Appendix: Semi-Supervised Learning via Weight-aware Distillation under Class Distribution Mismatch

Pan Du[1,2], Suyun Zhao[1,2,*], Zisen Sheng[1, 2], Cuiping Li[1,2], Hong Chen[1,2]
Key Lab of Data Engineering and Knowledge Engineering of MOE Renmin University of China[1]
Renmin University of China, Beijing, China[2]
{du_pan,shengzisen}@163.com, {zhaosuyun, zisen, licuiping, chong}@ruc.edu.cn

1. More Related Work

Contrastive Learning. Recently, unsupervised learning approaches based on contrastive learning have drawn the most attention due to their outstanding performance [12]. Contrastive learning aims to learn an encoder ϕ to maximize the mutual information between the different views of one instance and thus obtain the representations that contain all shared information, such as semantics [1, 3, 4, 10, 18]. Conventionally, the goal is achieved by constructing positive and negative pairs and embedding the instance(anchor) close to the positive instance while pushing it away from the negative instance in training [6]. Specifically, the positive pairs are usually the local patches and the whole images or the different augmentations of the same instance, while the negative ones are all the remaining instances. Consequently, the instances with the same semantics are aligned and thus benefit the downstream tasks. Here we briefly introduce two classical contrastive learning approaches, NCE [18] and SimCLR [3].

i) NCE. NCE [18] is an instance-level discrimination approach, which first incorporates contrastive loss(NCE loss) to discriminate different instances. Specifically, NCE considers each instance as a distinct class of its own and train the encoder to distinguish between all individual instance. Many objectives are equivalent to the NCE [1, 3, 4, 10] and prove that NCE is the lower bound of the mutual information [10], denoted by $\mathcal{I}$, as formulated in Eq.1.

$$\mathcal{I}(X;X^+) \geq \mathcal{I}_{NCE}(X;X^+) = -\frac{1}{|X^-|+1}\mathbb{E}_{X,X^+,X^-} \left[log \frac{exp\left(f(\boldsymbol{x},\boldsymbol{x}^+)\right)}{exp\left(f(\boldsymbol{x},\boldsymbol{x}^+)\right) + \sum_{\boldsymbol{x}^-\in X^-} exp\left(f(\boldsymbol{x},\boldsymbol{x}^-)\right)}\right], \tag{1}$$

where $\boldsymbol{x}$, $\boldsymbol{x}^+$, and $\boldsymbol{x}^-$ are realizations of three random variables, X, X^+, and X^-. The $\boldsymbol{x}$, $\boldsymbol{x}^+$, and $\boldsymbol{x}^-$ are called the anchor, positive instance, and negative one, respectively; $f(\cdot,\cdot) = cos(\cdot,\cdot)$, $\|\boldsymbol{x}\| = 1$. Note that the positive and negative instances are generated by a memory bank that stores all the representations in training.

ii) SimCLR. SimCLR [3] simplifies the contrastive loss by only comparing the instance in a batch and considers the augmentation of the instance as the positive one. Assume that the batch B contains N instances. Then, SimCLR conducts augmentations, such as random crop, color jitter, and horizontal, to batch B and obtains the augmented batch $\tilde{B}$ consisting of $2N$ instances. Consider the instance $\tilde{x}$ from $\tilde{B}$ as the anchor, and the positive one is $\tilde{x}^+$. The negatives are all remainder $2N-2$ instances in $\tilde{B}$. Then, the loss of SimCLR is defined as Eq.2.

$$\mathcal{L}_{SimCLR} = -\frac{1}{2N}\mathbb{E}_X \left[log \frac{exp\left(f(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{x}}^+)\right)}{\sum_{\tilde{\boldsymbol{x}}^-\in\tilde{B}^-} exp\left(f(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{x}}^-)\right)}\right]. \tag{2}$$

In our study, we adopt the encoder that is optimized by Eq.2 as the teacher model to learn a robust representation space.

2. A Theoretical Derivation

Essentially, the Eq.1 is a cross-entropy loss that aims to classify the positive instance correctly. Then, the optimal probability of the NCE loss can be defined as Eq.3 [10].

$$\begin{aligned} &P(z_j|X,z_i) \\ &= \frac{p(z_j,X|z_i)}{\sum_{r=1}^{N} p(z_r,X|z_i)} \\ &= \frac{p(z_j|z_i)\Pi_{l\neq j}p(z_l)}{\sum_{r=1}^{N} p(z_r|z_i)\Pi_{l\neq r}p(z_l)} \\ &= \frac{\frac{p(z_j|z_i)}{p(z_j)}}{\sum_{r=1}^{N}\frac{p(z_r|z_i)}{p(z_r)}}, \end{aligned} \tag{3}$$

where z_i and z_j indicate the representation of x_i and x_j, respectively, and x_j is the positive instance of x_i.

The denominator of the NCE loss is a constant, as well as Eq.3, and thus we can obtain Eq.4.

$$exp\{f(z_j, z_i)\} \propto \frac{p(z_j|z_i)}{p(z_j)}, \tag{4}$$

Then, the Eq.4 can be further denoted as Eq.5.

$$f(z_j, z_i) \propto log\left[\frac{p(z_j)|z_i)}{p(z_j)}\right]. \tag{5}$$

Therefore, the inner product of z_i and z_j is proportional to the point mutual information between them.

