretical frameworks have some limitations, prompting us to establish a new one that demonstrates the generalization error's relation to two terms of distribution discrepancy. The first term is caused by pseudo-label prediction because the pseudo-label predictor is trained with the distribution of labeled data but applied to the distribution of unlabeled data. The second term is caused by target prediction because the target predictor is trained with the mixture distribution of labeled data and weighted unlabeled data but applied to the target distribution. Our theoretical framework yields a general optimization objective and enables us to identify three main shortcomings of previous SSL algorithms: coupling between the pseudo-label predictor and the target predictor, biased pseudo-labels, and restricted sample weights. To overcome these shortcomings, we propose Bidirectional Adaptation which decouples the pseudo-label predictor and the target predictor, adapts to the distribution of unlabeled data for the debiased pseudo-label predictor and adapts to the target distribution for the debiased target predictor. A vast number of experiments confirm the correctness of our theoretical analysis and the efficacy of our proposed method.

Our Contributions. Our contributions are threefold. Firstly, we introduce a novel theoretical framework that provides guidance on addressing inconsistent distributions in SSL. Secondly, we identify three main shortcomings of previous SSL algorithms through formalization. Thirdly, we propose a practical framework that is robust to inconsistent distributions and demonstrate its effectiveness through a large number of experiments.

2. Related Works

2.1. Robust SSL

To apply SSL techniques to wider applications, there is an urgent need to study robust SSL methods that do not suffer severe performance degradation when unlabeled data is corrupted. Chen et al. (Chen et al., 2020) investigate the SSL scenario with inconsistent label distributions between labeled and unlabeled data. Oliver et al. (Oliver et al., 2018) demonstrate that using unlabeled data from unseen classes can actually hurt performance compared to not using any unlabeled data at all. Guo et al. (Guo et al., 2020; Guo & Li, 2022) propose robust SSL algorithms for unseen-class unlabeled data. The above works only focus on inconsistent label distributions ignoring intra-class feature distributions. Huang et al. (Huang et al., 2021) propose an algorithm that aims to solve the problem where both label distributions and intra-class feature distributions are inconsistent, but this work doesn't provide theoretical analysis.

2.2. DA

DA (Farahani et al., 2021) is a sub-field within transfer learning that aims to cope with the discrepancy of distributions across domains such that the trained model can be generalized into the domain of interest. DA methods align the distributions by minimizing the distance between distribu-

tions such as maximum mean discrepancy (MMD) (Gretton et al., 2006), Wasserstein metric, and contrastive domain discrepancy (CDD) (Kang et al., 2019). Weight-based methods (Huang et al., 2006; Sugiyama et al., 2007) make two distributions closer by assigning higher weights to samples close to both distributions and vice versa. Feature-based methods (Gopalan et al., 2011; Gong et al., 2012) map samples from different distributions to a consistent distribution in a feature space directly. There are also many deep DA methods (Glorot et al., 2011; Long et al., 2015; Ganin & Lempitsky, 2015; Zhu et al., 2020) use neural networks to diminish the domain gap. After adaptation, the trained model typically achieves better performance on the target domain but may perform worse on the source domain.

2.3. Theory of SSL

From negative perspectives, many works (Seeger, 2000; Wasserman & Lafferty, 2007; Quinonero-Candela et al., 2008) prove that when there are no assumptions in SSL, $p(x)$ which contains no information about $p(y|x)$ can not have any impact on the learning process of $p(y|x)$, and unlabeled data is useless. Ben et al. (Ben-David et al., 2008) prove that a semi-supervised learner cannot have essentially better sample complexity bounds than a supervised learner without effective assumptions. From positive perspectives, many works (Sinha & Belkin, 2007; Rigollet, 2007; Singh et al., 2008; Wei et al., 2020; Sanz-Alonso & Yang, 2022) prove that SSL can perform well with assumptions about $p(y|x)$. Leskes (Leskes, 2005) presents that unlabeled data can help to obtain a new hypothesis space which is a subset of the original hypothesis class. Wang et al. (Wang et al., 2022) study the distribution bias between labeled and unlabeled data but only consider the bias caused by sampling. Most of the previous theoretical works are strongly restrictive and cannot provide sufficient guidance for algorithm design in real-world applications.

3. Theoretical Results

3.1. Problem Setting and Notations

Let $\mathcal{X} \subseteq \mathbb{R}^d$ be the feature space and $\mathcal{Y} \subseteq \{0, \cdots, k-1\}$ be the label space. In SSL, we are given n_l labeled samples $D_L = \{(x_i, y_i)|(x_i, y_i) \sim p_L(x, y)\}_{i=1}^{n_l}$ from $p_L(x, y)$ which is the joint distribution of labeled data and n_u unlabeled samples $D_U = \{(x_i)|x_i \sim p_U(x)\}_{i=1}^{n_u}$ from $p_U(x)$ which is the distribution of unlabeled data. The purpose of SSL is to learn a predictor with the smallest generalization error on the target distribution $p_T(x, y)$. In this paper, we assume that $p_L(x, y) = p_T(x, y)$ and $p_L(x) \neq p_U(x)$ which is equivalent to the combination of unobserved $p_L(y) \neq p_U(y)$ and $p_L(x|y) \neq p_U(x|y)$ for the constant learning target $p(y|x)$.

Natarajan dimension (Natarajan, 1989; Ben-David et al., 1992) is an extension of Vapnik-Chervonen dimension (Vapnik & Chervonenkis, 1971) in multi-classification problems. We denote $Ndim(\mathcal{H})$ the Natarajan dimension of a hypothesis space $\mathcal{H}$. To simplify the expression, we denote the variance term associated with the hypothesis space complexity in the generalization error with the number of samples n, the number of classes k, and the probability δ:

$$var(\mathcal{H}, n, k, \delta) = \sqrt{\frac{16Ndim(\mathcal{H}) \ln \sqrt{2n}k + 8 \ln \frac{2}{\delta}}{n}} \quad (1)$$

We denote the mixture of two distributions $\mathcal{D}_1$ and $\mathcal{D}_2$ with proportion α as:

$$Mix_\alpha(\mathcal{D}_1, \mathcal{D}_2) = \alpha\mathcal{D}_1 + (1-\alpha)\mathcal{D}_2 \quad (2)$$

The discrepancy between two distributions $\mathcal{D}_1$ and $\mathcal{D}_2$ corresponding to a predictor f can be denoted as:

$$\begin{aligned} Disc(f, \mathcal{D}_1, \mathcal{D}_2) = |&p_{x,y\sim\mathcal{D}_1}(f(x) \neq y) \\ &-p_{x,y\sim\mathcal{D}_2}(f(x) \neq y)| \end{aligned} \quad (3)$$

3.2. Foundation of SSL Theoretical Framework.

The objective of SSL is to learn a predictor f from a hypothesis space $\mathcal{F}$ that minimizes the generalization error on the target distribution. The assumption in SSL provides prior knowledge about the relationship between $p(x)$ and $p(y|x)$. Previous SSL theoretical frameworks use unlabeled data to exclude unimportant functions in $\mathcal{F}$ (Leskes, 2005) or map the original hypothesis space $\mathcal{F}$ to a new one (Balcan & Blum, 2010). However, these theoretical frameworks cannot provide sufficient guidance for practice. Moreover, these frameworks have limitations since they assume that all samples come from the same distribution.

We propose a new theoretical framework, which is applicable to current mainstream SSL algorithms and can analyze the situation where the distributions of labeled and unlabeled data are inconsistent. It is known that there is an assumption and a base learner in SSL. The assumption can produce the prior knowledge about $p(y|x)$ which can be used to provide pseudo-labels explicitly or implicitly as additional supervision information. The base learner can use the original and additional supervision information to learn the final $p(y|x)$ for prediction. We find that the purposes of both the assumption and the base learner are to fit $p(y|x)$, but there must be inconsistency between the results of them. Therefore, we regard the process of learning prior knowledge from the assumption using labeled and unlabeled data as the process of learning a pseudo-label predictor from a hypothesis space. Formally, there is an assumption $\mathcal{A}$ in SSL which can use D_L and D_U to learn a function h as a pseudo-label predictor from the hypothesis space $\mathcal{H}$. This predictor can help to obtain unlabeled dataset with pseudo-labels, denoted as $\tilde{D}_U = \{(X_1^U, \tilde{y}_1^U), (X_2^U, \tilde{y}_2^U), \ldots, (X_{n_u}^U, \tilde{y}_{n_u}^U)\}$.

3.3. SSL with Consistent Distribution

We first consider a simple case where the distributions of labeled data and unlabeled data are consistent, the error of the pseudo-label predictor on unlabeled data can be estimated by the error on labeled data mediated by generalization error on the consistent data distribution.

Theorem 3.1. *For any pseudo-label predictor $h \in \mathcal{H}$, $0 \leq \delta_1 \leq 1$ and $0 \leq \delta_2 \leq 1$, with the probability of at least $(1-\delta_1)(1-\delta_2)$:*

$$\begin{aligned}\hat{E}(h, D_U) \leq & \hat{E}(h, D_L) + var(\mathcal{H}, n_l, k, \delta_1) \\ & + var(\mathcal{H}, n_u, k, \delta_2)\end{aligned} \tag{4}$$

where $\hat{E}(h, D_L)$ is the empirical error of h on D_L and $\hat{E}(h, D_U)$ is the empirical error of h on D_U with ground truth labels.

