Local Prediction Aggregation: A Frustratingly Easy Source-free Domain Adaptation Method

Shiqi Yang [1] **Yaxing Wang** [2] **Kai Wang** [1] **Joost van de Weijer** [1] **Shangling Jui** [3]

Abstract

We propose a simple but effective source-free domain adaptation (SFDA) method. Treating SFDA as an unsupervised clustering problem and following the intuition that local neighbors in feature space should have more similar predictions than other features, we propose to optimize an objective of prediction consistency. This objective encourages local neighborhood features in feature space to have similar predictions while features farther away in feature space have dissimilar predictions, leading to efficient feature clustering and cluster assignment simultaneously. For efficient training, we seek to optimize an upper-bound of the objective which contains two simple terms. Furthermore, we relate popular existing methods in domain adaptation, source-free domain adaptation and contrastive learning via the perspective of discriminability and diversity. The experimental results prove the superiority of our method, and our method can be adopted as a simple but strong baseline for future research in SFDA.

1. Introduction

Supervised learning methods which are based on training with huge amounts of labeled data are advancing almost all fields of computer vision. However, the learned models typically perform decently on test data which have a similar distribution with the training set. Significant performance degradation will occur if directly applying those models to a new domain different from the training set, where the data distribution (such as variation of background, styles or camera parameter) is considerably different. This kind of distribution shift is formally denoted as domain/distribution shift. It limits the generalization of the model to unseen domains which is important in real-world applications. There are several research fields trying to tackle this problem. One of them is *Domain Adaptation* (DA), which aims to reduce the domain shift between the labeled source domain and unlabeled target domain. Typical works (Gong et al., 2012; Pan & Yang, 2009) resort to learn domain-invariant features, thus improving generalization ability of the model between different domains. And in the past few years, the main research line of domain adaptation is either trying to minimize the distribution discrepancy between two domains (Long et al., 2018a; 2015; 2016), or deploying adversarial training on features to learn domain invariant representation (Tzeng et al., 2017; Zhang et al., 2019b; Cicek & Soatto, 2019; Lu et al., 2020). Some methods also tackle domain shift from the view of semi-supervised learning (Zhang et al., 2020; Liang et al., 2021a) or clustering (Deng et al., 2019; Tang et al., 2020; Cui et al., 2020).

Although those domain adaptation methods achieve good results, for many real-world tasks the accessibility of labeled source data cannot be ensured. With the growing attention for data privacy and intellectual property from both daily users and business, often it is impossible to have constant access to source domain data, for example when deploying recognition/diagnosis model to countless mobile terminals or hospital severs where devices are decentralized. Due to its high practical value, *source-free domain adaptation* (SFDA) (Li et al., 2020; Liang et al., 2020b) has been proposed to deal with situations where those privacy or property issues need to be carefully considered. Under the SFDA setting, we are provided with a model which is already pretrained on the source domain, and during whole adaptation period we only have access to unlabeled target data. In terms of final goal, the SFDA setting is close to unsupervised clustering, where the model is expected to give the right predictions after training on unlabeled data. However, SFDA has a huge advantage over typical unsupervised clustering, that is the source model provides a good initialization for target adaptation. In other words the source-pretrained model already learned a good feature representation, since the target domain shares some similarity with the source domain. A successful SFDA method should fully exploit source information for target adaptation.

[1]Computer Vision Center, Autonomous University of Barcelona, Spain [2]Nankai University, China [3]Huawei Kirin Solution, China. Correspondence to: Yaxing Wang <yaxing@nankai.edu.cn>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

To address SFDA from the view of unsupervised clustering, we propose a simple solution dubbed as LPA (Local Prediction Aggregation). Based on the fact that target features from the source model already form some semantic structure and following the intuition that for a target feature from (source-pretrained) model, local neighbors in feature space should have more similar prediction than other features, we propose to minimize an objective function which encourages local prediction aggregation; it encourages similar features in feature space to have similar prediction, while dissimilar features to have dissimilar prediction. Therefore, we can efficiently and simultaneously cluster target features and do cluster assignment. To accelerate and simplify the training, we upper-bound this objective, resulting in a simple final objective which only contains two types of dot product terms. Further, we unify several popular domain adaptation, source-free domain adaptation and contrastive learning methods from the perspective towards discriminability and diversity. Experimental results on several benchmarks prove the superiority of our proposed method. We improve the state-of-the-art on the challenging VisDA with 2.1% to 88.0%.

