SEMANTIC-AWARE REPRESENTATION LEARNING VIA PROBABILITY CONTRASTIVE LOSS

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent feature contrastive learning (FCL) has shown promising performance in unsupervised representation learning. For the close-set representation learning where labeled data and unlabeled data belong to the same semantic space, however, FCL cannot show overwhelming gains due to not involving the class semantics during optimization. Consequently, the produced features do not guarantee to be easily classified by the class weights learned from labeled data although they are information-rich. To tackle this issue, we propose a novel probability contrastive learning (PCL) in this paper, which not only produces rich features but also enforces them to be distributed around the class prototypes. Specifically, we propose to use the output probabilities after softmax to perform contrastive learning instead of the extracted features in FCL. Evidently, such a way can exploit the class semantics during optimization. Moreover, we propose to remove the ℓ_2 normalization in the traditional FCL and directly use the ℓ_1-normalized probability for contrastive learning. Our proposed PCL is simple and effective. We conduct extensive experiments on three close-set image classification tasks, *i.e.*, unsupervised domain adaptation, semi-supervised learning, and semi-supervised domain adaptation. The results on multiple datasets demonstrate that our PCL can consistently get considerable gains and achieves the state-of-the-art performance for all three tasks.

1 INTRODUCTION

The goal of representation learning is to extract a compact feature from the raw data, and the researches in recent decades are mainly carried out under the fully supervised learning framework (Krizhevsky et al., 2012; He et al., 2016; Hu et al., 2018). However, such fully-supervised learning algorithms are generally data hungry, and the expensive data labeling limits its practical application in many real-world scenarios.

Therefore, how to utilize some labeled data to assist robust representations learning of unlabeled data in the target domain attracts a lot of research attentions. To this end, the solutions with different perspectives have been proposed that usually exploit different labeled data. For example, semi-supervised learning assumes the labeled data and unlabeled data are collected from the same distribution, and then use both of them during optimization. Domain adaptation leverages a labeled source domain to aid the unlabeled target domain, where the source and target domains have the same semantics but different data distributions. Few-shot learning aims to learn a feature extractor from labeled base set that is expected to be generalizable for new tasks. Zero-shot learning explores the relationship between attributes and classes, and then recognizes new classes based on the predicted attributes. In this work, we divide these tasks into two broad categories according to the consistency of semantic spaces between the labeled and unlabeled (target) data, namely, close-set representation learning (CLRL) and open-set representation learning (OPRL). Specifically, the semantic space of unlabeled data is defined by labeled data in CLRL, *e.g.*, semi-supervised learning and domain adaptation. For OPRL, the target tasks have different semantic space from the base data for learning, *e.g.*, few-shot learning and zero-shot learning. In this paper, we particularly focus on CLRL where the semantic space is predefined regardless of domain shift of data.

In recent years, the unsupervised techniques represented by contrastive learning (Chen et al., 2020a; Xu et al., 2020; He et al., 2020) have shown great potentials in representation learning. Generally, the normalized features before the classifier are used to calculate the contrastive loss, and so we call it feature contrast learning (FCL). In FCL, a pair of features from the same image with different transformations are regarded as positive that would be enforced closer, while all feature pairs from different images are treated as negative that are expected to be pushed away. Through such instance contrastive learning, the model can extract rich information of images. Ultimately, the trained model is used as the feature extractor, and a simple linear classifier (Chen et al., 2020b;c) or a KNN classifier (Xu et al., 2020) can achieve amazing performance. For example, SimCLRv2 (Chen et al., 2020b) can achieve over 70% top-1 accuracy with ResNet-50 (He et al., 2016) on the ImageNet (Russakovsky et al., 2015), whereas the full supervised counterpart has an accuracy of 76.6%. Due to the success of FCL in representation learning, many CLRL methods (Li et al., 2021b; Wang et al., 2021; Zhang et al., 2021c) introduce the FCL loss represented by infoNCE (Oord et al., 2018) besides the traditional cross-entropy loss. However, the optimization of FCL does not take into account the class semantics due to directly operating on the features before class weights. Consequently, the learned class weights in the CLRL tasks are usually deviated from the feature center due to the small amount or domain shift of labeled data w.r.t. the target data, as shown in Figure 1(b). To sum up, FCL cannot work well enough for CLRL although it can cluster the image features of similar semantics.

To address this problem, we propose a novel approach named probability contrastive learning (PCL) in this paper. Apart from the FCL, PCL use the output probability of classifier with softmax to construct positive and negative samples rather than the widely used features. Intuitively, the probability contains the information of image features and class weights, and thus PCL has chance to make the unlabeled features cluster around the class weight as expected. Figure 1(c) illustrates the results and we will elaborate on the details in the section 3. Furthermore, we remove the ℓ_2 normalization used in the traditional feature contrastive learning and directly use the output of Softmax. Such a form can further strengthen the similarity of the learned feature and some class weight due to the sparsity characteristic of the ℓ_1 normalization.

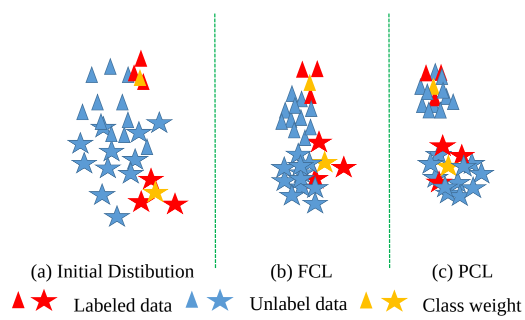

Figure 1: Feature contrastive learning *v.s.* Probability contrastive learning. (a) The distribution of initial features are relatively scattered. (b) After FCL, the features of same semantic are clustered, but the learned class weights are deviated from the class center. (c) After PCL, the features with similar semantics can be not only clustered but also distributed around the class weights.

Our main contributions are three-folds. First, we found that the traditional feature contrastive learning cannot work well in the CLRL tasks due to not involving the optimization of class weights. Second, we design a novel probability contrastive learning approach and propose to remove the ℓ_2 normalization in FCL. As a result, our method can enforce the learned features to be distributed around the class weights. Third, we conduct extensive experiments on three different CLRL tasks to verify the effectiveness of our proposed PCL, *i.e.*, SSL, UDA, and SSDA. The results shows the superiority of our method to previous state-of-art methods in simplicity and performance.

2 RELATED WORK

2.1 CONTRASTIVE REPRESENTATION LEARNING

Contrastive learning (Chen et al., 2020a; He et al., 2020; Grill et al., 2020; Caron et al., 2020) is a framework that learns similar/dissimilar representations from data that are organized into similar/dissimilar pairs. Since there is no label information, an instance discrimination pretext task (Wu et al., 2018) is used, where a query and a key form a positive pair if they are data-augmented versions of the same image. They form a negative pair otherwise. In these works, an effective contrastive loss function, called InfoNCE (Oord et al., 2018) is widely adopted.

SimCLR (Chen et al., 2020a) uses self-supervised contrastive learning to first achieve the performance of a supervised ResNet-50 with only a linear classifier trained on self-supervised representations on full ImageNet. He et al. (2020) propose MoCo and Chen et al. (2020c) extends MoCo to MoCo v2, where a small batch size can also achieve competitive results on full ImageNet (Russakovsky et al., 2015). In addition, many other methods (Grill et al., 2020; Caron et al., 2020) are also proposed to further boost performance.