The above generalization error bound is also the upper bound of the label noise rate of unlabeled data with pseudo-labels. Considering a naive SSL algorithm, that is, the base learner uses both labeled data D_L and noisy unlabeled data $\tilde{D}_U$ for learning, then we have,

Theorem 3.2. *For any target predictor $f \in \mathcal{F}$, pseudo-label predictor $h \in \mathcal{H}$, $0 \leq \delta_1 \leq 1$, $0 \leq \delta_2 \leq 1$ and $0 \leq \delta_3 \leq 1$, with the probability of at least $(1-\delta_1)(1-\delta_2)(1-\delta_3)$:*

$$\begin{aligned}& E(f, \mathcal{D}_T | h, D_L, D_U) \leq \frac{n_l}{n_l + n_u} \hat{E}(f, D_L) \\ & + \frac{n_u}{n_l + n_u} \hat{E}(f, \tilde{D}_U) + var(\mathcal{F}, n_l + n_u, k, \delta_1) \\ & + \frac{n_u}{n_l + n_u} (\hat{E}(h, D_L) + var(\mathcal{H}, n_l, k, \delta_2) \\ & + var(\mathcal{H}, n_u, k, \delta_3))\end{aligned} \tag{5}$$

where $E(f, \mathcal{D}_T | h, D_L, D_U)$ is the generalization error of f on the distribution $p_T(x, y)$ corresponding to pseudo-label predictor h, $\hat{E}(f, D_L)$ is the empirical error of the target predictor f on the dataset D_L and $\hat{E}(f, \tilde{D}_U)$ is the disagreement rate between the noisy pseudo-labels and the prediction results of f on the unlabeled dataset $\tilde{D}_U$.

3.4. SSL with Inconsistent Distributions

When the distributions of labeled data and unlabeled data are inconsistent, it requires additional consideration of the distribution discrepancy estimating the label noisy rate of unlabeled data with pseudo-labels.

Theorem 3.3. *For any pseudo-label predictor $h \in \mathcal{H}$, $0 \leq \delta_1 \leq 1$ and $0 \leq \delta_2 \leq 1$, with the probability of at least $(1-\delta_1)(1-\delta_2)$:*

$$\begin{aligned}\hat{E}(h, D_U) \leq & \hat{E}(h, D_L) + var(\mathcal{H}, n_l, k, \delta_1) \\ & + var(\mathcal{H}, n_u, k, \delta_2) + Disc(h, \mathcal{D}_L, \mathcal{D}_U)\end{aligned} \tag{6}$$

SSL algorithms typically require selection or weighting of unlabeled samples using a weighting function $w : \mathcal{X} \to \mathbb{R}$, which increases the influence of beneficial samples and reduces the influence of harmful samples. In classical SSL algorithms, w is often an indicator function with a threshold based on the confidence of the predictor. We denote the weighted unlabeled dataset with noisy pseudo-labels as $\tilde{D}_U^w = w(\tilde{D}_U)$ and the sum of weights of all unlabeled samples as $n_u^w = \sum_{(x,y) \in D_U} w(x)$.

In the case of inconsistent distributions, SSL algorithms use both D_L and $\tilde{D}_U^w$ for training but need to be tested on the target distribution. This inconsistency can lead to additional errors in target predictions.

Theorem 3.4. *Assuming that the probabilities of the pseudo-label predictor making wrong predictions for each sample are equal without considering the difference among them, for any target predictor $f \in \mathcal{F}$, pseudo-label predictor $h \in \mathcal{H}$, $0 \leq \delta_1 \leq 1$, $0 \leq \delta_2 \leq 1$ and $0 \leq \delta_3 \leq 1$, with the probability of at least $(1-\delta_1)(1-\delta_2)(1-\delta_3)$:*

$$\begin{aligned}& E(f, \mathcal{D}_T | h, D_L, D_U) \leq \frac{n_l}{n_l + n_u^w} \hat{E}(f, D_L) \\ & + \frac{n_u^w}{n_l + n_u^w} \hat{E}(f, \tilde{D}_U^w) + var(\mathcal{F}, n_l + n_u^w, k, \delta_1) \\ & + Disc(f, \mathcal{D}_T, Mix_{\frac{n_l}{n_l + n_u^w}}(\mathcal{D}_L, \mathcal{D}_U^w)) \\ & + \frac{n_u^w}{n_l + n_u^w} (\hat{E}(h, D_L) + var(\mathcal{H}, n_l, k, \delta_2) \\ & + var(\mathcal{H}, n_u, k, \delta_3) + Disc(h, \mathcal{D}_L, \mathcal{D}_U))\end{aligned} \tag{7}$$

where $\hat{E}(f, \tilde{D}_U^w)$ is the weighted disagreement rate between the noisy pseudo-labels and the prediction results of f on the unlabeled dataset $\tilde{D}_U$.

4. Analysis of SSL Algorithms

The objective of SSL is to minimize the generalization error of the target predictor on the target distribution $E(f, \mathcal{D}_T | h, D_L, D_U)$. Based on the theoretical results presented above, we can get the optimization objective without considering the complexity of hypothesis spaces:

$$\begin{aligned}& \min_{f \in \mathcal{F}, h \in \mathcal{H}} [\frac{n_l}{n_l + n_u^w} \hat{E}(f, D_L) + \frac{n_u^w}{n_l + n_u^w} \hat{E}(f, \tilde{D}_U) \\ & + Disc(f, \mathcal{D}_T, Mix_{\frac{n_l}{n_l + n_u^w}}(\mathcal{D}_L, \mathcal{D}_U^w)) \\ & + \frac{n_u^w}{n_l + n_u^w} \hat{E}(h, D_L) + \frac{n_u^w}{n_l + n_u^w} Disc(h, \mathcal{D}_L, \mathcal{D}_U)]\end{aligned} \tag{8}$$

We provide an overview of how past SSL algorithms have attempted to optimize this objective and identify three shortcomings in their attempts to optimize it effectively.

4.1. Formalization of SSL Algorithms

Classical deep SSL algorithms mainly have two strategies: pseudo-labeling and consistency regularization, and the combination of them can often achieve better results. We find that in both strategies, the pseudo-label predictor h is a modification of the target predictor f, and the algorithm leverages the differences between the predictions of f and h to improve learning performance.

In pseudo-labeling methods, the pseudo labels which are the outputs of h are modified from the outputs of f. There is a mapping function p used to obtain pseudo labels from the hypothesis space $\mathcal{F}$ to the hypothesis space $\mathcal{H}$ that $\forall f \in \mathcal{F}, h = p \circ f \in \mathcal{H}$. For example, in naive Pseudo-Label algorithm (Lee, 2013), p is a function that converts soft labels into hard labels. In Temporal Ensembling algorithm (Laine & Aila, 2017), p is an EMA function that ensembles historical predicted results.

In consistency regularization methods, h uses data augmentation which generates inconsistency based on f. There is an augment function a used to augment raw data that $\forall f \in \mathcal{F}, h = f \circ a \in \mathcal{H}$. For example, in VAT algorithm (Miyato et al., 2018), a is an adversarial augment function. In UDA algorithm (Xie et al., 2020), a is RandAugment for image classification.

In mixture methods that combine pseudo-labeling and consistency regularization, there both exist a mapping function p and an augment function a that $\forall f \in \mathcal{F}, h = p \circ f \circ a \in \mathcal{H}$. For example, in FixMatch method (Sohn et al., 2020), p is a function that converts soft labels into hard labels and a is RandAugment for image classification.

4.2. Shortcomings of SSL Algorithms

Previous SSL algorithms cannot effectively optimize the objective mentioned above, mainly for three reasons: coupling between the pseudo-label predictor and the target predictor, biased pseudo labels, and restricted sample weights.

Coupling between predictors. In classical SSL algorithms, the pseudo-label predictor h is usually obtained by modifying the target predictor f, and there is a severe coupling between h and f. When the distributions are consistent, the coupling is harmless. But when the distributions are inconsistent, f and h will have completely different optimization objectives. Without decoupling them, the learner needs to optimize two objectives $\frac{n_l}{n_l+n_u^w}\hat{E}(f, D_L) + \frac{n_u^w}{n_l+n_u^w}\hat{E}(f, \tilde{D}_U) + Disc(f, \mathcal{D}_T, Mix_{\frac{n_l}{n_l+n_u^w}}(\mathcal{D}_L, \mathcal{D}_U^w))$ and $\frac{n_u^w}{n_l+n_u^w}\hat{E}(p \circ f \circ a, D_L) + \frac{n_u^w}{n_l+n_u^w} Disc(p \circ f \circ a, \mathcal{D}_L, \mathcal{D}_U)$ jointly. It means there is a trade-off between them which results in only one of the Pareto frontier solutions of dual-objective optimization can be obtained. This coupling severely limits the performances on both objectives.

Biased pseudo labels. When the distributions of labeled and unlabeled data are consistent, a learner trained with labeled data can be directly used for predicting pseudo-labels. When the distributions are inconsistent, direct predictions will lead to severe performance degradation because of $Disc(h, \mathcal{D}_L, \mathcal{D}_U)$ which makes pseudo-labels biased.

Restricted sample weights. Sample selection or sample weighting help alleviate the discrepancy between the distribution of unlabeled data and the target distribution $Disc(f, \mathcal{D}_T, Mix_{\frac{n_l}{n_l+n_u^w}}(\mathcal{D}_L, \mathcal{D}_U^w))$. Assuming $p_L(x, y) = p_T(x, y)$, the weighting function w should make $\forall x \in \mathcal{X}, w(x)p_U(x) = p_T(x)$ ideally to relieve $Disc(f, \mathcal{D}_T, Mix_{\frac{n_l}{n_l+n_u^w}}(\mathcal{D}_L, \mathcal{D}_U))$ as much as possible. However, in previous SSL algorithms, the weighting function is usually an indicator function with a threshold that outputs 0 or 1. 0-1 weights don't have sufficient distribution adaptability and depend heavily on the choice of threshold.