3. Polynomial Decay

This section provides the details about polynomial decay, which is used to dynamically adjust the α, and then reduce the gradually increased negative effect from unknown categories with the training. Then, the polynomial decay in our work is formulated as (6).

$$\alpha = \alpha_0 \times (1 - i/I)^2 \tag{6}$$

where α_0 is the initial value of α, i indicates the i-th update the α, and the I means the times to decrease α until it reach to 0. In our experiments, I is fixed as 5. Consequently, α decreases with the iteration increasing, preventing the invasion from unknown categories.

4. Experiments

This section introduces some additional experimental results and dataset details.

4.1. Dataset Details

CIFAR10 and CIFAR100. The target and unknown categories in CIFAR10 and CIFAR100 are shown in Tables 1 & 2, respectively, and the target ones in highlighted in bold.

Cross-dataset. Cross-dataset comprises subsamples from CIFAR10, CIFAR100, Flowers [9], Food-101 [2], and Places-365 [19]. Flowers has 8189 images with 102 categories, each containing images between 40 to 258. Food-101 has 101,000 images from 101 categories, while Places-365 comprises 1,803,460 training and 36,000 validation images from 365 classes. In the cross-dataset, six classes of animals ("bird," "cat," "dog," "deer," "frog," "horse") in CIFAR10 are seen as target one, while the other 668 classes from four external datasets (i.e., CIFAR100, Flowers, Places-365, Food-101) are unknown categories.

4.2. Experiments on More Mismatch Proportions

To further explore the performance of WAD, we evaluate it with 0% and 100% mismatch proportions. The 0% mismatch proportion means the unlabeled data contains no instance with unknown categories, while all instances are from unknown categories when the mismatch proportion is 100%. The results are shown in Tables 3 & 4.

From Tables 3 & 4, we have five findings. i) WAD surpasses all the compared approaches in CIFAR10 and CIFAR100 under 0% and 100% mismatch proportions. This shows the robustness of WAD. ii) WAD has a tiny improvement over the baseline under a 100% mismatch proportion on CIFAR10 and CIFAR100. The reason may be that WAD achieves tiny benefits from some instances with unknown categories similar to targets. iii) The accuracies of some approaches, such as DS^{3}L and CCSSL, are lower than the baseline on CIFAR10 with 0% mismatch proportion, as well as cross-dataset. The reason is that the pseudo-labels in those approaches heavily rely on the performance of the target classifier, which is trained on limited labeled instances. And the poor classifier easily leads to incorrect labeling. iv) As shown in Table 4, T2T outperforms WAD under the 0% mismatch proportion, but this may be attributed to the pretraining task, as T2T $\mathbf{w}\backslash\mathbf{o}$ pre. is lower than WAD. v) Although WAD's performance is worse than the baseline under the 100% mismatch proportion on the cross-dataset, WAD still surpasses all compared approaches. These results further demonstrate the robustness of WAD.

4.3. Further Analysis of DS^{3}L

In this subsection, we provide more analysis for the performance degradation of DS^{3}L.

There may be two factors that result in that. i) The DS^{3}L weighting the instances according to the consistent empirical loss, i.e., it assumes the two views of instances with unknown categories have inconsistent predictions. Supposing that the two views of an unlabeled instance are predicted to be in the same category, it is called consistent prediction. Then, to verify it, on CIFAR10 with 60% mismatch proportion, we visualize the number of instances with unknown categories, which have a consistent prediction of two views, as shown in Figure 1. The dotted line indicates the number of instances with unknown categories in unlabeled data. From Figure 1, we observe that the two views of most instances with unknown categories tend to have a consistent prediction, which is opposite to the assumption in DS^{3}L and thus result in the invasion of many instances with unknown categories in training. This may be the main reason for the performance degradation. ii) The consistent loss and weight heavily rely on the performance of the target classifier and a meta-net. Once some unlabeled instances bias the target classifier trained on limited labeled instances, the subsequently updated target classifier may be out of control and thus result in low performance even under 0% mismatch proportion, as shown in Table 3 & 4.

4.4. The Visualization of Reliable Instances

To verify how many instances selected from unlabeled data are reliable, on CIFAR10 with 60% mismatch proportion, we visualize the number of unlabeled instances with

Dataset	Categories
CIFAR10	**airplane, automobile,** bird, cat, deer, dog, frog, horse, ship, truck

Table 1. Categories of CIFAR10. The target categories are highlighted in **bold**, while the others are unknown categories.

Dataset	Categories
CIFAR100	mammals beaver, dolphin, otter, seal, whale, aquarium fish, flatfish, ray, shark, trout, orchids, poppies, roses, sunflowers, tulips, containers bottles, bowls, cans, cups, plates, apples, mushrooms, oranges, pears, sweet peppers, clock, computer keyboard, lamp, telephone, television, furniture bed, chair, couch, table, wardrobe, bee, beetle, butterfly, caterpillar, cockroach, **bear, leopard, lion, tiger, wolf,** bridge, castle, house, road, skyscraper, cloud, forest, mountain, plain, sea, **camel, cattle, chimpanzee, elephant, kangaroo,** **fox, porcupine, possum, raccoon, skunk,** crab, lobster, snail, spider, worm, baby, boy, girl, man, woman, crocodile, dinosaur, lizard, snake, turtle, **hamster, mouse, rabbit, shrew, squirrel,** maple, oak, palm, pine, willow, bicycle, bus, motorcycle, pickup truck, train, lawn-mower, rocket, streetcar, tank, tractor

Table 2. Categories of CIFAR100. The target categories are highlighted in **bold**, while the others are unknown categories.