Khosla et al. (2020) introduce supervised contrastive learning to encourage more compact representation. Cui et al. (2021) introduce a set of parametric class-wise learnable centers to tackle long-tailed recognition. All the above works conduct contrastive learning in feature level. In this work, we propose to use Probability Contrastive Learning (PCL) for the close-set representation learning tasks.

2.2 SEMI-SUPERVISED LEARNING

Semi-Supervised Learning (SSL) (Berthelot et al., 2019; Li et al., 2017; Dai et al., 2017) aims to leverage the vast amount of unlabeled data with limited labeled data to improve performance. Yang et al. (2021) classifies the SSL methods into five categories, *i.e.*, generative methods, consistency regularization methods, graph-based methods, pseudo-labeling methods, and hybrid methods. Recently, the consistency-based approach has attracted the attention of many reseachers. Mean teacher (Tarvainen & Valpola, 2017) uses two different models to ensure consistency across similar images. Mix-Match (Berthelot et al., 2019) and ReMixMatch (Berthelot et al., 2020) use interpolation between labeled and unlabeled data to generate perturbed features. FixMatch (Sohn et al., 2020a) achieves impressive performance by generating the confident pseudo labels of the unlabeled samples and treating them as labels for the perturbed samples. Due to the effectiveness and simplicity of FixMatch, it is also widely applied in other semi-supervised tasks, such as semantic segmentation (Chen et al., 2021; Zou et al., 2020) and object detection (Sohn et al., 2020b; Xu et al., 2021a; Tang et al., 2021). In this work, we consider semi-supervised image classification task and also use FixMatch as a strong baseline.

2.3 UNSUPERVISED DOMAIN ADAPTATION

Unsupervised Domain Adaptation (UDA) aims to transfer the knowledge from a labeled source domain to an unlabeled target domain. The mainstream approaches tend to address UDA by learning domain-invariant representation. These approaches can be categorized into two categories. One category explicitly reduces the domain discrepancy measured by some distribution discrepancy metrics. Tzeng et al. (2014); Long et al. (2015; 2017); Yan et al. (2017) measure the domain similarity in terms of Maximum Mean Discrepancy (MMD) (Borgwardt et al., 2006), while Sun et al. (2016); Sun & Saenko (2016); Peng et al. (2019) introduce metrics based on second- or higher-order statistics. Another popular line learns domain-invariant representation using adversarial training. It has been widely studied in Ganin & Lempitsky (2015); Tzeng et al. (2017); Liu et al. (2019); Saito et al. (2018); Cui et al. (2020b).

Different from the seminal UDA framework, where unlabeled target data are utilized to explicitly minimize the domain divergence, recent UDA methods (Jin et al., 2020; French et al., 2017; Tang et al., 2020) have been proposed to explore the data structure of unlabeled data. Our proposed method in this paper belongs to this type of approaches.

2.4 SEMI-SUPERVISED DOMAIN ADAPTATION

Semi-Supervised Domain Adaptation (SSDA) aims to reduce the discrepancy between the source and target distribution in the presence of limited labeled target samples. Saito et al. (2019) first proposes to align the distributions using adversarial training on entropy. Kim & Kim (2020) shows the presence of intra-domain discrepancy in the target distribution and introduced a framework to mitigate it. Jiang et al. (2020) uses consistency alongside multiple adversarial strategies on top of MME. Li & Hospedales (2020) presents a meta-learning framework for SSDA. Yang et al. (2020) breaks down the SSDA problem into two subproblems, namely, SSL in the target domain and UDA problem across the source and target domains, and then proposes to learn the optimal weights of the network using co-training. Li et al. (2021a) proposes an adversarial adaptive clustering loss to group features of unlabeled target data into clusters and perform cluster-wise feature alignment across the source and target domains. They also use FixMatch to improve the performance.

3 METHOD

In this section, we first review feature contrastive learning widely used in unsupervised learning, and then elaborate on our probability contrastive learning for the close-set representation learning. Formally, let $\mathcal{B} = \{(x_i, \tilde{x}_i)\}_{i=1}^N$ be a batch of data pairs, where N is the batch size, and x_i and $\tilde{x}_i$ are two random transformations of a sample. We define the model $M = E \circ F$ with the feature extractor E and the classifier F. Here F has the parameters $W = (\mathbf{w}_1, ..., \mathbf{w}_C)$, where C is the number of classes, and $\mathbf{w}_k$ is the class weights of the k-th class (also called class prototype). We use the E to extract the features from $\mathcal{B}$, and get $\mathcal{F} = \{(\mathbf{f}_i, \tilde{\mathbf{f}}_i)\}_{i=1}^N$.

3.1 FEATURE CONTRASTIVE LEARNING

For a query feature $\mathbf{f}_i$, the feature $\tilde{\mathbf{f}}_i$ is the positive and all other samples are the negative. Then the InfoNCE loss (Oord et al., 2018) has the following form

$$\ell_{\mathbf{f}_i^n} = -\log \frac{\exp(s\mathbf{f}_i^{n\top}\tilde{\mathbf{f}}_i^n)}{\sum_{j\neq i}\exp(s\mathbf{f}_i^{n\top}\mathbf{f}_j^n) + \sum_k \exp(s\mathbf{f}_i^{n\top}\tilde{\mathbf{f}}_k^n)}, \tag{1}$$

where $\mathbf{f}^n = \frac{\mathbf{f}}{||\mathbf{f}||_2}$ is a standard ℓ_2-normalization operation widely used in feature contrastive learning (Oord et al., 2018; Wang et al., 2021; Chen et al., 2020a), and s is the scaling factor. We can get the gradient of the loss function $\ell_{\mathbf{f}_i^n}$ with respect to the feature $\mathbf{f}_i^n$ as

$$\nabla_{\mathbf{f}_i^n} L_{\mathbf{f}_i^n} = \overbrace{-(1-\tilde{p}_{i,i})s\tilde{\mathbf{f}}_i^n}^{Close} + \overbrace{\sum_{j\neq i} p_{i,j}s\mathbf{f}_j^n + \sum_{k\neq i}\tilde{p}_{i,k}s\tilde{\mathbf{f}}_k^n}^{Separate}, \tag{2}$$

where

$$p_{i,j} = \frac{\exp(s\mathbf{f}_i^{n\top}\mathbf{f}_j^n)}{\sum_{m\neq i}\exp(s\mathbf{f}_i^{n\top}\mathbf{f}_m^n) + \sum_k \exp(s\mathbf{f}_i^{n\top}\tilde{\mathbf{f}}_k^n)}, \tag{3}$$

$$\tilde{p}_{i,j} = \frac{\exp(s\mathbf{f}_i^{n\top}\tilde{\mathbf{f}}_j^n)}{\sum_{m\neq i}\exp(s\mathbf{f}_i^{n\top}\mathbf{f}_m^n) + \sum_k \exp(s\mathbf{f}_i^{n\top}\tilde{\mathbf{f}}_k^n)}. \tag{4}$$

The first term in (2) indicates that the query $\mathbf{f}_i$ would be close to $\tilde{\mathbf{f}}_i$, and the second term indicates that $\mathbf{f}_i$ would be far away from all other negative samples. Therefore, we can understand the optimization process of FCL as reducing the distance between the query feature and the positive sample while increasing the distance with negative samples. FCL is powerful in extracting the discriminative features for image classification. Finally, a simple linear classifier or KNN classifier over the self-supervised features can classify the test samples well (Xu et al., 2020; Chen et al., 2020a).