2. Related Work

Domain Adaptation. Early DA methods such as (Long et al., 2015; Sun et al., 2016; Tzeng et al., 2014) adopt moment matching to align feature distributions. For adversarial learning methods, DANN (Ganin et al., 2016) formulates domain adaptation as an adversarial two-player game. The adversarial training of CDAN (Long et al., 2018b) is conditioned on several sources of information. DIRT-T (Shu et al., 2018) performs domain adversarial training with an added term that penalizes violations of the cluster assumption. Additionally, (Lee et al., 2019; Lu et al., 2020; Saito et al., 2018) adopts prediction diversity between multiple learnable classifiers to achieve local or category-level feature alignment between source and target domains. SRDC (Tang et al., 2020) proposes to directly uncover the intrinsic target discrimination via discriminative clustering to achieve adaptation. CST (Liu et al., 2021) proposes a simple self-training strategy to improve the rough pseudo label under domain shift. LAMDA (Le et al., 2021) develops a new theoretical setting to investigate label shift in domain adaptation and achieves good results on current benchmarks.

Source-free Domain Adaptation. The above-mentioned normal domain adaptation methods need to access source domain data at all time during adaptation. In recent years plenty of methods emerge trying to tackle source-free domain adaptation. USFDA (Kundu et al., 2020a) and FS (Kundu et al., 2020b) resort to synthesize extra training samples in order to get compact decision boundaries, which is beneficial for both the detection of open classes and also target adaptation. SHOT (Liang et al., 2020a) proposes to freeze the source classifier and it clusters target features by maximizing mutual information along with pseudo labeling for extra supervision. 3C-GAN (Li et al., 2020) synthesizes labeled target-style training images. It is based on a conditional GAN to provide supervision for adaptation. A^2Net (Xia et al., 2021) proposes to learn an additional target-specific classifier for hard samples and adopts a contrastive category-wise matching module to cluster target features. HCL (Huang et al., 2021a) adopts Instance Discrimination (Wu et al., 2018) for features from current and historical models to cluster features, along with a generated pseudo label conditioned on historical consistency. G-SFDA (Yang et al., 2021b) and NRC (Yang et al., 2021a) propose neighborhood clustering which enforces prediction consistency between local neighborhood features.

Deep Clustering and Contrastive Learning. Recent Deep Clustering methods can be roughly divided into two groups, they the differ in how they learn the feature representation and cluster assignments, either simultaneously or alternatively. For example, DAC (Chang et al., 2017) and DCCM (Wu et al., 2019) alternately update cluster assignments and between-sample similarity. Simultaneous clustering methods IIC (Ji et al., 2019) and ISMAT (Hu et al., 2017) are based on mutual information maximizing between samples and theirs augmentations. Recent unsupervised clustering works (Li et al., 2021b; Tsai et al., 2021; Shen et al., 2021) start to rely on contrastive learning, where InfoNCE (Oord et al., 2018) is typically deployed. And recently NNCLR (Dwibedi et al., 2021) proposes to use nearest neighbors in the latent space as positives in contrastive learning to cover more semantic variations than pre-defined transformations. However an inevitable problem of normal contrastive learning is class collision where negative samples are from the same class. To tackle this issue, recent works (Li et al., 2021a; Huang et al., 2021b) propose to estimate cluster prototypes and integrate them into contrastive learning.