5. Bidirectional Adaptation: A Framework for SSL with Inconsistent Data Distributions

To address the shortcomings of previous SSL methods, we propose a new SSL framework called Bidirectional Adaptation which adapts to the distribution of unlabeled data for debiased pseudo-label predictor and adapts to the target distribution for debiased target predictor. This framework can relieve both discrepancy terms $Disc(h, \mathcal{D}_L, \mathcal{D}_U)$ and $Disc(f, \mathcal{D}_T, Mix_{\frac{n_l}{n_l+n_u^w}}(\mathcal{D}_L, \mathcal{D}_U^w))$ without conflict.

Decoupling between predictors. This framework decouples the pseudo-label predictor and the target predictor by transforming a dual-objective optimization problem into two single-objective optimization problems. It avoids the trade-off between two objectives. Optimizing them respectively can yield a better solution than dual-objective optimization.

Theorem 5.1. *For the mapping function p and the augmentation function a, if $\forall f \in \mathcal{F}$, $h = p \circ f \circ a \in \mathcal{H}$, it can be proved that:*

$$\min_{\substack{f \in \mathcal{F} \\ h \in \mathcal{H}}} E(f, \mathcal{D}_T | h, D_L, D_U) \leq \min_{\substack{f \in \mathcal{F} \\ h = p \circ f \circ a}} E(f, \mathcal{D}_T | h, D_L, D_U) \tag{9}$$

Debiased pseudo labels. To relieve $Disc(h, \mathcal{D}_L, \mathcal{D}_U)$, We find that the optimization objective of the pseudo-label predictor is consistent with that of unsupervised DA. So we can adopt existing DA techniques (Glorot et al., 2011; Long et al., 2015; Zhu et al., 2020) to train the debiased pseudo-label predictor h. And then the pseudo-label predictor is used to predict debiased pseudo-labels adapting to the distribution of unlabeled data. In this step, soft pseudo-labels

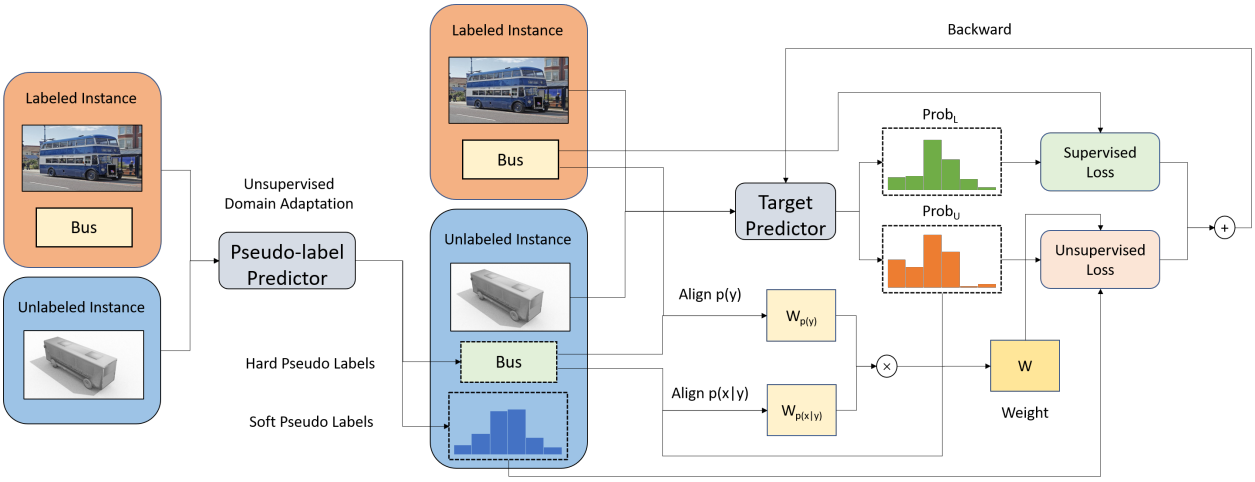

Figure 2. Illustration of the Bidirectional Adaptation framework.

sl and hard pseudo-labels hl can both be obtained. That is:

$$sl_i = h(y|x_i^u) \tag{10}$$

$$hl_i = \underset{j\in\{0,\dots,k-1\}}{\arg\max}\ h(y=j|x_i^u) \tag{11}$$

where $h(y|x_i^u) \in [0,1]^k$ is the predicted probabilities for unlabeled sample x_i^u produced by h. hl can be used for both intra-class and inter-class distribution alignments (Kang et al., 2019). sl which contains more information can be used as a teacher (Zhou & Jiang, 2004; Hinton et al., 2015) of the target predictor to compute the unsupervised loss.

Unrestricted sample weights. Unlike the pseudo-label predictor, the target predictor focuses on performing well on the distribution of labeled data. So the bias of the target predictor is difficult to be alleviated by previous DA techniques which focus on performing well on unlabeled data and don't care about the performance degradation on the distribution of labeled data. Weight-based methods which increase the weights of unlabeled samples in the distribution and decrease the weights of unlabeled samples out of the distribution are preferred for the target predictor as it less hurt the performance on labeled data. To relieve $Disc(f, \mathcal{D}_T, Mix_{\frac{n_l}{n_l+n_u^w}}(\mathcal{D}_L, \mathcal{D}_U^w))$, our method removes the restriction on weights imposed by the indicator function in previous SSL algorithms and adopts a more reasonable weighting function for distribution alignment adapting to the target distribution. After obtaining pseudo-labels, the previously unobserved $p_U(y)$ and $p_U(x|y)$ can now be estimated. According to Bayes formula, $p(x) = \frac{p(y)p(x|y)}{p(y|x)}$, for constant $p(y|x)$, $p(x)$ can be aligned by aligning $p(x|y)$ using intra-class weights and aligning $p(y)$ using inter-class weights respectively.

Aligning $p(x|y)$ with intra-class weights. Increasing the weights of unlabeled samples consistent to the distribution of labeled data and vice versa can effectively reduce the distribution discrepancy. Neural networks' confidence can measure the consistency effectively (Sohn et al., 2020; Chen et al., 2020). However, the degrees of confidence for different classes are usually different because some classes are relatively difficult to learn (Zhang et al., 2021), so confidence-based distribution alignment is only fair within each class. Therefore, we use the ratio of the confidence of a sample and the average confidence within its class to measure how consistent it matches the distribution of labeled data. Considering that in the case with a large number of classes, the effect of batch-wise alignment will be limited because the number of samples in the same class in each batch is small. However, if the probabilities of the historical batches are stored for alignment, not only the storage resource consumption is large, but also the computational complexity is high. So we adopt an on-the-fly method for maintaining the average of the confidence within each class $avgcon$. In addition, considering that the confidence will gradually increase with the training of the model (Xu et al., 2021), in order to avoid the intra-class weights from increasing, it is necessary to normalize them in each batch.

$$w_{p(x|y)}(x_i^u) = \frac{\mu B \times \frac{Prob_{u_i}[hl_{B_i}]}{avgcon[hl_{B_i}]}}{\sum_{j=1}^{\mu B} \frac{Prob_{u_j}[hl_{B_j}]}{avgcon[hl_{B_i}]}} \tag{12}$$

Aligning $p(y)$ with inter-class weights. For aligning $p(y)$, we determine the weight of each class according to its proportion in labeled data and its proportion in unlabeled data based on hard pseudo-labels hl. For each class c, we can count the number of times it appears in labels $n_l^c = \sum_{i=1}^{n_l} \mathbb{I}(c == y_i)$ and pseudo-labels $n_u^c = \sum_{i=1}^{n_u} \mathbb{I}(c == hl_i)$

Algorithm 1 Bidirectional Adaptation.

Input: labeled dataset $D_L = \{(x_1^l, y_1), \ldots, (x_{n_l}^l, y_{n_l})\}$, unlabeled dataset $D_U = \{x_1^u, \ldots, x_{n_u}^u\}$, DA algorithm A, the total number of iterations T, the learning rate η, the batch size B, the number of classes k, the ratio of the number of labeled samples to the number of unlabeled samples μ in each batch, the ratio of unsupervised loss to supervised loss λ_u.
Output: pseudo-label predictor h, target predictor f.
Perform the DA algorithm: $h = A(D_L, D_U)$
for $i = 1$ **to** n_u **do**
  Compute the soft pseudo-label sl_i by Equation (10)
  Compute the hard pseudo-label hl_i by Equation (11)
end for
Initialize f with parameters θ_0
for $c = 0$ **to** $k-1$ **do**
  $avgcon[c] = 0$, $cnt[c] = 0$
end for
$avgcon_{all} = 0$, $cnt_{all} = 0$
for $t = 0$ **to** $T-1$ **do**
  $\{x_B^l, y_B\} \leftarrow Sample(D_L)$
  $\{x_B^u, hl_B, sl_B\} \leftarrow Sample(\{D_U, hl, sl\})$
  $Prob_l = f(y|x_B^l; \theta_t)$, $Prob_u = f(y|x_B^u; \theta_t)$
  for $i = 0$ **to** $\mu B - 1$ **do**
    Compute the weight of $x_{B_i}^u$ by Equations (12) to (14)
    $avgcon[hl_{B_i}] = \frac{avgcon[hl_{B_i}] \times cnt[hl_{B_i}] + prob_{u_i}[hl_{B_i}]}{cnt[hl_{B_i}]+1}$
    $cnt[hl_{B_i}] = cnt[hl_{B_i}] + 1$
  end for
  Compute the loss $\mathcal{L}$ by Equations (15) to (17)
  $\theta_{t+1} = \theta_t - \eta \nabla_\theta \mathcal{L}$
end for

respectively in advance.