3.2 PROBABILITY CONTRASTIVE LEARNING

In unsupervised learning, there is no classifier and it aims to learn a task-independent initialization model on unlabeled data that is expected to be well generalized to downstream tasks. So feature contrastive learning is a natural choice. However, for the CLRL tasks such as UDA, we need the unlabeled data to be correctly classified by the classifier learned on the labeled data. Therefore, the learned features need not only be tightly clustered, but also be distributed around the class weights W. Observing Equation (2), we find that the optimization process of FCL is only related to the extracted features and the classifier is not involved. Consequently, FCL can only optimize the features to cluster together with similar semantics, and cannot guarantee the extracted features are easily classified. In order to solve this problem, we propose a novel InfoNCE loss based on probability, namely probability contrastive learning. Specifically, we replace the normalized features

$\mathbf{f}_i^n$ in the Equation (1) with the probability distribution $\mathbf{p}_i^W = \texttt{softmax}(W^\top \mathbf{f}_i)$, which establishes the association between the features and class weights. Then our proposed loss is defined as

$$\ell_{\mathbf{p}_i^W} = -\log \frac{\exp(s\mathbf{p}_i^{W\top}\tilde{\mathbf{p}}_i^W)}{\sum_{j\neq i}\exp(s\mathbf{p}_i^{W\top}\mathbf{p}_j^W) + \sum_k \exp(s\mathbf{p}_i^{W\top}\tilde{\mathbf{p}}_k^W)}. \tag{5}$$

Comparing Equation (1) and Equation (5), we can see two main differences. First, Equation (5) uses the probability after Softmax for contrastive learning instead of the extracted features. Second, Equation (5) removes the ℓ_2-norm normalization. Evidently, our proposed PCL is quite concise.

3.3 How does probability contrastive learning work?

In this section, we answer a question: how can PCL make the features closer to the class weights?

To minimize Equation (5), we need to maximize $\mathbf{p}_i^{W\top}\tilde{\mathbf{p}}_i^W$. Note that both $\mathbf{p}_i^W = (p_{i,1}^W, ..., p_{i,C}^W)$ and $\tilde{\mathbf{p}}_i^W = (\tilde{p}_{i,1}^W, ..., \tilde{p}_{i,C}^W)$ are the probability distributions. Therefore, for $\forall c \in \{1, ..., c, ..., C\}$, we have $0 \leq p_{i,c}^W \leq 1$ and $0 \leq \tilde{p}_{i,c}^W \leq 1$. The ℓ_1-norm of $\mathbf{p}_i^W$ and $\tilde{\mathbf{p}}_i^W$ equals to one, *i.e.*, $||\mathbf{p}_i^W||_1 = \sum_c p_{i,c}^W = 1$ and $||\tilde{\mathbf{p}}_i^W||_1 = \sum_c \tilde{p}_{i,c}^W = 1$. Obviously, $\mathbf{p}_i^{W\top}\tilde{\mathbf{p}}_i^W = \sum_c p_{i,c}^W \tilde{p}_{i,c}^W \leq 1$. The equality holds if and only if $\mathbf{p}_i^W = \tilde{\mathbf{p}}_i^W$ and both of them have a one-hot form. In other words, $\mathbf{p}_i^W$ and $\tilde{\mathbf{p}}_i^W$ need have the following form

$$\mathbf{p}_i^W = \tilde{\mathbf{p}}_i^W = (0, .., \overbrace{1}^{c}, .., 0), \tag{6}$$

where c indicates that the probability of the c-th category is 1. Note that $\mathbf{p}_i^W$ and $\tilde{\mathbf{p}}_i^W$ are the probability obtained from the feature $\mathbf{f}_i$ and $\tilde{\mathbf{f}}_i$ with the classifier F and softmax. The c-th position of 1 means that the $\mathbf{f}_i$ and $\tilde{\mathbf{f}}_i$ are simultaneously close to the c-th class weight $\mathbf{w}_c$. Compared with FCL, this characteristic benefits PCL to enforce the features $\mathbf{f}_i$ and $\tilde{\mathbf{f}}_i$ to cluster around some class weight $\mathbf{w}_c$.

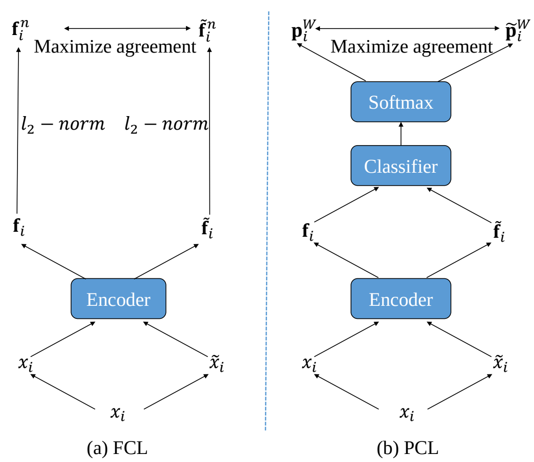

Figure 2: Framework of FCL and PCL. Different from FCL, PCL uses the output of softmax to perform contrastive learning and removes the ℓ_2-norm normalization.

Why would $\mathbf{w}_c$ be the class weight of the category corresponding to $\mathbf{f}_i$? Assuming that c_i is the ground-truth category of $\mathbf{f}_i$, let B_{c_i} denotes the set of all samples in the unlabeled data that belong to the class c_i. Since there is a high semantic similarity between labeled and unlabeled data for the same class, there are usually more features close to $\mathbf{w}_{c_i}$ in B_{c_i} than to the other $\{\mathbf{w}_{c_j}\}_{j\neq i}$. As a result, the features in B_{c_i} tend to cluster around $\mathbf{w}_{c_i}$, *i.e.*, $\mathbf{w}_c = \mathbf{w}_{c_i}$.

3.4 Loss Function

Our loss function is defined as:

$$L = L_{ori} + \lambda L_{PCL}. \tag{7}$$

Here L_{ori} represents the loss function used by the baseline model and $L_{PCL} = \sum_i(\ell_{\mathbf{p}_i^W} + \ell_{\tilde{\mathbf{p}}_i^W})$.

4 Experiments

In this section, we conduct the experiments on three close-set tasks, namely, unsupervised domain adaptation, semi-supervised learning, and semi-supervised domain adaptation. As our proposed loss can be directly applied to any existing methods by adding two augmentations on unlabeled images, we follow the experimental settings of the baseline methods, and only add the proposed loss for training.

4.1 RESULTS ON SEMI-SUPERVISED LEARNING

Setup We conduct experiments on CIFAR-10 and CIFAR-100 datasets. ***CIFAR-10 (CIFAR-100)*** (Krizhevsky et al., 2009) contains 50,000 images of size 32×32 from 10 (100) classes. We vary the amount of labeled data and focus on the label-scarce scenario where few labels are available. We take FixMatch (Sohn et al., 2020a) as our baseline, and the backbone and training hyperparameters are exactly the same with FixMatch. We evaluate on 5 runs with different random seeds, and report the mean and standard variance here. For the experimental details and hyperparameter settings, please refer to appendix A.1.1.

Comparision to SOTA We compare our method with previous state-of-the-art approaches, including "**ReMixMatch**" (Berthelot et al., 2020), "**FixMatch**" (Sohn et al., 2020a), "**CoMatch**" (Li et al., 2021b), "**SsCL**" (Zhang et al., 2021c), "**SemCo**" (Nassar et al., 2021), "**Dash**" (Xu et al., 2021b).