	CIFAR10		CIFAR100	
Method	0%	100%	0%	100%
Baseline	94.33±0.45	94.33±0.45	36.98±1.79	36.98±1.79
DS^{3}L	92.08±0.89	90.82 ±2.50	24.57±4.26	23.15 ±4.67
UASD	95.17±0.41	94.95±0.25	41.62±0.60	39.28±0.95
CCSSL	86.82±0.28	80.47±1.25	42.77±0.78	37.02±1.31
T2T	-	-	43.13±0.42	32.20±1.86
T2T**w\o** pre.	-	-	44.10±0.93	37.40±6.17
ORCA	95.03±0.91	94.37±0.78	31.93±2.30	31.82±0.83
ORCA **w\o** pre.	94.62±0.76	93.62±0.38	24.58±0.45	23.40±2.81
WAD	**97.97**±**0.77**	**95.30**±**0.78**	**51.87**±**0.72**	**39.35**±**0.71**

Table 3. Experimental results on CIFAR10 and CIFAR100 under 0% and 100% mismatch proportions.

target categories in the selected top $\alpha\%$ instances. The result is shown in Figure 2. The tiny gap between the solid and dotted lines demonstrates the high reliability of the selected instances.

4.5. Experiments on Tiny-Imagenet

In this subsection, we compare WAD with more methods, such as OpenLDN [13], OpenMatch [14], OpenCos [11], and CSI [17], on Tiny-Imagenet [7], which we see 20 categories as the target and 180 as unknown. The results are presented in Table 5. As the CSI method is designed for representation learning in OOD detection, we combine it with vanilla pseudo-labeling [8] for semi-supervised tasks.

	Cross-dataset	
Method	0%	100%
Baseline	66.83±1.37	**66.83**±**1.37**
DS^{3}L	47.79±6.04	51.02±3.99
UASD	60.02±1.08	58.20±0.80
CCSSL	65.12±0.30	60.13±0.62
T2T	**69.02**±**0.34**	59.76±2.70
T2T **w\o** pre.	67.49±0.82	59.19±0.85
ORCA	65.50±1.06	62.25±0.27
ORCA **w\o** pre.	63.59±1.05	60.03±1.24
WAD	67.97±0.57	63.25±1.40

Table 4. Experimental results on cross-dataset under 0% and 100% mismatch proportions.

From Table 5, we have three findings. i) WAD consistently outperforms the baseline across various mismatch proportions, demonstrating its effectiveness in open-world datasets. ii) WAD exhibits superior performance at 0%, 20%, and 40% mismatch proportions and performs competitively with OpenMatch at 60% and 80%. This difference arises from the distinct weighting techniques. WAD considers pseudo-labeling and soft weighting equally important, while OpenMatch prioritizes filtering instances with unknown categories by hard weighting, i.e., $w = 0$

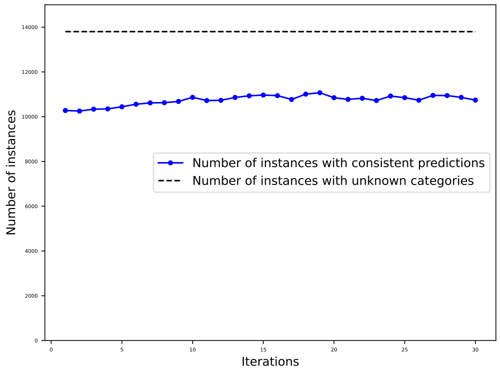

Figure 1. The number of unlabeled instances with consistent predictions in unknown categories.

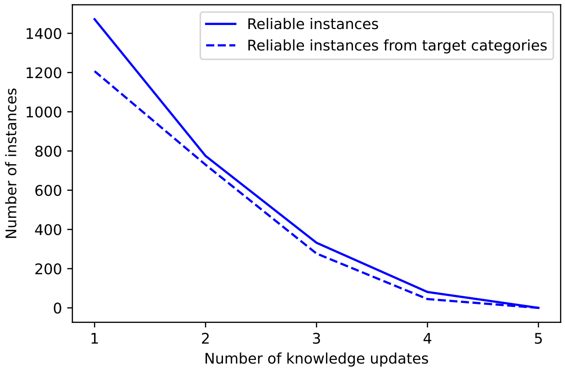

Figure 2. The visualization of the selected reliable instances.

or 1. Thus, OpenMatch, by hard weighting, may filter some instances with target categories under low mismatch proportions, while WAD may be affected by instances with unknown categories under high mismatch proportions due to soft weighting. iii) Some methods show performance decline below the baseline under 20% to 80% mismatch proportions. This may be attributed to two factors. First, the invasion error is responsible. It is observed that UASD performs better than the baseline at 0% mismatch proportion but declines at non-zero mismatch proportion due to the invasion of instances with unknown categories. Second, the pseudo-labeling error contributes to the decline. We observe that the methods, such as "CSI + vanilla pseudo-labeling," DS^{3}L, CCSSL, and ORCA, perform lower than the baseline, even at 0% mismatch proportion. Thus, to enhance the target classifier, it is crucial to address both the pseudo-labeling error and the invasion error simultaneously under class distribution mismatch.