3. Method

For source-free domain adaptation (SFDA), we are given source-pretrained model in the beginning and an unlabeled target domain with N_t samples as $\mathcal{D}_t = \{x_i^t\}_{i=1}^{N_t}$. Target domain have same C classes as source domain in this paper (known as the closed-set setting). The goal of SFDA is to adapt the model to target domain without source data. We divide the model into two parts: the feature extractor f, and the classifier g. The output of the feature extractor is denoted as feature ($\boldsymbol{z_i} = f(x) \in \mathbb{R}^h$), where h is dimension of the feature space. The output of classifier is denoted as ($p_i = \delta(g(z_i)) \in \mathbb{R}^C$) where δ is the softmax function. We denote $P \in \mathbb{R}^{bs \times C}$ as the prediction matrix in a mini-batch.

Algorithm 1 Local Prediction Aggregation for SFDA

```
Require: Source-pretrained model and target data $\mathcal{D}_t$
1: Build memory bank storing all *target* features and predictions
2: while Adaptation do
3:     Sample batch $\mathcal{T}$ from $\mathcal{D}_t$ and Update memory bank
4:     For each feature $z_i$ in $\mathcal{T}$, retrieve $K$-nearest neighbors ($\mathcal{C}_i$)
       and their predictions from memory bank
5:     Update model by minimizing Eq. 7
6: end while
```

Regarding the SFDA as an unsupervised clustering problem, we address SFDA problem by clustering target features based on the proposed LPA. In additionally, we relate our method with several existing DA, SFDA and contrastive learning methods.

3.1. Local Prediction Aggregation for Source-free Domain Adaptation

Since the source-pretrained model already learns a good feature representation, it can provides a decent initialization for target adaptation. We propose to achieve SFDA by aggregating predictions for features that are located close in feature space, while dispersing predictions of those features farther away in feature space.

We define p_{ij} as the probability that the feature $z_i \in \mathbb{R}^h$ has similar (or the same) prediction to feature z_j:

$$p_{ij} = \frac{e^{p_i^T p_j}}{\sum_{k=1}^{N_t} e^{p_i^t p_k}} \tag{1}$$

This equation can be interpreted as the possibility that p_j is selected as the neighbor of p_i in the output space (Goldberger et al., 2004).

We then define two sets for each feature z_i: close neighbor set $\mathcal{C}_i$ containing K-nearest neighbors of z_i (with distances as cosine similarity), and background set $\mathcal{B}_i$ which contains the features that are not in $\mathcal{C}_i$ (features potentially from different classes). To retrieve nearest neighbors for training, we build two memory banks to store all *target* features along with their predictions just like former works (Liang et al., 2021a; Yang et al., 2021b;a; Saito et al., 2020), which is efficient in both memory and computation, since only the features along with their predictions already computed in each mini-batch are used to update the memory bank.

Intuitively, for each feature z_i, the features in $\mathcal{B}_i$ should have less similar predictions than those in $\mathcal{C}_i$[1]. To achieve this, we first define two likelihood functions:

$$P(\mathcal{C}_i|\theta) = \prod_{j\in\mathcal{C}_i} p_{ij} = \prod_{j\in\mathcal{C}_i} \frac{e^{p_i^T p_j}}{\sum_{k=1}^{N_t} e^{p_i^T p_k}} \tag{2}$$

$$P(\mathcal{B}_i|\theta) = \prod_{j\in\mathcal{B}_i} p_{ij} = \prod_{j\in\mathcal{B}_i} \frac{e^{p_i^T p_j}}{\sum_{k=1}^{N_t} e^{p_i^T p_k}} \tag{3}$$

where θ denotes parameters of the model, for readability we omit θ in following equations. The probability p_j in Eq. 2 is the stored prediction for neighborhood feature z_j, which is retrieved from the memory bank.

We then propose to achieve target features clustering by minimizing the following negative log-likelihood, denoted as *LPA* (**L**ocal **P**rediction **A**ggregation):

$$\tilde{L}_i(\mathcal{C}_i, \mathcal{B}_i) = -\log \frac{P(\mathcal{C}_i)}{P(\mathcal{B}_i)} \tag{4}$$

Noting that, if we only have $P(\mathcal{C}_i)$, it will be similar to Instance Discrimination (Wu et al., 2018), but we also consider $P(\mathcal{B}_i)$ and we operate on predictions instead of features. If regarding weights of the classifier g as classes prototypes, optimizing Eq. 4 is not only pulling features towards their closest neighbors and pushing them away from background features, but also towards (or away from) corresponding class prototypes. Therefore, we can achieve feature clustering and cluster assignment simultaneously.