The quantitative evaluation results on the CIFAR-10/100 experiments are reported in Table 1. First, in the case of extremely limited labeled data (4 samples per class), our method get significant gains on CIFAR-10 (+4.8%) and CIFAR-100 (+4.0%) compared with the baseline FixMatch. Second, our method achieves the comparable performance with the baseline method when more labeled data are used. Actually, for the the case of less labeled data, it is difficult for the class weights to be accurately learned from labeled data. Owing to involving the optimization of class weights during the training, our proposed PCL can cluster the features of unlabeled samples around the class weights, and thus bring considerable gains. However, in the case of more labeled data, it is easy to learn accurate class weights from the labeled data since the labeled data and unlabeled data come from the same distribution for SSL. Consequently, our method cannot bring significant gains. It is worth noting that our performance is superior to SsCL and CoMatch that use feature contrast learning. In particular, CoMatch boosts the original feature contrastive learning through memory-smoothed pseudo-labeling and graph structure, while our method does not require any additional techniques. We believe these techniques can further enhance the performance of probability contrastive learning.

Table 1: Accuracy of SSL for CIFAR-10 and CIFAR-100 on 5 different folds. * means our reimplementation.

Method	CIFAR-10			CIFAR-100		
	40 labels	250 labels	4000 labels	400 labels	2500 labels	10000 labels
ReMixMatch(ICLR'20)	80.90±9.64	94.56±0.05	95.28±0.13	55.72±2.06	72.57±0.31	76.97±0.56
FixMatch(NeurIPS'20)	86.19±3.37	94.93±0.65	95.74±0.05	51.15±1.75	71.71±0.11	77.40±0.12
SemCo(CVPR'21)	—	94.88±0.27	**96.20±0.08**	—	68.07±0.01	75.55±0.12
CoMatch(ICCV'21)	93.09±1.39	95.09±0.33	—	—	—	—
Dash(ICML'21)	86.78±3.75	95.44±0.13	95.92±0.06	55.24±0.96	72.82±0.21	78.03±0.14
SsCL(Arxiv'21)	89.71±2.61	94.88±0.41	95.49±0.13	—	—	—
FixMatch(NeurIPS'20)*	89.26±3.92	**95.46±0.18**	96.08±0.13	53.58± 2.09	73.68±0.56	78.84±0.25
+Our PCL	**94.12±1.19**	95.44±0.05	95.96±0.09	**57.62±2.52**	**73.98±0.49**	**78.99**±0.17

4.2 RESULTS ON UNSUPERVISED DOMAIN ADAPTATION

Setup We evaluated our method in the following two standard benchmarks for UDA. ***Office-Home*** (Venkateswara et al., 2017) consists of images of everyday objects organized into four domains: Artistic (Ar), Clipart (Cl), Product (Pr), and Real-world (Rw). It contains 15,500 images of 65 classes. ***VisDA-2017*** (Peng et al., 2017) is a large-scale dataset for synthetic-to-real domain adaptation. It contains 152,397 synthetic images for the source domain and 55,388 real-world images for the target domain. We take GVB (Cui et al., 2020b) as our baseline. For the experimental details and hyperparameter settings, please refer to appendix A.1.2.

Comparision to SOTA Here we compare different representative methods, including "**DAN**" (Long et al., 2015), "**DANN**" (Ganin & Lempitsky, 2015), "**GTA**" (Sankaranarayanan et al., 2018), "**MCD**" (Saito et al., 2018), "**TAT**" (Liu et al., 2019), "**SymNet**" (Zhang et al., 2019a), "**MDD**" (Zhang et al., 2019b), "**BNM**" (Cui et al., 2020a), "**MetaAlign**" (Wei et al., 2021), "**FixBi**" (Na et al., 2021), "**CAN**" Kang et al. (2019), and "**GVB**" (Cui et al., 2020b).

Table 2: Accuracies (%) of Synthetic $\rightarrow$ Real on VisDA-2017 for unsupervised domain adaptation methods using ResNet-50.

Method	Acc	Method	Acc
DAN(arxiv'15)	61.6	DANN(ICML'15)	57.4
GTA (CVPR'18)	69.5	MDD (ICML'19)	74.6
CDAN (NeurIPS'20)	70.0	GVB (CVPR'20)	75.3
GVB*	75.0	+ FixMatch	80.4
+ Our PCL	80.8	**Our PCL** + FixMatch	**82.5**

Table 2 and table 3 give the results on VisDA-2017 and
Office-Home. In particular, we first add FixMatch on GVB, and it can achieve remarkable performance improvement. Such results are consistent with Zhang et al. (2021b) which reveals that the semi-supervised models are strong unsupervised domain adaptation learners. We further evaluate our proposed PCL by adding it to GVB and GVB with FixMatch. It can be seen that our method can bring consistent improvements and achieve state-of-the-art performance combined with FixMatch.

Table 3: Classification accuracy (%) of different UDAs on Office-Home with ResNet-50 as backbone.

Method	A→C	A→P	A→R	C→A	C→P	C→R	P→A	P→C	P→R	R→A	R→C	R→P	Avg
Source-Only	34.9	50.0	58.0	37.4	41.9	46.2	38.5	31.2	60.4	53.9	41.2	59.9	46.1
TAT (ICML'19)	51.6	69.5	75.4	59.4	69.5	68.6	59.5	50.5	76.8	70.9	56.6	81.6	65.8
SymNet (CVPR'19)	47.7	72.9	78.5	64.2	71.3	74.2	63.6	47.6	79.4	73.8	50.8	82.6	67.2
MDD (ICML'19)	54.9	73.7	77.8	60.0	71.4	71.8	61.2	53.6	78.1	72.5	60.2	82.3	68.1
BNM (CVPR'20)	56.2	73.7	79.0	63.1	73.6	74.0	62.4	54.8	80.7	72.4	58.9	83.5	69.4
FixBi (CVPR'21)	58.1	77.3	80.4	67.7	79.5	78.1	65.8	57.9	81.7	76.4	62.9	86.7	72.7
GVB (CVPR'20)	57.0	74.7	79.8	64.6	74.1	74.6	65.2	55.1	81.0	74.6	59.7	84.3	70.4
+ MetaAlign(CVPR'21)	59.3	76.0	80.2	65.7	74.7	75.1	65.7	56.5	81.6	74.1	61.1	85.2	71.3
+ **Our PCL**	59.7	75.9	80.4	69.3	75.5	77.1	67.0	58.3	81.0	75.2	63.9	84.6	72.3
+ FixMatch	59.8	78.1	81.3	67.7	78.2	76.7	68.7	60.2	**83.9**	75.1	65.5	**86.4**	73.5
+ **Our PCL** + FixMatch	**60.8**	**79.8**	**81.6**	**70.1**	**78.9**	**78.9**	**69.9**	**60.7**	83.3	**77.1**	**66.4**	85.9	**74.5**

4.3 RESULTS ON SEMI-SUPERVISED DOMAIN ADAPTATION

Setup We evaluate the effectiveness of our proposed approach on two SSDA image classification benchmarks, *i.e.*, *DomainNet* (Peng et al., 2019) and *Office-Home*. ***DomainNet*** is initially a multi-source domain adaptation benchmark. Similar to MME (Saito et al., 2019), we only select 4 domains Real, Clipart, Painting, and Sketch (abbr. **R**, **C**, **P**, and **S**), each of which contains images of 126 categories. ***Office-Home*** is a widely used UDA benchmark and consists of Real, Clipart, Art, and Product (abbr. **R**, **C**, **A**, and **P**) domains with 65 classes. For fair comparison, the settings of our benchmark datasets refer to the existing SSDA approaches (Saito et al., 2019; Qin et al., 2020; Kim & Kim, 2020), including adaptation scenarios of each dataset, the number of labeled target data (typically 1-shot or 3-shot per class), sample selection strategies, etc. In particular, we choose MME (Saito et al., 2019) as our baseline and report the results on ResNet34 (He et al., 2016) and AlexNet (Krizhevsky et al., 2012). For the experimental details and hyperparameter settings, please refer to appendix A.1.3.