Method	0%	20%	40%	60%	80%
Baseline	31.85±0.92	31.85±0.92	31.85±0.92	31.85±0.92	31.85±0.92
OpenLDN	34.20±0.56	33.70±0.71	33.35±0.78	32.85±0.21	32.95±0.21
OpenMatch	41.45±0.21	41.15±0.49	42.95±1.20	**41.00±0.28**	**39.55±1.20**
OpenCoS	40.65±1.06	40.45±0.35	40.55±2.62	39.85±0.78	37.15±1.61
CSI + vanilla pseudo-labeling	22.40±1.56	22.10±0.42	22.25±0.49	21.80±0.14	20.90±1.13
DS^{3}L	26.00±0.14	24.59±0.28	24.34±0.49	26.06±0.50	24.89±1.85
UASD	34.99±1.29	30.83±0.41	30.84±0.63	25.11±0.08	25.88±2.37
CCSSL	25.75±0.92	26.50±0.57	26.65±0.64	26.10±0.14	26.50±0.23
ORCA	26.50±1.27	25.25±1.34	26.85±2.05	27.20±0.42	27.70±0.21
T2T	33.10±1.56	33.95±0.07	33.60±1.13	34.55±0.07	34.70±0.28
WAD	**43.05±1.06**	**41.25±0.35**	**43.15±2.62**	39.90±0.28	37.85±0.21

Table 5. Experimental results on Tiny-ImageNet. **With 0% mismatch proportion, there are 9200 instances from target categories in the unlabeled data. At 20% mismatch proportion, the number of instances from target categories remains the same, but there are additional 2300 instances from unknown categories.**

5. Proof of WAD's SSL error

5.1. The Propertity of Lipschitz Continuity

First, we will state that the convolutional neural network(CNN) is $\lambda^l - Lipschitz\ continuous$. Then, the following definition [15] of $Lipschitz\ continuous$ with high dimensional instance x is given.

Definition 1 *A function $f : \mathcal{R}^n \rightarrow \mathcal{R}^m$ is called $Lipschitz\ continuous$ if there exists a constant L such that*

$$\forall x, y \in \mathcal{R}^n, ||f(x) - f(y)||_2 \leq L||x - y||_2.$$

The smallest L for which the previous inequality is true called the $Lipschitz\ constant$ of f and will be denoted $L(f)$.

Then we can conclude the following two Lemmas.

Lemma 1 *The Softmax function is $\lambda^s - Lipschitz\ continuous$.*

Proof 1 *We can minimize the Frobenius norm of the Jacobian matrix to solve the Lipschitz constant of the Softmax function. The softmax function is*

$$f_i(x) = \frac{exp(x_i)}{\sum_{j=1}^{K} exp(x_j)} \triangleq f_i \quad i = 1, 2, ..., K.$$

Its Jacobian matrix is

$$J = \left[\begin{array}{cccc} f_1(1-f_1) & -f_1 f_2 & \ldots & -f_1 f_K \\ -f_2 f_1 & f_2(1-f_2) & \ldots & -f_2 f_K \\ \ldots & \ldots & \ldots & \ldots \\ -f_K f_1 & -f_K f_2 & \ldots & -f_K(1-f_K) \end{array} \right].$$

And the Frobenius norm will be

$$||J||_{\mathcal{F}} = \sqrt{2\sum_{i=1}^{K}\sum_{j>i}^{K} f_i^2 f_j^2 + \sum_{i=1}^{K} f_i^2(1-f_i)^2}.$$

Then, we use the Lagrange multiplier method to solve the optimal solution for $||J||_{\mathcal{F}}$. Note the constraints are $f_1 +$

$f_2 + ... + f_K = 1$ in our work. Therefore, we can obtain the unconstrained function as,

$$F(x) = \sqrt{2\sum_{i=1}^{K}\sum_{j>i}^{K} f_i^2 f_j^2 + \sum_{i=1}^{K} f_i^2(1-f_i)^2} + \lambda(1 - f_1 - f_2 - ... - f_K),$$

where λ is a Lagrange multiplier. Furtherly, we can obtain several equalities as follows.

$$\begin{cases} \sum_{j\neq i}^{K} f_i f_j^2 + 2f_i(1-f_i)(1-2f_i) = 0, \quad \forall i = 1,2,...,K \\ f_1 + f_2 + ... + f_K = 1 \end{cases}$$

The solution is $f_i = \frac{1}{K}$,$\forall i = 1,2,...,K$. Hence, we get Lipschitz constant $L(f) = \frac{\sqrt{K-1}}{K} \triangleq \lambda^s$.

Lemma 2 *The fully-connected layer is $\lambda^c - Lipschitz$ continuous.*

Proof 2 *Assume that two inputs x_i, $x_j \in \mathcal{R}^n$, and their output at one fully-connected layer is y_i, $y_j \in \mathcal{R}^n$. The function $f : \mathcal{R}^n \to \mathcal{R}^n$ can be defined as $f(x) = y = \mathbf{W}x$. Similar to the above proof, we get the $\lambda^c = \tau$ if $||\mathbf{W}||_{\mathcal{F}} \leq \tau$. Considering that the max pooling layer and Convolutional layer are special fully-connected layers, both are $\lambda^c - Lipschitz$ continuous.*

Lemma 3 *ReLU is $1 - Lipschitz$ continuous.*

Proof 3 *ReLU$(\cdot)$ is defined as $max(0,\cdot)$, and we have*

$$\begin{aligned} &||ReLU(x_i) - ReLU(x_j)||_2 \\ &= ||max(0,x_i) - max(0,x_j)||_2 \\ &\leq ||x_i - x_j||_2. \end{aligned}$$

Thus, ReLU is $1 - Lipschitz$ continuous.
Combing $Lemma 1\&2\&3$, we can state that CNN is $\lambda^l - Lipschitz$ $continuous$.