$$w_{p(y)}(x_i^u) = \frac{n_u \times n_l^{hl_i}}{n_l \times n_u^{hl_i}} \tag{13}$$

The product of the intra-class weights $w_{p(x|y)}(x_i^u)$ and the inter-class weights $w_{p(y)}(x_i^u)$ can be used as the final weights of unlabeled data for distribution alignment.

$$w(x_i^u) = w_{p(x|y)}(x_i^u) \times w_{p(y)}(x_i^u) \tag{14}$$

Loss function for training the target predictor. For each batch, the supervised loss $\mathcal{L}_S$ can be defined as the cross-entropy between the true label y_i and the probability $Prob_{l_i} = f(y|x_i^l; \theta)$ where $f(y|x_i^l; \theta) \in [0,1]^k$ is the predicted probability produced by f with current parameters θ for the labeled sample x_i^l, and the unsupervised loss $\mathcal{L}_U$ can be defined as the weighted cross-entropy between the soft pseudo-label sl_{B_i} and the probability $Prob_{u_i} = f(y|x_i^u; \theta)$ for the unlabeled sample x_i^u. The total loss is the combination of $\mathcal{L}_S$ and $\mathcal{L}_U$ with a hyper-parameter λ_u. We denote $CE(\cdot,\cdot)$ the cross-entropy function.

$$\mathcal{L}_S = \frac{1}{B} \sum_{i=1}^{B} CE(Prob_{l_i}, y_i) \tag{15}$$

$$\mathcal{L}_U = \frac{1}{\mu B} \sum_{i=1}^{\mu B} w(x_i^u) \times CE(Prob_{u_i}, sl_{B_i}) \tag{16}$$

$$\mathcal{L} = \mathcal{L}_S + \lambda_u \times \mathcal{L}_U \tag{17}$$

The overall algorithm is summarized in Algorithm 1 and the main computation flowchart is illustrated in Figure 2.

6. Experiments

6.1. Experiment for Theoretical Arguments

We conduct an experiment on the extracted features of Office-Caltech (Gong et al., 2012) dataset to prove that both of the distribution discrepancies we explain in our theoretical framework actually exist, and only by alleviating both can the performance be the best. Office-Caltech contains images from four domains: Amazon, Caltech, Webcam, and Dslr. We choose the 4096-dimensional feature vectors extracted by the DECAF convolutional neural network (Donahue et al., 2014), pre-trained on ImageNet (Russakovsky et al., 2015). These vectors are commonly used to evaluate statistical machine learning algorithms when dealing with inconsistent distributions. We have 12 settings of the combination of labeled data domain and unlabeled data domain and 4 settings of the number of labels $\{20, 30, 40, 50\}$. For reliability, 5 replicates of each experiment with random seeds $0 \sim 4$ are performed. We use the average accuracy of all combinations and replicates for each label setting to compare the performance of different methods.

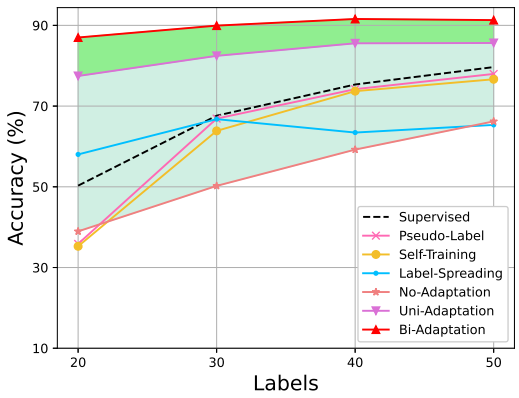

Figure 3. Accuracy of compared methods on the Office-Caltech dataset with various numbers of labels.

We demonstrate the existence of both distribution discrepancies by comparing 3 methods: No-Adaptation which uses

Table 1. Experiments on Image-CLEF with 150 labels and 300 labels.

Methods	150 labels					
	C/I	C/P	I/C	I/P	P/C	P/I
Supervised	96.09 ± 0.57	96.09 ± 0.57	91.29 ± 1.10	91.29 ± 1.10	73.69 ± 1.52	73.69 ± 1.52
Mean Teacher	96.00 ± 0.79	95.91 ± 0.68	90.67 ± 0.82	91.16 ± 0.88	74.36 ± 2.54	74.44 ± 1.84
FixMatch	95.56 ± 1.09	95.11 ± 0.92	89.02 ± 2.15	91.02 ± 1.48	73.18 ± 1.64	$\mathbf{76.22 \pm 1.01}$
FlexMatch	95.38 ± 0.84	94.62 ± 0.80	89.47 ± 1.35	91.33 ± 1.02	72.98 ± 1.02	74.80 ± 1.18
UASD	95.16 ± 0.91	95.33 ± 0.72	89.38 ± 1.02	88.31 ± 0.75	71.91 ± 1.41	70.13 ± 0.88
CAFA	96.09 ± 0.68	95.87 ± 0.84	90.89 ± 0.74	$\mathbf{91.38 \pm 0.69}$	74.09 ± 2.41	74.09 ± 2.00
Ours	$\mathbf{96.67 \pm 0.82}$	$\mathbf{96.36 \pm 0.80}$	$\mathbf{91.33 \pm 1.10}$	91.16 ± 0.78	$\mathbf{75.02 \pm 1.51}$	76.13 ± 1.25
Methods	300 labels					
	C/I	C/P	I/C	I/P	P/C	P/I
Supervised	97.13 ± 0.62	97.13 ± 0.62	92.80 ± 0.81	92.80 ± 0.81	76.47 ± 1.71	76.47 ± 1.71
MeanTeacher	96.67 ± 1.13	96.53 ± 1.07	93.13 ± 1.74	$\mathbf{93.27 \pm 2.02}$	74.13 ± 1.10	73.40 ± 1.77
FixMatch	96.20 ± 1.56	96.67 ± 1.60	92.40 ± 0.64	93.13 ± 1.02	76.20 ± 1.56	77.00 ± 2.07
FlexMatch	96.60 ± 1.10	96.00 ± 0.97	91.73 ± 0.44	92.20 ± 0.69	75.00 ± 2.30	76.60 ± 1.16
UASD	96.67 ± 0.67	96.73 ± 0.49	92.80 ± 0.80	92.07 ± 0.72	73.67 ± 1.51	71.87 ± 1.52
CAFA	96.93 ± 0.92	96.80 ± 0.51	92.87 ± 0.61	93.07 ± 0.72	75.13 ± 2.70	74.67 ± 1.49
Ours	$\mathbf{97.33 \pm 0.62}$	$\mathbf{97.27 \pm 0.43}$	$\mathbf{93.67 \pm 2.07}$	93.20 ± 1.99	$\mathbf{77.20 \pm 2.03}$	$\mathbf{77.07 \pm 2.21}$

two naive SVMs as the pseudo-label predictor and the target predictor respectively, Uni-Adaptation which uses a JDOT-SVM (Courty et al., 2017) which uses optimal transport to relieve distribution discrepancies as the pseudo-label predictor, and uses a naive SVM as the target predictor, Bi-Adaptation which uses two JDOT-SVMs as the pseudo-label predictor and the target predictor respectively. Supervised learning baseline and 3 SSL algorithms Pseudo-Label, Self-Training (Yarowsky, 1995), and Label-Spreading (Zhou et al., 2003) are used for comparison too. This experiment results (as shown in Figure 3) demonstrate the existence of two distribution discrepancies and alleviating both of them can achieve better performance than not alleviating or alleviating only one of them. The area between the curve of Uni-Adaptation and the curve of No-Adaptation can be seen as a reflection of the distribution discrepancy $Disc(h, \mathcal{D}_L, \mathcal{D}_U)$. The area between the curve of Bi-Adaptation and the curve of Uni-Adaptation can be seen as a reflection of the distribution discrepancy $Disc(f, \mathcal{D}_T, Mix_{\frac{n_l}{n_l+n_u^w}}(\mathcal{D}_L, \mathcal{D}_U^w))$.

6.2. Experiments on Performance Robustness

We conduct extensive experiments on image datasets to demonstrate the effectiveness of our proposed method. We selected three commonly used datasets with multiple domains: Image-CLEF (Caputo et al., 2014), Office-31 (Saenko et al., 2010), and VisDA-2017 (Peng et al., 2018). Image-CLEF contains 600 images from three domains: Caltech (C), ImageNet (I), and Pascal (P). Each domain consists of 12 categories, and each category contains 50 images. Office-31 contains 4,110 images from three domains: 2,817 images from Amazon (A), 498 images from Dslr (D), and 795 images from Webcam (W). Each domain consists of 31 categories. VisDA-2017 is a challenging dataset due to the significant domain drift between 152,397 synthetic images (S) and 55,388 real images (R) from 12 categories. Only $p(x|y)$ is inconsistent in Image-CLEF, whereas both $p(y)$ and $p(x|y)$ are inconsistent in Office-31 and VisDA-2017.

For Image-CLEF and Office-31, we have 6 settings of the combination of labeled data domain and unlabeled data domain and 2 settings of the number of labeled samples $\{150, 300\}$. For VisDA-2017, we have 2 settings of the combination of labeled data domain and unlabeled data domain and 3 settings of the number of labeled samples $\{150, 300, 600\}$. For reliability, 5 replicates of each experiment with random seeds $0 \sim 4$ are performed. The average and standard deviation of the accuracy are reported.