Table 4: Accuracy(%) on *DomainNet* under the settings of 1-shot and 3-shot using Alexnet (A) and Resnet34 (R) as backbone networks. † means using Fixmath and ∗ means our reimplementation.

Net	Method	R→C		R→P		P→C		C→S		S→P		R→S		P→R		Mean	
		1-shot	3-shot	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot
A	S+T	43.3	47.1	42.4	45.0	40.1	44.9	33.6	36.4	35.7	38.4	29.1	33.3	55.8	58.7	40.0	43.4
	MME (ICCV'19)	48.9	55.6	48.0	49.0	46.7	51.7	36.3	39.4	39.4	43.0	33.3	37.9	56.8	60.7	44.2	48.2
	Meta-MME (ECCV'20)	-	56.4	-	50.2		51.9	-	39.6	-	43.7	-	38.7	-	60.7	-	48.8
	BiAT (IJCAI'20)	54.2	58.6	49.2	50.6	44.0	52.0	37.7	41.9	39.6	42.1	37.2	42.0	56.9	58.8	45.5	49.4
	APE (ECCV'20)	47.7	54.6	49.0	50.5	46.9	52.1	38.5	42.6	38.5	42.2	33.8	38.7	57.5	61.4	44.6	48.9
	CDAC† (CVPR'21)	56.9	61.4	**55.9**	57.5	51.6	58.9	44.8	50.7	48.1	51.7	44.1	46.7	63.8	66.8	52.1	56.2
	MME*	50.8	57.9	48.4	50.6	46.8	54.2	39.5	42.5	40.0	45.0	36.5	40.3	58.9	61.2	45.8	50.2
	+ **Our PCL**	55.1	59.5	54.6	57.2	52.4	56.7	44.2	48.2	49.6	52.6	42.0	46.9	64.2	67.1	51.7	55.5
	+ **Our PCL** + FixMatch	**58.2**	**62.5**	**55.9**	**59.3**	**57.5**	**60.6**	**47.1**	**51.2**	**51.9**	**56.0**	**44.9**	**48.8**	**65.2**	**67.8**	**54.4**	**58.0**
R	S+T	55.6	60.0	60.6	62.2	56.8	59.4	50.8	55.0	56.0	59.5	46.3	50.1	71.8	73.9	56.9	60.0
	MME (ICCV'19)	70.0	72.2	67.7	69.7	69.0	71.7	56.3	61.8	64.8	66.8	61.0	61.9	76.1	78.5	66.4	68.9
	UODA (arXiv 2020)	72.7	75.4	70.3	71.5	69.8	73.2	60.5	64.1	66.4	69.4	62.7	64.2	77.3	80.8	68.5	71.2
	Meta-MME(ECCV'20)	-	73.5	-	70.3	-	72.8	-	62.8	-	68.0	-	63.8	-	79.2	-	70.1
	BiAT (IJCAI'20)	73.0	74.9	68.0	68.8	71.6	74.6	57.9	61.5	63.9	67.5	58.5	62.1	77.0	78.6	67.1	69.7
	APE (ECCV'20)	70.4	76.6	70.8	72.1	72.9	76.7	56.7	63.1	64.5	66.1	63.0	67.8	76.6	79.4	67.6	71.7
	CDAC† (CVPR'21)	77.4	79.6	74.2	75.1	75.5	79.3	67.6	69.9	71.0	73.4	69.2	72.5	80.4	81.9	73.6	76.0
	MME*	71.0	71.4	68.9	70.0	69.2	72.6	59.8	62.7	65.6	68.2	63.2	64.3	77.8	77.9	67.9	69.5
	+ **Our PCL**	74.8	78.1	73.9	76.5	75.5	78.6	67.6	72.5	73.4	75.6	68.9	72.5	80.6	**84.6**	73.5	76.9
	+ **Our PCL** + FixMatch	**78.1**	**80.5**	**75.2**	**78.1**	**77.2**	**80.3**	**68.8**	**74.1**	**74.5**	**76.5**	**70.1**	**73.5**	**81.9**	84.1	**75.1**	**78.2**

Comparision to SOTA We compare our method with previous state-of-the-art approaches, including "**S+T**", "**MME**" (Saito et al., 2019), "**UODA**" (Qin et al., 2020), "**BiAT**" (Jiang et al., 2020), "**Meta-**

Table 5: Accuracy(%) on *Office-Home* under the setting of 3-shot using Alexnet (A) and Resnet34 (R) as backbone networks. † means using Fixmath and $*$ means our reimplementation.

Net	Method	R→C	R→P	R→A	P→R	P→C	P→A	A→P	A→C	A→R	C→R	C→A	C→P	Mean
	S+T	44.6	66.7	47.7	57.8	44.4	36.1	57.6	38.8	57.0	54.3	37.5	57.9	50.0
	MME (ICCV'19)	51.2	73.0	50.3	61.6	47.2	40.7	63.9	43.8	61.4	59.9	44.7	64.7	55.2
	Meta-MME (ECCV'20)	50.3	-	-	-	48.3	40.3	-	44.5	-	-	44.5	-	-
	BiAT (IJCAI'20)	-	-	-	-	-	-	-	-	-	-	-	-	56.4
A	APE (ECCV'20)	51.9	74.6	51.2	61.6	47.9	42.1	65.5	44.5	60.9	58.1	44.3	64.8	55.6
	CDAC† (CVPR'21)	54.9	75.8	51.8	64.3	**51.3**	43.6	65.1	47.5	63.1	63.0	44.9	65.6	56.8
	MME*	44.6	73.0	50.4	62.9	48.3	41.0	63.4	45.4	62.2	60.8	43.3	65.2	55.7
	+ Our PCL	51.5	73.6	53.1	64.9	49.3	43.7	66.0	46.7	64.5	63.8	46.0	67.2	57.5
	+ Our PCL + FixMatch	**55.5**	**77.4**	**53.6**	**67.7**	50.9	**46.5**	**69.1**	**50.7**	**67.5**	**66.2**	**47.3**	**69.4**	**60.2**
	S+T	55.7	80.8	67.8	73.1	53.8	63.5	73.1	54.0	74.2	68.3	57.6	72.3	66.2
	MME (ICCV'19)	64.6	85.5	71.3	80.1	64.6	65.5	79.0	63.6	79.7	76.6	67.2	79.3	73.1
	Meta-MME (ECCV'20)	65.2	-	-	-	64.5	66.7	-	63.3	-	-	67.5	-	-
R	APE (ECCV'20)	66.4	86.2	73.4	82.0	65.2	66.1	81.1	63.9	80.2	76.8	66.6	79.9	74.0
	CDAC† (CVPR'21)	**67.8**	85.6	72.2	81.9	**67.0**	67.5	80.3	**65.9**	80.6	80.2	67.4	81.4	74.8
	MME*	66.0	86.0	72.3	80.4	64.0	67.4	79.8	64.0	77.9	77.1	66.6	80.0	73.5
	+ Our PCL	65.4	86.7	74.5	83.1	62.9	71.0	82.8	63.7	81.0	81.1	71.0	83.1	75.5
	+ Our PCL + FixMatch	67.6	**88.7**	**75.8**	**84.1**	64.9	**73.6**	**85.7**	**65.9**	**82.1**	**82.3**	**73.0**	**82.5**	**77.2**

MME" (Li & Hospedales, 2020), "**APE**" (Kim & Kim, 2020), and "**CDAC**" (Li et al., 2021a). Here the model of the "S+T" method is trained using labeled source and labeled target data only.