Lemma 4 *If one Convolutional Neural Networks consists of n_c Convolutional layers, n_p max pooling layers, n_r ReLU and a softmax function, the CNN is $\lambda^l - Lipschitz$ continuous, where $\lambda^l = \lambda^s\{\lambda^c\}^{(n_c+n_p)}$.*

Proof 4 *We define the function of d^{th} layer as $h_d(x) = \mathbf{W_d}x$, so the CNN will be*

$$CNN(x) = \mathbf{W_{n_c+n_p+n_r+1}} \cdots \mathbf{W_2 W_1} x.$$

Then, we obtain,

$$\begin{aligned} &||CNN(x_i) - CNN(x_j)||_2 \\ &\leq \lambda^s\{\lambda^c\}^{(n_c+n_p)}||x_i - x_j||_2 \\ &\triangleq \lambda^l||x_i - x_j||_2. \end{aligned}$$

Thus, the CNNs are $\lambda^l - Lipschitz$ continuous.
Because the loss function can be rewritten as,

$$\begin{aligned} &|\ell(h_\theta(x_i), y_i) - \ell(h_\theta(x_j), y_j)| \\ &= \big|||CNN(x_i) - y_i||_2 - ||CNN(x_j) - y_j||_2\big| \\ &\leq ||CNN(x_i) - CNN(x_j)||_2 \\ &\leq \lambda^\ell||x_i - x_j||_2. \end{aligned}$$

where h_θ is the target model with the current parameter θ. Hence, we proof that the loss function is also $\lambda^l - Lipschitza$ continuous. In the theoretical study, we use the l_2 loss instead of the widely applied cross-entropy loss for the classification problem following [16].

5.2. The Proof of SSL Error

The population risk of the target classifier learned from both labeled and unlabeled datasets in the SSL setting, as shown in Eq.7, is controlled by the training error, generalization gap, and SSL error. The generalization gap is the gap between the population risk and the average prediction loss across all instances in T. Note that T contains all the accessible instances with target categories, including labeled and unlabeled. And every instance in T is assumed with ground truth labels in ideal. The training error is the average empirical loss across the labeled dataset and unlabeled one with pseudo labels ($\hat{T}$). It has been empirically observed that the training error can be reduced to almost zero in DNNs, and the generalization gap of DNNs can be bounded [16]. The SSL error is the gap between the average empirical loss across the instances with target categories and both the labeled dataset and unlabeled one with pseudo labels ($\hat{T}$).

$$\begin{aligned} &\mathbb{E}_{(\boldsymbol{x},y)\sim D}[l(\boldsymbol{x},y;h_{\hat{T}})] \\ &\leq \underbrace{\left|\mathbb{E}_{(\boldsymbol{x},y)\sim D}[l(\boldsymbol{x},y;h_{\hat{T}})] - \frac{1}{|T|}\sum_{(\boldsymbol{x},y)\in T} l(\boldsymbol{x},y;h_{\hat{T}})\right|}_{\textbf{generalization gap}} \\ &+ \underbrace{\left|\frac{1}{|\hat{T}|}\sum_{(\boldsymbol{x},y)\in \hat{T}} l(\boldsymbol{x},y;h_{\hat{T}})\right|}_{\textbf{training error}} \\ &+ \underbrace{\left|\frac{1}{|T|}\sum_{(\boldsymbol{x},y)\in T} l(\boldsymbol{x},y;h_{\hat{T}}) - \frac{1}{|\hat{T}|}\sum_{(\boldsymbol{x},y)\in \hat{T}} l(\boldsymbol{x},y;h_{\hat{T}})\right|}_{\textbf{SSL error}}, \end{aligned} \tag{7}$$

where $\hat{T} = \{(x_{i,l}, y_{i,l})\}_{i=1}^{m} \cup \{(x_{i,u}, \hat{y}_{i,u}\}_{i=1}^{n}$, $\mathcal{D}$ is the data distribution of the instances that belong to target categories in the realistic world, i.e., $\mathcal{D} = \mathcal{X} \times \mathcal{Y}$. $l(\cdot,\cdot;h_{\hat{T}}) : \mathcal{X} \times \mathcal{Y} \to \mathcal{R}$ denotes the loss function of the target classifier $h_{\hat{T}}$ learned from $\hat{T}$.

Thus, in the SSL setting, the essential component concerning population risk is the SSL error, which indicates the effectiveness of the target classifier learned from labeled data and unlabeled ones with pseudo labels. The SSL error can be minimized by improving the quality of pseudo labels when the labeled and unlabeled instances come from identical categories. However, under class distribution mismatch, the unlabeled data usually contains unknown categories, which invade the training of the target classifier as outliers and cause catastrophic errors for the target classifier. To circumvent this problem, we further deconstruct the SSL error as pseudo-labeling and invasion error, as shown in Eq.8.