Our method is compared with supervised learning, classical deep SSL methods Mean Teacher (Tarvainen & Valpola, 2017), FixMatch (Sohn et al., 2020) and FlexMatch (Zhang et al., 2021), and robust deep SSL methods UASD (Chen et al., 2020) and CAFA (Huang et al., 2021).

We adopt DSAN (Zhu et al., 2020) as the basic unsupervised DA method in all experiments. The batch size B is set to 64, the max iteration T is set to 2000, the ratio of unlabeled to labeled data μ is set to 1.0, and the ratio of unsupervised to supervised loss λ_u is set to 0.1. For fairness, all methods use ResNet-50 (He et al., 2016) pre-trained on ImageNet (Russakovsky et al., 2015) as the backbone network and SGD with the initial learning rate 5×10^{-4} as the optimizer. The algorithm implementations are based on the SSL toolkit LAMDA-SSL (Jia et al., 2023). This work uses the Huawei

Table 2. Experiments on Office-31 with 150 labels and 300 labels.

Methods	150 labels					
	A/D	A/W	D/A	D/W	W/A	W/D
Supervised	72.45 ± 0.71	72.45 ± 0.71	95.11 ± 0.98	95.11 ± 0.98	$\mathbf{93.27 \pm 0.84}$	93.27 ± 0.84
Mean Teacher	72.31 ± 1.26	72.01 ± 1.00	94.37 ± 1.45	94.02 ± 1.34	90.98 ± 1.56	91.26 ± 1.73
FixMatch	70.97 ± 1.10	72.75 ± 0.49	93.85 ± 1.39	94.48 ± 1.69	91.13 ± 1.28	91.72 ± 1.09
FlexMatch	66.40 ± 3.20	67.40 ± 2.07	87.07 ± 3.08	86.67 ± 2.27	82.54 ± 2.73	87.72 ± 1.76
UASD	71.39 ± 0.96	71.38 ± 0.91	94.83 ± 0.93	93.33 ± 1.06	92.25 ± 1.46	91.10 ± 1.29
CAFA	72.17 ± 1.30	72.08 ± 1.22	94.31 ± 1.38	94.60 ± 1.14	90.98 ± 1.72	91.19 ± 1.60
Ours	$\mathbf{73.54 \pm 1.36}$	$\mathbf{73.70 \pm 1.26}$	$\mathbf{95.29 \pm 0.84}$	$\mathbf{96.26 \pm 1.04}$	93.05 ± 2.13	$\mathbf{94.42 \pm 0.41}$
Methods	300 labels					
	A/D	A/W	D/A	D/W	W/A	W/D
Supervised	78.17 ± 0.72	78.17 ± 0.72	$\mathbf{97.98 \pm 0.71}$	97.98 ± 0.71	97.54 ± 0.80	97.54 ± 0.80
Mean Teacher	77.93 ± 0.72	77.86 ± 0.98	96.06 ± 0.83	96.26 ± 0.58	95.76 ± 0.86	95.56 ± 0.82
FixMatch	76.85 ± 1.18	78.35 ± 1.03	96.57 ± 1.53	96.26 ± 1.54	95.76 ± 1.36	95.27 ± 1.26
FlexMatch	71.39 ± 5.67	72.79 ± 2.75	95.76 ± 1.09	92.52 ± 2.56	92.57 ± 2.03	92.28 ± 1.12
UASD	78.14 ± 0.94	78.35 ± 0.97	96.67 ± 0.68	97.17 ± 1.11	96.97 ± 0.82	96.57 ± 0.97
CAFA	77.51 ± 1.09	77.46 ± 0.84	97.17 ± 0.56	97.68 ± 0.85	96.53 ± 0.83	96.28 ± 1.03
Ours	$\mathbf{78.56 \pm 0.81}$	$\mathbf{78.74 \pm 0.82}$	97.68 ± 0.99	$\mathbf{98.38 \pm 0.74}$	$\mathbf{97.98 \pm 0.36}$	$\mathbf{98.02 \pm 0.66}$

Table 3. Experiments on VisDA-2017 with 150 labels, 300 labels and 600 labels.

Methods	150 labels		300 labels		600 labels	
	S/R	R/S	S/R	R/S	S/R	R/S
Supervised	85.33 ± 1.54	78.50 ± 0.68	89.64 ± 0.73	81.81 ± 0.62	92.20 ± 0.45	84.13 ± 0.36
Mean Teacher	84.15 ± 1.08	73.68 ± 1.00	86.90 ± 0.61	76.90 ± 0.46	89.05 ± 0.48	79.86 ± 0.30
FixMatch	78.46 ± 4.15	67.10 ± 9.46	82.88 ± 0.85	71.74 ± 0.45	87.68 ± 1.15	79.54 ± 1.88
FlexMatch	83.43 ± 1.74	67.90 ± 1.77	88.09 ± 0.53	75.17 ± 1.34	90.11 ± 1.09	79.28 ± 0.38
UASD	85.58 ± 1.55	78.59 ± 0.41	89.58 ± 0.79	81.82 ± 0.68	92.29 ± 0.45	84.04 ± 0.31
CAFA	83.95 ± 1.79	72.89 ± 1.03	87.81 ± 0.47	76.48 ± 0.72	89.84 ± 0.62	78.63 ± 0.44
Ours	$\mathbf{85.92 \pm 1.16}$	$\mathbf{79.15 \pm 0.39}$	$\mathbf{89.85 \pm 0.71}$	$\mathbf{82.27 \pm 0.60}$	$\mathbf{92.46 \pm 0.38}$	$\mathbf{84.28 \pm 0.36}$

MindSpore platform for experimental testing partially.

Experiments in Tables 1 to 3 show the effectiveness of our method. As the distributions are inconsistent, classical SSL methods can not perform well because of the shortcomings analyzed in Section 4.2. Robust SSL methods which only focus on label distributions such as UASD can not perform well with inconsistent intra-class feature distributions. Robust SSL methods which focus on both distributions such as CAFA can't alleviate two discrepancies jointly. Unlike other SSL algorithms that perform worse than the baseline, our proposed method achieves more robust performance.

7. Conclusion

In this paper, we focus on the problem of inconsistent distributions in SSL algorithms, which is of great significance for expanding their application scope. We provide ample theoretical results for this problem, indicating that there are two terms of distribution discrepancy that need to be alleviated in the generalization error. Based on theoretical results, we obtain the optimization objective of SSL with inconsistent distributions. By formalizing previous SSL algorithms, we find three main shortcomings that cause poor performance on this problem: coupling between the pseudo-label predictor and the target predictor, biased pseudo-labels, and restricted sample weights. To overcome these shortcomings, we propose a practical framework called Bidirectional Adaptation to alleviate both terms of discrepancy for more robust performance against inconsistent distributions. A vast number of experiments confirm the correctness of our theoretical framework and the efficacy of our practical framework.

Acknowledgements

This research was supported by the National Key R&D Program of China (2022YFC3340901), the National Science Foundation of China (62176118) and CAAI-Huawei MindSpore Open Fund.

References

Balcan, M.-F. and Blum, A. A discriminative model for semi-supervised learning. *Journal of the ACM*, 57(3): 1–46, 2010.

Ben-David, S., Cesa-Bianchi, N., and Long, P. M. Characterizations of learnability for classes of {O,. . . , n}-valued functions. In *Proceedings of the 5th Annual Conference on Computational Learning Theory*, pp. 333–340, 1992.

Ben-David, S., Lu, T., and Pál, D. Does unlabeled data provably help? worst-case analysis of the sample complexity of semi-supervised learning. In *Proceedings of the 21st Annual Conference on Computational Learning Theory*, pp. 33–44, 2008.

Cao, K., Wei, C., Gaidon, A., Arechiga, N., and Ma, T. Learning imbalanced datasets with label-distribution-aware margin loss. In *Advances in Neural Information Processing Systems*, pp. 1565–1576, 2019.

Caputo, B., Müller, H., Martinez-Gomez, J., Villegas, M., Acar, B., Patricia, N., Marvasti, N., Üsküdarlı, S., Paredes, R., and Cazorla, M. Imageclef 2014: Overview and analysis of the results. In *Proceedings of 5th International Conference of the Cross-Language Evaluation Forum for European Languages*, pp. 192–211, 2014.

Chapelle, O., Scholkopf, B., and Zien, A. *Semi-Supervised Learning*. The MIT Press, 2006.

Chen, Y., Zhu, X., Li, W., and Gong, S. Semi-supervised learning under class distribution mismatch. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence*, pp. 3569–3576, 2020.

Courty, N., Flamary, R., Habrard, A., and Rakotomamonjy, A. Joint distribution optimal transportation for domain adaptation. In *Advances in Neural Information Processing Systems*, pp. 3730–3739, 2017.

Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., and Darrell, T. Decaf: A deep convolutional activation feature for generic visual recognition. In *Proceedings of the 31st International Conference on Machine Learning*, pp. 647–655, 2014.

Farahani, A., Voghoei, S., Rasheed, K., and Arabnia, H. R. A brief review of domain adaptation. In *Advances in Data Science and Information Engineering*, pp. 877–894, 2021.

Ganin, Y. and Lempitsky, V. Unsupervised domain adaptation by backpropagation. In *Proceedings of the 32nd International Conference on Machine Learning*, pp. 1180–1189, 2015.

Glorot, X., Bordes, A., and Bengio, Y. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th International Conference on Machine Learning*, pp. 513–520, 2011.

Gong, B., Shi, Y., Sha, F., and Grauman, K. Geodesic flow kernel for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2066–2073, 2012.

Gopalan, R., Li, R., and Chellappa, R. Domain adaptation for object recognition: An unsupervised approach. In *Proceedings of IEEE International Conference on Computer Vision*, pp. 999–1006, 2011.