Table 4 and Table 5 give the results on DomainNet and Office-Home. It can be seen that our method on all datasets and all settings can get a significant gain compared with the baseline MME*. For DomainNet, under four different settings, our method can obtain a gain of more than 5%. It strongly proves the effectiveness of our method. Furthermore, our method outperforms other methods except CDAC that adopts the FixMatch technique. For fair comparison, we also add FixMatch to our method. We can see that our method combined with FixMatch can achieve the state-of-art performance for all settings.

5 Ablation Study

In this section, we analyze our proposed PCL from the used space and normalization. As SSDA task combines the characteristics of both SSL and UDA, we particularly choose the SSDA task here. We conduct the experiments on DomainNet under the setting of 3-shot and adopt Resnet34 as the backbone.

5.1 Feature Space v.s. Probability Space

We first investigate the effect of the features at different locations. In particular, we consider the features before the classifier, which is used in the traditional feature contrastive learning (FCL), and the features before the softmax after the classifier, which is called logits contrastive learning (LCL). Following the traditional feature contrastive learning, we perform ℓ_2-norm normalization on the features and logits. Table 6 gives the results and we have the following observations. First, the traditional FCL and LCL can improve the performance of the baseline. Second, the gain of our probability contrastive learning over FCL and LCL is more than 5%, which indicates the importance of applying softmax function, *i.e.*, probability contrastive learning is essentially helpful.

Table 6: Ablation study on effect of different features on DomainNet under the setting of 3-shot and Resnet34.

Method	R→C	R→P	P→C	C→S	S→P	R→S	P→R	Mean
Baseline	71.4	70.0	72.6	62.7	68.2	64.3	77.9	69.5
FCL	72.5	71.6	73.1	66.4	70.2	64.5	80.8	71.3
LCL	72.8	70.6	72.5	66.4	70.5	64.5	81.3	71.2
PCL	**78.1**	**76.5**	**78.6**	**72.5**	**75.6**	**72.5**	**84.6**	**76.9**

5.2 Effect of ℓ_2-norm Normalization

In this section, we investigate whether our PCL requires ℓ_2-norm normalization like the standard FCL. Talble 7 gives the results. It can be seen that the accuracy would be reduced by 2% if the probabilities are normalized by the ℓ_2-norm. This is because the ℓ_2 normalization on the probability only make a pair of features keep the same direction for the inner product of 1, and it is no longer necessary to enforce them keep in the one-hot form. Therefore, the ℓ_2-norm normalization reduces the proximity of the learned features to the class weights.

5.3 t-SNE VISUALIZATION

Figure 3 shows the relationship between the extracted unlabeled features and the class weights for the three methods, including MME, MME+FCL, and MME+PCL, respectively. Firstly, compared to MME, MME+FCL produce more compact feature clusters for the same category and more separate feature distributions for different categories. However, for both MME+FCL and MME, the learned class weights are deviated from the feature centers. Secondly, the class weights of MME+PCL are closer to the feature centers than MME+FCL. It demonstrates that probability contrastive learning is effective in enforcing the features closer to the class weights.

Table 7: Ablation study on effect of applying ℓ_2-norm normalization to probability, where DomainNet under the setting of 3-shot and Resnet34 are used.

Method	R→C	R→P	P→C	C→S	S→P	R→S	P→R	Mean
Baseline	71.4	70.0	72.6	62.7	68.2	64.3	77.9	69.5
ℓ_2-norm	75.1	74.4	76.2	70.3	73.5	69.9	82.5	74.6
w/o ℓ_2-norm	**78.1**	**76.5**	**78.6**	**72.5**	**75.6**	**72.5**	**84.6**	**76.9**

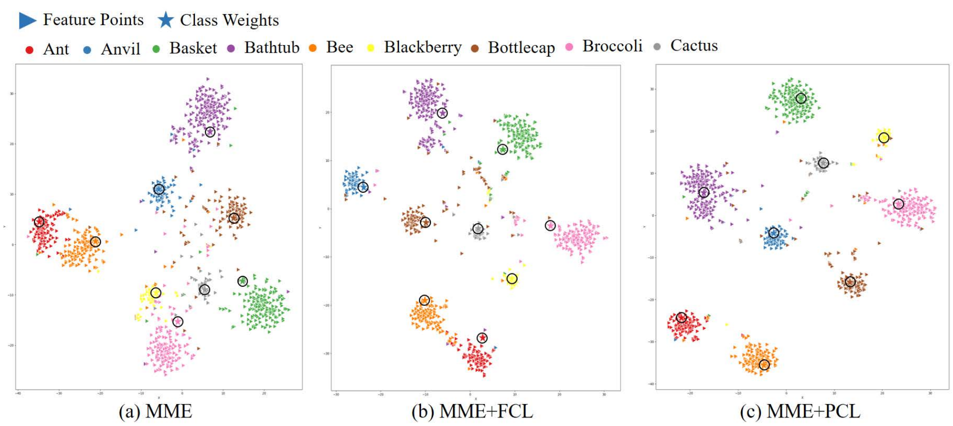

Figure 3: The t-SNE visualization of learned features. We focus on the relationship between features and class weights on the C→S task of DomainNet dataset with Resnet34 under the setting of 3-shot. Best viewed in color.

6 CONCLUSION

In this paper, we found that the traditional feature contrastive learning can only cluster the features of similar semantics and cannot enforce the learned features to be distributed around the class prototypes due to the class weights are not involved during optimization. To solve this problem, we propose a novel probability contrastive learning. Specifically, we use the probabilities after Softmax instead of the features, and remove the ℓ_2-norm normalization widely used in FCL. We experimentally verified the effectiveness of our proposed methods in three CLRL tasks with multiple datasets. We believe that our PCL provides an innovative route for representation learning in various visual tasks.

7 REPRODUCIBILITY STATEMENT

We have submitted the source code as supplementary material to facilitate the verification of the method's reproducibility.

REFERENCES

David Berthelot, Nicholas Carlini, Ian J. Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *NeurIPS*, 2019.

David Berthelot, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. In *ICLR*, 2020.

Karsten M Borgwardt, Arthur Gretton, Malte J Rasch, Hans-Peter Kriegel, Bernhard Schölkopf, and Alex J Smola. Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics*, 22(14):e49–e57, 2006.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *NeurIPS*, 2020.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020a.

Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. *arXiv preprint arXiv:2006.10029*, 2020b.

Xiaokang Chen, Yuhui Yuan, Gang Zeng, and Jingdong Wang. Semi-supervised semantic segmentation with cross pseudo supervision. In *CVPR*, 2021.

Xinlei Chen, Haoqi Fan, Ross B. Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *CoRR*, 2020c.

Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. 2020 ieee. In *CVPRW*, 2020.

Jiequan Cui, Zhisheng Zhong, Shu Liu, Bei Yu, and Jiaya Jia. Parametric contrastive learning. *arXiv preprint arXiv:2107.12028*, 2021.

Shuhao Cui, Shuhui Wang, Junbao Zhuo, Liang Li, Qingming Huang, and Qi Tian. Towards discriminability and diversity: Batch nuclear-norm maximization under label insufficient situations. In *CVPR*, 2020a.