$$\begin{aligned}
&\left|\frac{1}{|T|}\sum_{(\boldsymbol{x},y)\in T} l(\boldsymbol{x},y;h_{\hat{T}}) - \frac{1}{|\hat{T}|}\sum_{(\boldsymbol{x},y)\in \hat{T}} l(\boldsymbol{x},y;h_{\hat{T}})\right| \\
&= \Bigg|\frac{1}{|T|}\sum_{(\boldsymbol{x},y)\in T} l(\boldsymbol{x},y;h_{\hat{T}}) \\
&- \frac{1}{|\hat{T}|}\left(\sum_{(\boldsymbol{x},y)\in \hat{T}\backslash U} l(\boldsymbol{x},y;h_{\hat{T}}) + \sum_{(\boldsymbol{x},y)\in U} l(\boldsymbol{x},y;h_{\hat{T}})\right)\Bigg| \\
&= \Bigg|\frac{1}{|T|}\sum_{(\boldsymbol{x},y)\in T} l(\boldsymbol{x},y;h_{\hat{T}}) - \frac{1}{|\hat{T}|}\sum_{(\boldsymbol{x},y)\in \hat{T}\backslash U} l(\boldsymbol{x},y;h_{\hat{T}}) \\
&- \frac{1}{|\hat{T}|}\sum_{(\boldsymbol{x},y)\in U} l(\boldsymbol{x},y;h_{\hat{T}})\Bigg| \\
&\leq \underbrace{\left|\frac{1}{|T|}\sum_{(\boldsymbol{x},y)\in T} l(\boldsymbol{x},y;h_{\hat{T}}) - \frac{1}{|\hat{T}|}\sum_{(\boldsymbol{x},y)\in \hat{T}\backslash U} l(\boldsymbol{x},y;h_{\hat{T}})\right|}_{\textbf{Pseudo-labeling error}} \\
&+ \underbrace{\left|\frac{1}{|\hat{T}|}\sum_{(\boldsymbol{x},y)\in U} l(\boldsymbol{x},y;h_{\hat{T}})\right|}_{\textbf{Invasion error}}
\end{aligned} \tag{8}$$

5.2.1 The Bound for Pseudo-labeling Error

Minimizing the pseudo-labeling error is equivalent to improving the quality of the pseudo-labels. In the following, we verify that the quality of pseudo labels does bound the pseudo-labeling error.

In our work, we determine the pseudo label according to the maximum PMI. The pseudo label is defined as Eq.9.

$$\hat{y}_{i,u} = \arg\max_{k} \xi_k \tag{9}$$

wherein

$$\xi = max\xi_k,\ \xi_k = cos < z_{i,u}, z_{j,l,k} >$$

where $|z_{i,u} = 1| = |z_{j,l,k}| = 1$, $k \in \mathcal{Y}$, and $\mathcal{Y} = 1, 2, ..., K$. $z_{i,u}$ and $z_{j,l,k}$ represents the representations of the unlabeled instance $x_{i,u}$ and labeled ones $x_{j,l,k}$ with categories k respectively. ξ indicates the similarity between the unlabeled representation and its nearest labeled one and it is proportional to the PMI, that decides the pseudo label.

Moreover, weight is assigned to each unlabeled instance to prevent the ones with unknown categories from training. The weight is defined as,

$$\boldsymbol{w}_{i,u} = g_1\left(\widetilde{p}_{i,u}\right) \times g_2\left(1 - \frac{\widetilde{q}_{i,u}}{\widetilde{p}_{i,u}}\right) \tag{10}$$

wherein

$$\begin{aligned}
\widetilde{p}_{i,u} &= \max_{j} f(z_{i,u}, z_{j,l,k}) \\
\widetilde{q}_{i,u} &= \max_{v,k\neq\hat{y}_{i,u}} f(z_{i,u}, z_{v,l,k})
\end{aligned}$$

where $g_1(\cdot)$ and $g_2(\cdot)$ can be interpreted as any monotonically increasing functions. The functions $g_1(\cdot)$ and $g_2(\cdot)$, which we adopt here, are identical mappings.

Then, we calculate the Euclidean distance between $z_{i,u}$ and its nearest labeled one ($z_{j,st}$), denoted as $D_{i,u}$.

$$\begin{aligned}
D_{i,u} &= \sqrt{|z_{i,u}| + |z_{j,st}| - 2|z_{i,u}||z_{j,st}|cos < z_{i,u}, z_{j,st} >} \\
&= \sqrt{2 - 2cos < z_{i,u}, z_{j,st} >} \\
&= \sqrt{2 - 2\xi}(\xi \leq 1,\ \forall x_{i,u} \in T\backslash U)
\end{aligned} \tag{11}$$

Assume that there exists z'_j around $z_{i,u}$, and z'_j has 0 loss. We denote that the population distribution of $z_{i,u}$ is $\tau(z_i)$, and the empirical distribution of $z_{i,u}$ is $\tau_k(z_i)$. Then, we start our proof with bounding $E_{y_i\sim\tau(z_i)}[l(x_i, y_i; h_{\hat{T}})]$, which is the prediction error's expectation of the instance $x_{i,u}$.

$$\begin{aligned}
&E_{y_i\sim\tau(z_i)}[l(x_i, y_i; h_{\hat{T}})] \\
&= \sum_{k\in\mathcal{Y}} p_{y_i\sim\tau_k(z_i)}(y_i = k) l(x_i, y_i; h_{\hat{T}}) \\
&= \sum_{k\in\mathcal{Y}} p_{y_i\sim\tau_k(z_i)}(y_i = k) \boldsymbol{w}_{i,u} \ell(h_{\hat{T}}(x_i), y_i) \\
&\leq \sum_{k\in\mathcal{Y}} p_{y_i\sim\tau_k(z'_{j,st})}(y_i = k) \boldsymbol{w}_{i,u} \ell(h_{\hat{T}}(x_i), y_i) \\
&+ \sum_{k\in\mathcal{Y}} |\tau_k(z_i) - \tau_k(z'_{j,st})| \boldsymbol{w}_{i,u} \ell(h_{\hat{T}}(x_i), y_i) \\
&\underset{\boldsymbol{w}_{i,u}\leq 2}{\leq} 2\Bigg[\sum_{k\in\mathcal{Y}} p_{y_i\sim\tau_k(z'_{j,st})}(y_i = k) \ell(h_{\hat{T}}(x_i), y_i) \\
&+ \sum_{k\in\mathcal{Y}} |\tau_k(z_i) - \tau_k(z'_{j,st})| \ell(h_{\hat{T}}(x_i), y_i)\Bigg]
\end{aligned} \tag{12}$$

where $l(x_i, y_i; h_{\hat{T}}) = \boldsymbol{w}_{i,u}\ell(h_{\hat{T}}(x_i), y_i)$ is the loss function for training the target classifier $h_{\hat{T}}$ in our work and $\boldsymbol{w}_{i,u} \leq$