Gretton, A., Borgwardt, K., Rasch, M., Schölkopf, B., and Smola, A. A kernel method for the two-sample-problem. In *Advances in Neural Information Processing Systems*, pp. 513–520, 2006.

Guo, L.-Z. and Li, Y.-F. Class-imbalanced semi-supervised learning with adaptive thresholding. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 8082–8094, 2022.

Guo, L.-Z., Zhang, Z.-Y., Jiang, Y., Li, Y.-F., and Zhou, Z.-H. Safe deep semi-supervised learning for unseen-class unlabeled data. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 3897–3906, 2020.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Hinton, G., Vinyals, O., Dean, J., et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Huang, J., Gretton, A., Borgwardt, K., Schölkopf, B., and Smola, A. Correcting sample selection bias by unlabeled data. In *Advances in Neural Information Processing Systems*, pp. 601–608, 2006.

Huang, Z., Xue, C., Han, B., Yang, J., and Gong, C. Universal semi-supervised learning. In *Advances in Neural Information Processing Systems*, pp. 26714–26725, 2021.

Jeong, J., Lee, S., Kim, J., and Kwak, N. Consistency-based semi-supervised learning for object detection. In *Advances in Neural Information Processing Systems*, pp. 10758–10767, 2019.

Jia, L.-H., Guo, L.-Z., Zhou, Z., and Li, Y.-F. Lamda-ssl: Semi-supervised learning in python. *Science China Information Sciences*, 2023.

Kang, G., Jiang, L., Yang, Y., and Hauptmann, A. G. Contrastive adaptation network for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4893–4902, 2019.

Laine, S. and Aila, T. Temporal ensembling for semi-supervised learning. In *Proceedings of the 5th International Conference on Learning Representations*, 2017.

Lee, D.-H. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Proceedings of the 30th International Conference on Machine Learning Workshop on Challenges in Representation Learning*, pp. 896, 2013.

Leskes, B. The value of agreement, a new boosting algorithm. In *Proceedings of the 18th Annual Conference on Computational Learning Theory*, pp. 95–110, 2005.

Li, Y.-F. and Zhou, Z.-H. Towards making unlabeled data never hurt. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(1):175–188, 2014.

Li, Y.-F., Guo, L.-Z., and Zhou, Z.-H. Towards safe weakly supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(1):334–346, 2021.

Long, M., Cao, Y., Wang, J., and Jordan, M. Learning transferable features with deep adaptation networks. In *Proceedings of the 32nd International Conference on Machine Learning*, pp. 97–105, 2015.

Miyato, T., Dai, A. M., and Goodfellow, I. J. Adversarial training methods for semi-supervised text classification. In *Proceedings of the 5th International Conference on Learning Representations*, 2017.

Miyato, T., Maeda, S.-i., Koyama, M., and Ishii, S. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41 (8):1979–1993, 2018.

Natarajan, B. K. On learning sets and functions. *Machine Learning*, 4(1):67–97, 1989.

Oliver, A., Odena, A., Raffel, C. A., Cubuk, E. D., and Goodfellow, I. Realistic evaluation of deep semi-supervised learning algorithms. In *Advances in Neural Information Processing Systems*, pp. 3239–3250, 2018.

Peng, X., Usman, B., Kaushik, N., Wang, D., Hoffman, J., and Saenko, K. Visda: A synthetic-to-real benchmark for visual domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 2021–2026, 2018.

Peters, J., Janzing, D., and Schölkopf, B. *Elements of causal inference: foundations and learning algorithms*. The MIT Press, 2017.

Quinonero-Candela, J., Sugiyama, M., Schwaighofer, A., and Lawrence, N. D. *Dataset shift in machine learning*. The MIT Press, 2008.

Rigollet, P. Generalization error bounds in semi-supervised classification under the cluster assumption. *Journal of Machine Learning Research*, 8(7):1369–1392, 2007.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3): 211–252, 2015.

Saenko, K., Kulis, B., Fritz, M., and Darrell, T. Adapting visual category models to new domains. In *Proceedings of the 11th European Conference on Computer Vision*, pp. 213–226, 2010.

Sanz-Alonso, D. and Yang, R. Unlabeled data help in graph-based semi-supervised learning: a bayesian nonparametrics perspective. *Journal of Machine Learning Research*, 23(97):1–28, 2022.

Seeger, M. Input-dependent regularization of conditional density models. Technical report, 2000.

Shao, J.-J., Guo, L.-Z., Yang, X.-W., and Li, Y.-F. Log: Active model adaptation for label-efficient ood generalization. In *Advances in Neural Information Processing Systems*, pp. 11023–11034, 2022a.

Shao, J.-J., Xu, Y., Cheng, Z., and Li, Y.-F. Active model adaptation under unknown shift. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1558–1566, 2022b.

Singh, A., Nowak, R., and Zhu, J. Unlabeled data: Now it helps, now it doesn't. In *Advances in Neural Information Processing Systems*, pp. 1513–1520, 2008.

Sinha, K. and Belkin, M. The value of labeled and unlabeled examples when the model is imperfect. In *Advances in Neural Information Processing Systems*, pp. 1361–1368, 2007.

Sohn, K., Berthelot, D., Carlini, N., Zhang, Z., Zhang, H., Raffel, C. A., Cubuk, E. D., Kurakin, A., and Li, C.-L. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *Advances in Neural Information Processing Systems*, pp. 596–608, 2020.

Souly, N., Spampinato, C., and Shah, M. Semi supervised semantic segmentation using generative adversarial network. In *Proceedings of IEEE International Conference on Computer Vision*, pp. 5688–5696, 2017.

Sugiyama, M., Krauledat, M., and Müller, K.-R. Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, 8(5):985–1005, 2007.

Tarvainen, A. and Valpola, H. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems*, pp. 1195–1204, 2017.

Vapnik, V. and Chervonenkis, A. Y. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability Its Applications*, 16(2): 264–280, 1971.

Wang, F., Wang, Q., Li, W., Xu, D., and Van Gool, L. Revisiting deep semi-supervised learning: An empirical distribution alignment framework and its generalization bound. *arXiv preprint arXiv:2203.06639*, 2022.

Wasserman, L. and Lafferty, J. Statistical analysis of semi-supervised regression. In *Advances in Neural Information Processing Systems*, pp. 801–808, 2007.

Wei, C., Shen, K., Chen, Y., and Ma, T. Theoretical analysis of self-training with deep networks on unlabeled data. In *Proceedings of the 9th International Conference on Learning Representations*, 2020.

Xie, Q., Dai, Z., Hovy, E., Luong, T., and Le, Q. Unsupervised data augmentation for consistency training. In *Advances in Neural Information Processing Systems*, pp. 6256–6268, 2020.

Xu, Y., Shang, L., Ye, J., Qian, Q., Li, Y.-F., Sun, B., Li, H., and Jin, R. Dash: Semi-supervised learning with dynamic thresholding. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 11525–11536, 2021.

Yarowsky, D. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pp. 189–196, 1995.

Yu, Q., Ikami, D., Irie, G., and Aizawa, K. Multi-task curriculum framework for open-set semi-supervised learning. In *Proceedings of the 16th European Conference on Computer Vision*, pp. 438–454, 2020.

Zhang, B., Wang, Y., Hou, W., Wu, H., Wang, J., Okumura, M., and Shinozaki, T. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. In *Advances in Neural Information Processing Systems*, pp. 18408–18419, 2021.

Zhou, D., Bousquet, O., Lal, T., Weston, J., and Schölkopf, B. Learning with local and global consistency. In *Advances in Neural Information Processing Systems*, pp. 321–328, 2003.

Zhou, Z.-H. and Jiang, Y. Nec4. 5: neural ensemble based c4. 5. *IEEE Transactions on Knowledge and Data Engineering*, 16(6):770–773, 2004.

Zhu, Y., Zhuang, F., Wang, J., Ke, G., Chen, J., Bian, J., Xiong, H., and He, Q. Deep subdomain adaptation network for image classification. *IEEE Transactions on Neural Networks and Learning Systems*, 32(4):1713–1722, 2020.

A. Theorem Proof

A.1. Proof of Theorem 3.1

When all samples are from the same distribution, $\mathcal{D}_L = \mathcal{D}_U$, in the case of using only n_l labeled samples for supervised learning, for any $h \in \mathcal{H}$ and $0 \leq \delta_1 \leq 1$, with the probability of at least $1 - \delta_1$:

$$E(h, \mathcal{D}_L) \leq \hat{E}(h, D_L) + var(\mathcal{H}, n_l, k, \delta_1) \tag{18}$$

where $\hat{E}(h, D_L)$ is the empirical error of h on the dataset D_L and $E(f, \mathcal{D}_L)$ is the generalization error of f on the distribution $\mathcal{D}_L$.