Shuhao Cui, Shuhui Wang, Junbao Zhuo, Chi Su, Qingming Huang, and Qi Tian. Gradually vanishing bridge for adversarial domain adaptation. In *CVPR*, 2020b.

Zihang Dai, Z. Yang, Fan Yang, William W. Cohen, and R. Salakhutdinov. Good semi-supervised learning that requires a bad gan. *ArXiv*, abs/1705.09783, 2017.

Geoffrey French, Michal Mackiewicz, and Mark Fisher. Self-ensembling for visual domain adaptation. *arXiv preprint arXiv:1706.05208*, 2017.

Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML*, 2015.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. In *NeurIPS*, 2020.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.

Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, 2018.

Pin Jiang, Aming Wu, Yahong Han, Yunfeng Shao, and Bingshuai Li. Bidirectional adversarial training for semi-supervised domain adaptation. In *IJCAI*, 2020.

Ying Jin, Ximei Wang, Mingsheng Long, and Jianmin Wang. Minimum class confusion for versatile domain adaptation. In *ECCV*, 2020.

Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. In *CVPR*, 2019.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *NeurIPS*, 2020.

Taekyung Kim and Changick Kim. Attract, perturb, and explore: Learning a feature alignment network for semi-supervised domain adaptation. In *ECCV*, 2020.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012.

Chongxuan Li, T. Xu, J. Zhu, and B. Zhang. Triple generative adversarial nets. *ArXiv*, abs/1703.02291, 2017.

Da Li and Timothy Hospedales. Online meta-learning for multi-source and semi-supervised domain adaptation. In *ECCV*, 2020.

Jichang Li, Guanbin Li, Yemin Shi, and Yizhou Yu. Cross-domain adaptive clustering for semi-supervised domain adaptation. In *CVPR*, 2021a.

Junnan Li, Caiming Xiong, and Steven Hoi. Comatch: Semi-supervised learning with contrastive graph regularization. *ICCV*, 2021b.

Hong Liu, Mingsheng Long, Jianmin Wang, and Michael Jordan. Transferable adversarial training: A general approach to adapting deep classifiers. In *ICML*, 2019.

Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I Jordan. Learning transferable features with deep adaptation networks. In *ICML*, 2015.

Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *ICML*, 2017.

Jaemin Na, Heechul Jung, Hyung Jin Chang, and Wonjun Hwang. Fixbi: Bridging domain spaces for unsupervised domain adaptation. In *CVPR*, 2021.

Islam Nassar, Samitha Herath, Ehsan Abbasnejad, Wray Buntine, and Gholamreza Haffari. All labels are not created equal: Enhancing semi-supervision via label grouping and co-training. In *CVPR*, 2021.

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

Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. Visda: The visual domain adaptation challenge. In *arXiv preprint arXiv:1710.06924*, 2017.

Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *ICCV*, 2019.

Can Qin, Lichen Wang, Qianqian Ma, Yu Yin, Huan Wang, and Yun Fu. Opposite structure learning for semi-supervised domain adaptation. *arXiv preprint arXiv:2002.02545*, 2020.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, and Michael Bernstein. ImageNet large scale visual recognition challenge. *IJCV*, 2015.

Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *ECCV*, 2010.

Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *CVPR*, 2018.

Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, and Kate Saenko. Semi-supervised domain adaptation via minimax entropy. In *ICCV*, 2019.

Swami Sankaranarayanan, Yogesh Balaji, Carlos D Castillo, and Rama Chellappa. Generate to adapt: Aligning domains using generative adversarial networks. In *CVPR*, 2018.

Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *NeurIPS*, 2020a.

Kihyuk Sohn, Zizhao Zhang, Chun-Liang Li, Han Zhang, Chen-Yu Lee, and Tomas Pfister. A simple semi-supervised learning framework for object detection. *arXiv preprint arXiv:2005.04757*, 2020b.

Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *ECCV*, 2016.

Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation. In *AAAI*, 2016.

Hui Tang, Ke Chen, and Kui Jia. Unsupervised domain adaptation via structurally regularized deep clustering. In *CVPR*, 2020.

Yihe Tang, Weifeng Chen, Yijun Luo, and Yuting Zhang. Humble teachers teach better students for semi-supervised object detection. In *CVPR*, 2021.

Antti Tarvainen and H. Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *NIPS*, 2017.

Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.

Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *CVPR*, 2017.

Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, , and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *CVPR*, 2017.

Rui Wang, Zuxuan Wu, Zejia Weng, Jingjing Chen, Guo-Jun Qi, and Yu-Gang Jiang. Cross-domain contrastive learning for unsupervised domain adaptation. *arXiv preprint arXiv:2106.05528*, 2021.

Guoqiang Wei, Cuiling Lan, Wenjun Zeng, and Zhibo Chen. Metaalign: Coordinating domain alignment and classification for unsupervised domain adaptation. In *CVPR*, 2021.

Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, 2018.

Haohang Xu, Xiaopeng Zhang, Hao Li, Lingxi Xie, Hongkai Xiong, and Qi Tian. Hierarchical semantic aggregation for contrastive representation learning. *arXiv e-prints*, pp. arXiv–2012, 2020.

Mengde Xu, Zheng Zhang, Han Hu, Jianfeng Wang, Lijuan Wang, Fangyun Wei, Xiang Bai, and Zicheng Liu. End-to-end semi-supervised object detection with soft teacher. *arXiv preprint arXiv:2106.09018*, 2021a.

Yi Xu, Lei Shang, Jinxing Ye, Qi Qian, Yu-Feng Li, Baigui Sun, Hao Li, and Rong Jin. Dash: Semi-supervised learning with dynamic thresholding. In *ICML*, 2021b.

Hongliang Yan, Yukang Ding, Peihua Li, Qilong Wang, Yong Xu, and Wangmeng Zuo. Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation. In *CVPR*, 2017.

L. Yang, Yan Wang, Mingfei Gao, Abhinav Shrivastava, Kilian Q. Weinberger, Wei-Lun Chao, and Ser-Nam Lim. Deep co-training with task decomposition for semi-supervised domain adaptation. In *CVPR*, 2020.

Xiangli Yang, Zixing Song, Irwin King, and Zenglin Xu. A survey on deep semi-supervised learning. *arXiv preprint arXiv:2103.00550*, 2021.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In Richard C. Wilson, Edwin R. Hancock, and William A. P. Smith (eds.), *BMVC*, 2016.

Pan Zhang, Bo Zhang, Ting Zhang, Dong Chen, Yong Wang, and Fang Wen. Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation. In *CVPR*, 2021a.

Yabin Zhang, Hui Tang, Kui Jia, and Mingkui Tan. Domain-symmetric networks for adversarial domain adaptation. In *CVPR*, 2019a.

Yabin Zhang, Haojian Zhang, Bin Deng, Shuai Li, Kui Jia, and Lei Zhang. Semi-supervised models are strong unsupervised domain adaptation learners. *arXiv preprint arXiv:2106.00417*, 2021b.

Yuchen Zhang, Tianle Liu, Mingsheng Long, and Michael Jordan. Bridging theory and algorithm for domain adaptation. In *ICML*, 2019b.

Yuhang Zhang, Xiaopeng Zhang, Robert Qiu, Jie Li, Haohang Xu, Qi Tian, et al. Semi-supervised contrastive learning with similarity co-calibration. *arXiv preprint arXiv:2105.07387*, 2021c.

Yang Zou, Zhiding Yu, Xiaofeng Liu, BVK Kumar, and Jinsong Wang. Confidence regularized self-training. In *ICCV*, 2019.