2. $\ell(\cdot,\cdot)$ denotes the cross-entropy loss. Note that we use the l_2 loss instead of the widely applied cross-entropy loss in the theoretical study.

For $\sum_{k\in\mathcal{Y}} |\tau_k(z_i) - \tau_k(z'_{j,st})|\ell((h_{\hat{T}}(x_i), y_i)$, we obtain

$$\begin{aligned}
&\sum_{k\in\mathcal{Y}} |\tau_k(z_i) - \tau_k(z'_{j,st})|\ell((h_{\hat{T}}(x_i), y_i) \\
&= \sum_{k\in\mathcal{Y}} D_{i,u}\cdot \ell(h_{\hat{T}}(x_i), y_i) \\
&\le \sqrt{2-2\xi}\cdot\lambda^{\mu}\frac{H}{2}K \\
&\le \sqrt{2-2\xi}\cdot\lambda^{\mu}HK
\end{aligned} \tag{13}$$

where $H/2$ is the upper bound of $\ell(\cdot,\cdot)$ and then $l(x,y;h_{\hat{T}})$ is bounded by H due to $\boldsymbol{w}_{i,u}\le 2$.
For $\sum_{k\in\mathcal{Y}} p_{y_i\sim\tau_k(z'_{j,st})}(y_i=k)\ell((h_{\hat{T}}(x_i), y_i)$, we have

$$\begin{aligned}
&\sum_{k\in\mathcal{Y}} p_{y_i\sim\tau_k(z'_{j,st})}(y_i=k)\ell((h_{\hat{T}}(x_i), y_i) \\
&= \sum_{k\in\mathcal{Y}} p_{y_i\sim\tau_k(z'_{j,st})}(y_i=k)\{\ell((h_{\hat{T}}(x_i), y_i) - \ell(h_{\hat{T}}(x_{j,st}), y_{j,st})\} \\
&+ \sum_{k\in\mathcal{Y}} p_{y_i\sim\tau_k(z'_{j,st})}(y_i=k)\ell(h_{\hat{T}}(x_{j,st}), y_{j,st}) \\
&\le \lambda^{l}\sqrt{2-2\xi}.
\end{aligned} \tag{14}$$

Hence, $E_{y_i\sim\tau(z_i)}[l(x_i,y_i;h_{\hat{T}})] \le \sqrt{4-4\xi}\cdot(\lambda^{\mu}HK+\lambda^{l})$.

Furthermore, according to the Hoeffding inequality [5], we can obtain

$$\begin{aligned}
&\mathcal{P}\left\{\left|\frac{1}{|T|}\sum_{(\boldsymbol{x},y)\in T} l(\boldsymbol{x},y;h_{\hat{T}}) - E_{y\sim\tau(x)}\left[l(x,y;h_{\hat{T}})\right]\right| \ge t\right\} \\
&\le exp\left(\frac{-2|T|^2t^2}{\sum_{i=1}^{|T|}\left(\max\left[\boldsymbol{w}_{i,u}\ell(h_{\hat{T}}(x),y)\right] - \min\left[\boldsymbol{w}_{i,u}\ell(h_{\hat{T}}(x),y)\right]\right)^2}\right) \\
&\le exp(\frac{-2|T|t^2}{H^2}) \\
&\le exp(\frac{-|T|t^2}{2H^2}).
\end{aligned} \tag{15}$$

Let $exp(\frac{-|T|t^2}{2H^2})=\gamma$. With probability $1-\gamma$, we have

$$\begin{aligned}
&\left|\frac{1}{|T|}\sum_{(\boldsymbol{x},y)\in T} l(\boldsymbol{x},y;h_{\hat{T}}) - \frac{1}{|\hat{T}|}\sum_{(\boldsymbol{x},y)\in \hat{T}\setminus U} l(\boldsymbol{x},y;h_{\hat{T}})\right| \\
&\le \sqrt{4-4\xi}(\lambda^{l}+\lambda^{\mu}HK) + \sqrt{\frac{2H^2log(1/\gamma)}{|T|}}.
\end{aligned}$$

5.2.2 The Bound for Invation Error

The unlabeled instances with unknown categories will hurt the target classifier because they invade the training of the classifier as outliers. Hence, weights are exploited to measure the role of instances in the WAD framework. In the following, we verify that the weights of unlabeled instances with unknown categories can bound the invasion error. The definition of the weight is shown in the subsubsection 5.2.1.