When dataset D_U with n_u samples from the same distribution, for any $h \in \mathcal{H}$ and $0 \leq \delta_2 \leq 1$, with the probability of at least $1 - \delta_2$:

$$\begin{aligned}\hat{E}(h, D_U) &\leq E(h, \mathcal{D}_U) + var(\mathcal{H}, n_u, k, \delta_2) \\ &= E(h, \mathcal{D}_L) + var(\mathcal{H}, n_u, k, \delta_2)\end{aligned} \tag{19}$$

According to Equations (18) and (19), for any pseudo-label predictor $h \in \mathcal{H}$, $0 \leq \delta_1 \leq 1$ and $0 \leq \delta_2 \leq 1$, with the probability of at least $(1 - \delta_1)(1 - \delta_2)$:

$$\hat{E}(h, D_U) \leq \hat{E}(h, D_L) + var(\mathcal{H}, n_l, k, \delta_1) + var(\mathcal{H}, n_u, k, \delta_2) \tag{20}$$

A.2. Proof of Theorem 3.2

In SSL, the target predictor is trained with both labeled dataset D_L and unlabeled dataset with noisy pseudo-labels $\tilde{D}_U$ whose noisy rate is $\hat{E}(h, D_U)$. Two datasets can be considered as a mixed one with $n_l + n_u$ samples from the same distribution $Mix_{\frac{n_l}{n_l+n_u}}(\mathcal{D}_L, \mathcal{D}_U)$ whose noisy rate is $\frac{n_u}{n_l+n_u}\hat{E}(h, D_U)$. When all samples are from the same distribution, $Mix_{\frac{n_l}{n_l+n_u}}(\mathcal{D}_L, \mathcal{D}_U) = \mathcal{D}_L = \mathcal{D}_U = \mathcal{D}_T$.

So, for any target predictor $f \in \mathcal{F}$, pseudo-label predictor $h \in \mathcal{H}$, $0 \leq \delta_3 \leq 1$, with the probability of at least $1 - \delta_3$:

$$\begin{aligned}&E(f, \mathcal{D}_T | h, D_L, D_U) \\ =&E(f, Mix_{\frac{n_l}{n_l+n_u}}(\mathcal{D}_L, \mathcal{D}_U) | h, D_L, D_U) \\ \leq&\frac{n_l}{n_l + n_u}\hat{E}(f, D_L) + \frac{n_u}{n_l + n_u}\hat{E}(f, \tilde{D}_U) + var(\mathcal{F}, n_l + n_u, k, \delta_3) + \frac{n_u}{n_l + n_u}\hat{E}(h, D_U)\end{aligned} \tag{21}$$

where $E(f, \mathcal{D}_T | h, D_L, D_U)$ is the generalization error of f on the distribution $\mathcal{D}_T$ corresponding to pseudo-label predictor h, $\hat{E}(f, D_L)$ is the empirical error of the target predictor f on the dataset D_L and $\hat{E}(f, \tilde{D}_U)$ is the inconsistency rate between the noisy pseudo-labels and the prediction results of f on the unlabeled dataset $\tilde{D}_U$.

According to Equations (20) and (21), for any target predictor $f \in \mathcal{F}$, pseudo-label predictor $h \in \mathcal{H}$, $0 \leq \delta_1 \leq 1$, $0 \leq \delta_2 \leq 1$, $0 \leq \delta_3 \leq 1$, with the probability of at least $(1 - \delta_1)(1 - \delta_2)(1 - \delta_3)$:

$$\begin{aligned}&E(f, \mathcal{D}_T | h, D_L, D_U) \\ =&E(f, \mathcal{D}_L | h, D_L, D_U) \\ \leq&\frac{n_l}{n_l + n_u}\hat{E}(f, D_L) + \frac{n_u}{n_l + n_u}\hat{E}(f, \tilde{D}_U) + var(\mathcal{F}, n_l + n_u, k, \delta_1) \\ &+ \frac{n_u}{n_l + n_u}(\hat{E}(h, D_L) + var(\mathcal{H}, n_l, k, \delta_2) + var(\mathcal{H}, n_u, k, \delta_3))\end{aligned} \tag{22}$$

A.3. Proof of Theorem 3.3

When labeled data and unlabeled data are from different distributions, for any $h \in \mathcal{H}$:

$$\begin{aligned}&E(h, \mathcal{D}_U) \\ \leq&E(h, \mathcal{D}_L) + |p_{x,y\sim\mathcal{D}_L}(h(x) \neq y) - p_{x,y\sim\mathcal{D}_U}(h(x) \neq y)| \\ =&E(h, \mathcal{D}_L) + Disc(h, \mathcal{D}_L, \mathcal{D}_U)\end{aligned} \tag{23}$$

According to Equations (18), (19) and (23), for any pseudo-label predictor $h \in \mathcal{H}$, $0 \leq \delta_1 \leq 1$ and $0 \leq \delta_2 \leq 1$, with the probability of at least $(1-\delta_1)(1-\delta_2)$:

$$\begin{aligned}
&\hat{E}(h, D_U) \\
\leq &E(h, \mathcal{D}_L) + Disc(h, \mathcal{D}_L, \mathcal{D}_U) \\
\leq &\hat{E}(h, D_L) + var(\mathcal{H}, n_l, k, \delta_1) + var(\mathcal{H}, n_u, k, \delta_2) + Disc(h, \mathcal{D}_L, \mathcal{D}_U)
\end{aligned} \tag{24}$$

A.4. Proof of Theorem 3.4

In SSL with inconsistent distributions, the target predictor is trained with both labeled dataset D_L and weighted unlabeled dataset with noisy pseudo-labels $\tilde{D}_U^w$. Assuming that the probabilities of the pseudo-label predictor making wrong predictions for each sample are equal without considering the difference between samples, the noisy rate of $\tilde{D}_U^w$ is $\hat{E}(h, D_U)$. D_L and $\tilde{D}_U^w$ can be considered as a mixed dataset with $n_l + n_u^w$ samples from the mixed distribution $Mix_{\frac{n_l}{n_l+n_u^w}}(\mathcal{D}_L, \mathcal{D}_U^w)$ whose noisy rate is $\frac{n_u^w}{n_l+n_u^w}\hat{E}(h, D_U^w)$.

So, for any target predictor $f \in \mathcal{F}$, pseudo-label predictor $h \in \mathcal{H}$, $0 \leq \delta_3 \leq 1$, with the probability of at least $1-\delta_3$:

$$\begin{aligned}
&E(f, Mix_{\frac{n_l}{n_l+n_u^w}}(\mathcal{D}_L, \mathcal{D}_U^w)|h, D_L, D_U) \\
\leq &\frac{n_l}{n_l+n_u^w}\hat{E}(f, D_L) + \frac{n_u^w}{n_l+n_u^w}\hat{E}(f, \tilde{D}_U) + var(\mathcal{F}, n_l+n_u^w, k, \delta_3) + \frac{n_u^w}{n_l+n_u^w}\hat{E}(h, D_U)
\end{aligned} \tag{25}$$

where $E(f, Mix_{\frac{n_l}{n_l+n_u^w}}(\mathcal{D}_L, \mathcal{D}_U^w)|h, D_L, D_U)$ is the generalization error of f on the distribution $Mix_{\frac{n_l}{n_l+n_u^w}}(\mathcal{D}_L, \mathcal{D}_U^w)$ corresponding to pseudo-label predictor h.

When labeled data and unlabeled data are from different distributions, for any $f \in \mathcal{F}$:

$$\begin{aligned}
&E(f, \mathcal{D}_T|h, D_L, D_U) \\
\leq &E(f, Mix_{\frac{n_l}{n_l+n_u^w}}(\mathcal{D}_L, \mathcal{D}_U^w)|h, D_L, D_U) + |p_{x,y\sim\mathcal{D}_T}(h(x) \neq y) - p_{x,y\sim Mix_{\frac{n_l}{n_l+n_u^w}}(\mathcal{D}_L, \mathcal{D}_U^w)}(h(x) \neq y)| \\
= &E(f, Mix_{\frac{n_l}{n_l+n_u^w}}(\mathcal{D}_L, \mathcal{D}_U^w)|h, D_L, D_U) + Disc(f, \mathcal{D}_T, Mix_{\frac{n_l}{n_l+n_u^w}}(\mathcal{D}_L, \mathcal{D}_U^w))
\end{aligned} \tag{26}$$

According to Equations (24) to (26), for any target predictor $f \in \mathcal{F}$, pseudo-label predictor $h \in \mathcal{H}$, $0 \leq \delta_1 \leq 1$, $0 \leq \delta_2 \leq 1$ and $0 \leq \delta_3 \leq 1$, with the probability of at least $(1-\delta_1)(1-\delta_2)(1-\delta_3)$:

$$\begin{aligned}
&E(f, \mathcal{D}_T|h, D_L, D_U) \\
\leq &E(f, Mix_{\frac{n_l}{n_l+n_u^w}}(\mathcal{D}_L, \mathcal{D}_U^w)|h, D_L, D_U) + Disc(f, \mathcal{D}_T, Mix_{\frac{n_l}{n_l+n_u^w}}(\mathcal{D}_L, \mathcal{D}_U^w)) \\
\leq &\frac{n_l}{n_l+n_u^w}\hat{E}(f, D_L) + \frac{n_u^w}{n_l+n_u^w}\hat{E}(f, \tilde{D}_U) + var(\mathcal{F}, n_l+n_u^w, k, \delta_3) \\
&+ \frac{n_u^w}{n_l+n_u^w}\hat{E}(h, D_U) + Disc(f, \mathcal{D}_T, Mix_{\frac{n_l}{n_l+n_u^w}}(\mathcal{D}_L, \mathcal{D}_U^w)) \\
\leq &\frac{n_l}{n_l+n_u^w}\hat{E}(f, D_L) + \frac{n_u^w}{n_l+n_u^w}\hat{E}(f, \tilde{D}_U^w) + var(\mathcal{F}, n_l+n_u^w, k, \delta_1) + Disc(f, \mathcal{D}_T, Mix_{\frac{n_l}{n_l+n_u^w}}(\mathcal{D}_L, \mathcal{D}_U^w)) \\
&+ \frac{n_u^w}{n_l+n_u^w}(\hat{E}(h, D_L) + var(\mathcal{H}, n_l, k, \delta_2) + var(\mathcal{H}, n_u, k, \delta_3) + Disc(h, \mathcal{D}_L, \mathcal{D}_U))
\end{aligned} \tag{27}$$

where $\hat{E}(f, \tilde{D}_U^w)$ is the weighted empirical inconsistency rate between the noisy pseudo-labels and the prediction results of f on the unlabeled dataset $\tilde{D}_U$.