Yuliang Zou, Zizhao Zhang, Han Zhang, Chun-Liang Li, Xiao Bian, Jia-Bin Huang, and Tomas Pfister. Pseudoseg: Designing pseudo labels for semantic segmentation. *arXiv preprint arXiv:2010.09713*, 2020.

A APPENDIX

A.1 EXPERIMENTAL DETAILS AND MORE RESULTS

A.1.1 SEMI-SUPERVISED LEARNING

Following Berthelot et al. (2019); Sohn et al. (2020a), we report the performance of an EMA model and use a Wide ResNet-28-2 (Zagoruyko & Komodakis, 2016) for CIFAR-10 and use a WRN-28-8 for Cifar100. The models are trained using SGD with a momentum of 0.9 and a weight decay of 0.0005. We follow the original papers Berthelot et al. (2019); Sohn et al. (2020a) and train all models for 1024 epochs, using an learning rate of 0.03 with a cosine decay schedule. For the hyperparameters exist in FixMatch, we follow FixMatch and set $\lambda_{cls} = 1$, $\tau = 0.95$, $\mu = 7$, $B = 64$. For the hyper-parameters in PCL, we set $s = 4$ for CIFAR-10 and $s = 7$ for CIFAR-100. For CIFAR-100, we set $\lambda = 0.05$ for all setting. For CIFAR-10, we set $\lambda = 0.02$ under the setting of 40 labels and $\lambda = 0.002$ under the setting of 250 labels and 4000 labels.

A.1.2 UNSUPERVISED DOMAIN ADAPTATION

Following the standard transductive setting (Zhang et al., 2021b) for UDA, we use all labeled source data and all unlabeled target data, and test on the same unlabeled target data. We adopt the GVB-GD Cui et al. (2020b) architecture which means adopt GVB on both the generator and discriminator, and follow the original experimental settings in GVB-GD. For model selection, we use ResNet-50 (He et al., 2016) pre-trained on ImageNet as the backbone network for both Office-Home and VisDA-2017. For training hyper-parameters, we use mini-batch stochastic gradient descent (SGD) with a momentum of 0.9, a weight decay of 0.001. For Office-Home, the initial learning rate is set to 0.001. For VisDA-2017, an initial learning rate of 0.0003 is used. The max iteration number is set to 20k.

When applying FixMatch in domain adaptation task, there exist wrong predicted high confident target samples, which can hurt the performance in the target domain. To amend this, we follow (Zhang et al., 2021a) which uses a regularization term from Zou et al. (2019). It encourages the high confident output to be evenly distributed to all classes.

$$\ell_{reg} = -\sum_{i=1}^{N}\sum_{j=1}^{C}\frac{1}{C}\log p_t^{(i,j)}. \tag{A.1}$$

where N denotes for the number of high confident samples, C denotes for the number of classes.

For data augmentation in FixMatch and the proposed PCL, we use the RandAugment (Cubuk et al., 2020) to generate strong augmented images. For the hyper-parameters in PCL, we set $s = 7.0$ and $\lambda = 0.05$.

We also consider a small scale dataset Office-31 (Saenko et al., 2010), which is another standard benchmark for visual domain adaptation. It contains 4652 images in 31 categories, and can be split into three domains: Amazon (A), DSLR (D) and Webcam (W). For this dataset, we choose a stronger baseline CAN (Kang et al., 2019), and directly add our proposed PCL in the unlabeled images. The experimental results are show in Table A.1. Compared with our re-implemented CAN, the proposed PCL can further boost the accuracy by 0.7%.

Table A.1: Classification accuracy (mean ± std %) of different UDAs on Office31 with ResNet-50 as backbone. $*$ means our reimplementation.

Method	A → W	A → D	W → A	W → D	D → A	D → W	Avg
Source-Only	68.4±0.2	68.9±0.2	60.7±0.3	99.3±0.1	62.5±0.3	96.7±0.1	76.1
SymNet (CVPR'19)	90.8±0.1	93.9±0.5	72.5±0.5	100.0±.0	74.6±0.6	98.8±0.3	88.4
BNM (CVPR'20)	92.8	92.9	73.8	100.0	73.5	98.8	88.6
MDD (ICML'19)	94.5±0.3	93.5±0.2	72.2±0.1	100.0±.0	74.6±0.3	98.4±0.1	88.9
CAN (CVPR'20)	94.5±0.3	95.0±0.3	77.0±0.3	99.8±0.2	78.0±0.3	99.1±0.2	90.6
FixBi (CVPR'21)	96.1±0.2	95.0±0.4	79.4±0.3	100.0±0.0	78.7±0.5	99.3±0.2	91.4
CAN* (CVPR'20)	94.1±0.3	94.4±0.3	75.0±0.3	99.8±0.2	77.5±0.3	98.5±0.2	89.9
+PCL	94.8±0.3	94.8±0.3	77.5±0.3	99.8±0.2	78.6±0.3	98.5±0.2	90.7

A.1.3 SEMI-SUPERVISED DOMAIN ADAPTATION

Following MME (Saito et al., 2019), we remove the last linear layer of AlexNet and ResNet34, while adding a new classifier F. We also use the model pre-trained on ImageNet to initialize all layers except F. We adopt SGD with momentum of 0.9 and set the initial learning rate is 0.01 for fully-connected layers whereas it is set 0.001 for other layers. The max iteration number is set to 50k. For the hyper-parameters in PCL, we set $s = 7$ in all experiments. In DomainNet, we set $\lambda = 0.05$ under the setting of 1-shot for ResNet34 and AlexNet and set $\lambda = 0.1$ under the setting of 3-shot for AlexNet and set $\lambda = 0.2$ under the setting of 3-shot for ResNet34. In Office-Home, we set $\lambda = 0.02$ under the setting of 3-shot for AlexNet and set $\lambda = 0.2$ under the setting of 3-shot for ResNet34. In the SSDA task, we also use the regularization term in equation A.1.

Here we report the result on Office-Home under the setting of 1-shot. we set $\lambda = 0.02$ under the setting of 3-shot for AlexNet and set $\lambda = 0.05$ under the setting of 3-shot for ResNet34. Table A.2 gives the results. It can be seen that our method can also improve the performance on AlexNet and ResNet34.

Table A.2: Accuracy(%) on *Office-Home* under the setting of 1-shot using Alexnet (denote as A) and Resnet34 (denote as R) as backbone networks. $*$ means our reimplementation.

Net	Method	R→C	R→P	R→A	P→R	P→C	P→A	A→P	A→C	A→R	C→R	C→A	C→P	Mean
A	S+T	37.5	63.1	44.8	54.3	31.7	31.5	48.8	31.1	53.3	48.5	33.9	50.8	44.1
	MME	42.0	69.6	48.3	58.7	37.8	34.9	52.5	36.4	57.0	54.1	39.5	59.1	49.2
	BiAT	-	-	-	-	-	-	-	-	-	-	-	-	49.6
	MME^*	**44.6**	69.2	49.0	60.7	**40.0**	35.7	56.4	**38.7**	57.2	56.0	40.4	59.7	50.6
	+PCL	43.1	**71.1**	**50.5**	**61.2**	38.4	**38.4**	**61.1**	36.3	**59.8**	**58.7**	**42.1**	**63.2**	**52.0**
	MME^*	**62.6**	83.1	72.3	78.8	**58.5**	64.6	75.4	**60.5**	76.9	73.7	64.9	75.0	70.5
R	+PCL	59.4	**83.7**	**73.4**	**80.4**	54.9	**66.9**	**77.9**	58.1	**79.4**	**77.7**	**68.7**	**78.9**	**71.6**