Then, we can obtain the Eq.16.

$$\begin{aligned}
&\left|\frac{1}{|\hat{T}|}\sum_{(\boldsymbol{x},y)\in U} l(\boldsymbol{x},y;h_{\hat{T}})\right| \\
&= \left|\frac{1}{|\hat{T}|}\sum_{(\boldsymbol{x},y)\in U} \boldsymbol{w}_{i,u}\ell(h_{\hat{T}}(x),y)\right| \\
&\le \frac{H}{2|\hat{T}|}\sum_{(\boldsymbol{x},y)\in U} \boldsymbol{w}_{i,u} \\
&\le \frac{H|U|}{2|\hat{T}|}\frac{1}{|U|}\sum_{(\boldsymbol{x},y)\in U} \boldsymbol{w}_{i,u} \\
&\le \frac{\overline{\boldsymbol{w}}|U|H}{2|\hat{T}|} \\
&\le \frac{\overline{\boldsymbol{w}}|U|H}{|\hat{T}|}.
\end{aligned} \tag{16}$$

where $\overline{\boldsymbol{w}}$ indicates the mean value with respect to $\boldsymbol{w}_{i,u}$ of the unlabeled instances with unknown categories.

5.2.3 The Bound of WAD's SSL error

For pseudo-labeling error, with probability $1-\gamma$, we have

$$\begin{aligned}
&\left|\frac{1}{|T|}\sum_{(\boldsymbol{x},y)\in T} l(\boldsymbol{x},y;h_{\hat{T}}) - \frac{1}{|\hat{T}|}\sum_{(\boldsymbol{x},y)\in \hat{T}\setminus U} l(\boldsymbol{x},y;h_{\hat{T}})\right| \\
&\le \sqrt{4-4\xi}(\lambda^{l}+\lambda^{\mu}HK) + \sqrt{\frac{2H^2log(1/\gamma)}{|T|}}.
\end{aligned}$$

For invasion error, we have

$$\left|\frac{1}{|\hat{T}|}\sum_{(\boldsymbol{x},y)\in U} l(\boldsymbol{x},y;h_{\hat{T}})\right| \le \frac{\overline{\boldsymbol{w}}|U|H}{|\hat{T}|}.$$

Therefore, for SSL error of WAD under class distribution mismatch, with probability $1-\gamma$, we can obtain that

$$\begin{aligned}&\left|\frac{1}{|T|}\sum_{(\boldsymbol{x},y)\in T} l(\boldsymbol{x},y;h_{\hat{T}})-\frac{1}{|\hat{T}|}\sum_{(\boldsymbol{x},y)\in \hat{T}\backslash U} l(\boldsymbol{x},y;h_{\hat{T}})\right|\\ &+\left|\frac{1}{|\hat{T}|}\sum_{(\boldsymbol{x},y)\in U} l(\boldsymbol{x},y;h_{\hat{T}})\right|\\ &\leq\sqrt{4-4\xi}(\lambda^{l}+\lambda^{\mu}HK)+\frac{\overline{\boldsymbol{w}}|U|H}{|\hat{T}|}+\sqrt{\frac{2H^{2}log(1/\gamma)}{|T|}}.\end{aligned} \tag{17}$$

As shown in Eq.17, the SSL error is jointly controlled by the ξ, which is proportional to the PMI and determines the pseudo labels, and $\overline{\boldsymbol{w}}$, which is the average of the weights of the instances with unknown categories. In WAD, we assign the class label of the labeled instances with the maximum PMI to the unlabeled one to maximize the ξ, and simultaneously we decrease the weight of each instance with unknown categories to minimize the $\overline{\boldsymbol{w}}$, following that the SSL error is mitigated. Therefore, WAD's SSL error has a tight upper bound.

References

[1] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. *arXiv preprint arXiv:1906.00910*, 2019. 1

[2] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *European conference on computer vision*, pages 446–461. Springer, 2014. 2

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 1

[4] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations*, 2018. 1

[5] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. In *The Collected Works of Wassily Hoeffding*, pages 409–426. Springer, 1994. 7

[6] Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. A survey on contrastive self-supervised learning. *Technologies*, 9(1):2, 2021. 1

[7] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015. 3

[8] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896, 2013. 3

[9] M-E Nilsback and Andrew Zisserman. A visual vocabulary for flower classification. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1447–1454. IEEE, 2006. 2

[10] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 1

[11] Jongjin Park, Sukmin Yun, Jongheon Jeong, and Jinwoo Shin. Opencos: Contrastive semi-supervised learning for handling open-set unlabeled data. In *European Conference on Computer Vision*, pages 134–149. Springer, 2022. 3

[12] Haocong Rao, Siqi Wang, Xiping Hu, Mingkui Tan, Yi Guo, Jun Cheng, Xinwang Liu, and Bin Hu. A self-supervised gait encoding approach with locality-awareness for 3d skeleton based person re-identification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 1

[13] Mamshad Nayeem Rizve, Navid Kardan, Salman Khan, Fahad Shahbaz Khan, and Mubarak Shah. Openldn: Learning to discover novel classes for open-world semi-supervised learning. In *European Conference on Computer Vision*, pages 382–401. Springer, 2022. 3

[14] Kuniaki Saito, Donghyun Kim, and Kate Saenko. Openmatch: Open-set consistency regularization for semi-supervised learning with outliers. *arXiv preprint arXiv:2105.14148*, 2021. 3

[15] Kevin Scaman and Aladin Virmaux. Lipschitz regularity of deep neural networks: analysis and efficient estimation. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 3839–3848, 2018. 4

[16] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations*, 2018. 5

[17] Jihoon Tack, Sangwoo Mo, Jongheon Jeong, and Jinwoo Shin. Csi: Novelty detection via contrastive learning on distributionally shifted instances. *Advances in neural information processing systems*, 33:11839–11852, 2020. 3

[18] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3733–3742, 2018. 1

[19] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017. 2