B. Additional Experiments

B.1. Experiment with long-tailed unlabeled dataset with inconsistent distributions

Under the premise of inconsistent intra-class feature distributions, we consider a common case where unlabeled data follows the long-tailed class distribution. In Section 6.2, both Office-31 and VisDA-2017 have inconsistent label distribution

Table 4. Experiments on Image-CLEF with long-tailed unlabeled dataset.

Methods	C/I	C/P	I/C	I/P	P/C	P/I
Supervised	**97.13±0.62**	**97.13±0.62**	**92.80±0.81**	92.80 ± 0.81	76.47 ± 1.71	76.47 ± 1.71
FixMatch	96.27 ± 0.81	95.60 ± 0.68	**92.80±0.45**	92.87 ± 0.88	74.07 ± 2.02	75.67 ± 0.78
FlexMatch	96.27 ± 0.90	96.27 ± 0.65	91.60 ± 0.77	90.40 ± 0.90	74.33 ± 1.11	72.93 ± 1.04
UASD	97.08 ± 0.64	97.00 ± 0.47	92.40 ± 0.65	91.67 ± 0.63	73.60 ± 1.51	72.80 ± 1.45
CAFA	96.80 ± 0.50	97.00 ± 0.21	92.20 ± 0.91	93.00 ± 1.12	75.87 ± 1.80	76.40 ± 1.42
Ours	**97.13±0.73**	**97.13±0.62**	92.27 ± 0.89	**93.07±1.29**	**77.20±2.18**	**76.67±1.78**

Table 5. Experiments on Image-CLEF with open-set unlabeled dataset

Methods	C/I	C/P	I/C	I/P	P/C	P/I
Supervised	97.07 ± 0.68	97.07 ± 0.68	92.87 ± 0.83	92.87 ± 0.83	74.67 ± 1.85	74.67 ± 1.85
FixMatch	96.60 ± 1.00	96.73 ± 1.62	92.47 ± 0.80	92.80 ± 0.69	**75.47±1.82**	75.07 ± 2.26
DS3L	96.73 ± 1.14	96.80 ± 1.09	**93.27±1.36**	92.13 ± 1.51	74.60 ± 1.42	74.73 ± 1.51
UASD	96.80 ± 0.98	97.07 ± 0.65	91.67 ± 1.01	90.93 ± 0.57	73.73 ± 1.24	70.80 ± 1.57
CAFA	96.67 ± 0.62	96.60 ± 0.39	92.73 ± 0.49	**93.00±0.79**	75.40 ± 2.45	**75.60±2.32**
Ours	**97.40±0.65**	**97.13±0.50**	92.93 ± 0.86	92.93 ± 1.19	**75.47±2.33**	75.47 ± 2.51

originally. In this experiment, we use Image-CLEF whose label distribution is consistent and balanced originally. For labeled data, we use the balanced dataset directly. For unlabeled data, we denote the number of samples of class i as n_i and the ratio of the class imbalance as $\rho = \frac{n_0}{n_{k-1}}$, where $n_i = n_0 \cdot \rho^{-\frac{i}{k-1}}$ (Cao et al., 2019). We evaluate the classification performance with imbalance ratio $\rho = 10$. The number of labeled samples is set to 300. For reliability, 5 replicates of each experiment with random seed $0 \sim 4$ are performed. Our method is compared with supervised learning, classical deep SSL methods FixMatch (Sohn et al., 2020) and FlexMatch (Zhang et al., 2021), and robust deep SSL methods UASD (Chen et al., 2020) and CAFA (Huang et al., 2021). The average and standard deviation of the accuracy are reported in Table 4. This experiment shows that even if the unlabeled data are long-tailed and from other distributions, our method is still more robust than other SSL methods.

B.2. Experiment with open-set unlabeled dataset with inconsistent distributions

Under the premise of inconsistent intra-class feature distributions, we consider a common case where unlabeled samples may be from classes that are unseen in labeled samples. We conduct the experiment on Image-CLEF dataset. We denote the 12 classes from Image-CLEF as "0"∼"11". For labeled data, we only use samples from classes "0"∼"5". For unlabeled data, we use samples from all 12 classes. The number of labeled samples is set to 300. For reliability, 5 replicates of each experiment with random seed $0 \sim 4$ are performed. Our method is compared with supervised learning, classical deep SSL methods FixMatch (Sohn et al., 2020), robust deep SSL methods UASD (Chen et al., 2020), CAFA (Huang et al., 2021) and DS3L (Guo et al., 2020) which dedicates to alleviating the problem of unseen classes in unlabeled data. The average and standard deviation of the accuracy are reported in Table 5. This experiment shows that even if the unlabeled data has unseen classes and is from other distributions, our method is still more robust than other SSL methods.

Table 6. Ablation Study on VisDA-2017.

Decoupling between predictors	Debiased pseudo labels	Unrestricted sample weights		$Accuracy(\%) \pm std$	
		$w_{p(x\|y)}$	$w_{p(y)}$	R/S	S/R
×	×	×	×	77.60 ± 0.64	85.10 ± 0.65
✓	×	✓	✓	76.11 ± 1.61	84.63 ± 1.16
✓	✓	×	×	78.99 ± 0.51	85.54 ± 1.14
✓	✓	×	✓	78.96 ± 0.74	85.61 ± 1.31
✓	✓	✓	×	78.84 ± 0.84	85.64 ± 1.29
✓	✓	✓	✓	**79.15±0.39**	**85.92±1.16**

B.3. Ablation Study

Our method improves the shortcomings of previous SSL methods in three parts: decouple the pseudo-label predictor and the target predictor, debias the pseudo-labels by domain adaptation, and lift the restrictions of the weighting function. We conduct the ablation study on the VisDA-2017 dataset to understand the improvement of each part. The number of labeled samples is set to 150. For reliability, 5 replicates of each experiment with random seed $0 \sim 4$ are performed. The average and standard deviation of the accuracy are reported in Table 6.

This experiment shows that the improvements of all three parts of our method are effective. The best performance cannot be achieved without any of them.

C. Experimental Details

All experiments in Section 6.1 are conducted with a single Intel(R) Core(TM) i7-9750H CPU. For all naive SVMs used in experiments, we all use the implementation and hyperparameters provided by scikit-learn in default. For all JDOT-SVMs (Courty et al., 2017) used in experiments, all parameters shared with the naive SVM are set to be the same. Additionally, the algorithm used for transport computation is EMD, the algorithm used for optimization is Block Coordinate Descent (BCD), the number of Iterations for BCD is set to 15, and the trade-off parameter α in the loss function is set to 1.0. The detailed settings of comparison methods are shown as follows:

- Pseudo-Label: the threshold is set to 0.75 and the base learner is a naive SVM.
- Self-Training (Yarowsky, 1995): the threshold is set to 0.75, the maximum number of iterations is set to 30 and the base learner is a naive SVM.
- Lable-Spreading (Zhou et al., 2003): the hyperparameters provided by scikit-learn in default are used.

All experiments in Section 6.2 and Appendix B are conducted with 4 NVIDIA GeForce RTX 3090 GPUs and 12 NVIDIA Tesla V100 GPUs. We implement all methods in PyTorch. For all comparison methods, we referred to their official implementation and hyperparameters reported in their original paper. If the hyperparameters on the corresponding dataset are not provided for one method, we will further tune the hyperparameters for it. The detailed settings of comparison methods are shown as follows:

- Mean Teacher (Tarvainen & Valpola, 2017): the EMA decay is set to 0.999, the warmup rate of unsupervised loss w_u is set to 0.4, and the ratio of unsupervised loss λ_u is set to $\max(\frac{t}{T \cdot w}, 1.0)$ where t is current iteration and T is the number of iterations.
- FixMatch (Sohn et al., 2020): the ratio of unsupervised loss λ_u is set to 1.0, the threshold is set to 0.95, and the temperature of softmax is set to 0.5.
- FlexMatch (Zhang et al., 2021): the ratio of unsupervised loss λ_u is set to 1.0, the basic threshold is set to 0.95, the temperature of softmax is set to 0.5, and the threshold warmup mechanism is used.
- UASD (Chen et al., 2020): the epoch is set to $\lceil \frac{T \cdot B}{n_u} \rceil$ and the number of iterations per epoch is set to $\lceil \frac{n_u}{B} \rceil$ where T is the total number of iterations, B is the batch size and n_u is the number of unlabeled samples, the ratio of unsupervised loss λ_u is set to 1.0.
- CAFA (Huang et al., 2021): the base SSL algorithm used is Π-Model (Laine & Aila, 2017), the warmup rate of unsupervised loss w_u is set to $\frac{4}{15}$, The perturbation magnitude ϵ is set to 0.014 and the Beta distribution parameter α is set to 0.75, the warmup rate of adversarial loss w_a is set to $\frac{8}{15}$, the ratio of unsupervised loss λ_u is $\exp(-5 \cdot (1 - \min(\frac{t}{T \cdot w_u}, 1.0))^2)$ and the ratio of adversarial loss λ_a is $\exp(-5 \cdot (1 - \min(\frac{t}{T \cdot w_a}, 1.0))^2)$ in the t-th iteration where T is the number of iterations.
- DS3L (Guo et al., 2020): the base SSL algorithm used Pseudo-Label (Lee, 2013), a two-layer fully connected neural network whose hidden dimension is 100 is used as the weighting network, the ratio of unsupervised loss λ_u is set to 0.01.