To simplify the training, instead of manually and carefully sampling background features, we use all other features except z_i in the mini-batch as $\mathcal{B}_i$, which can be regarded as an estimation of the distribution of the whole dataset. We can reasonably believe that overall similarity of features in $\mathcal{C}_i$ is potentially higher than that of $\mathcal{B}_i$, even if $\mathcal{B}_i$ has intersection with $\mathcal{C}_i$ since features in $\mathcal{C}_i$ are the closest ones to feature z_i. By optimizing Eq. 4, we are encouraging features in $\mathcal{C}_i$, which have a higher chance of belonging to the same class, to have more similar predictions to z_i than those features in $\mathcal{B}_i$, which have a lower chance of belonging to the same class. Note all features will show up in both the first and second term; intra-cluster alignment and inter-cluster separability are expected to be achieved after training.

The Eq. 4 is partly inspired by an unsupervised clustering method Local Aggregation (LA) (Zhuang et al., 2019), which also defines two different feature sets and aims to cluster neighbors together while dispersing background samples. However LA defines $\mathcal{B}_i$ as a large amount of nearest neighbors (4096 in their paper), and $\mathcal{C}_i$ as those features that belong to the same cluster after several extra k-means clustering. LA proposes to directly minimize $-\log \frac{P(\mathcal{C}_i \cap \mathcal{B}_i)}{P(\mathcal{B}_i)}$ which is time consuming. While in our method, the definition of $\mathcal{B}_i$ and $\mathcal{C}_i$ is different and easy to implement. And

[1]For better understanding, we refer to $\mathcal{B}_i$ and $\mathcal{C}_i$ as index sets.

unlike LA operating on features our method operates on the prediction as mentioned above, more importantly we do not optimize the original objective which is hard to train in large scale, as we will illustrate below.

One problem optimizing Eq. 4 is that all target data are needed to compute Eq. 2-3, which is infeasible in real-world situation. Here we resort to get an upper-bound of Eq. 4:

$$\begin{aligned}
\tilde{L}_i(\mathcal{C}_i, \mathcal{B}_i) &= -\log \frac{P(\mathcal{C}_i)}{P(\mathcal{B}_i)} \\
&= -\sum_{j \in \mathcal{C}_i} [p_i^T p_j - \log(\sum_{k=1}^{N_t} e^{p_i^T p_k})] \\
&+ \sum_{m \in \mathcal{B}_i} [p_i^T p_m - \log(\sum_{k=1}^{N_t} e^{p_i^T p_k})] \\
&= -\sum_{j \in \mathcal{C}_i} p_i^T p_j + \sum_{m \in \mathcal{B}_i} p_i^T p_m \\
&+ (N_{\mathcal{C}_i} - N_{\mathcal{B}_i}) \log(\sum_{k=1}^{N_t} e^{p_i^T p_k})
\end{aligned} \tag{5}$$

Since we set $N_{\mathcal{C}_i} < N_{\mathcal{B}_i}$, with Jensen's inequality:

$$\begin{aligned}
\tilde{L}_i(\mathcal{C}_i, \mathcal{B}_i) &\leq -\sum_{j \in \mathcal{C}_i} p_i^T p_j + \sum_{m \in \mathcal{B}_i} p_i^T p_m \\
&+ (N_{\mathcal{C}_i} - N_{\mathcal{B}_i})(\sum_{k=1}^{N_t} \frac{1}{N_t} p_i^T p_k + \log N_t) \\
&\simeq \sum_{m \in \mathcal{B}_i} p_i^T p_m - \sum_{j \in \mathcal{C}_i} p_i^T p_j \\
&+ (N_{\mathcal{C}_i} - N_{\mathcal{B}_i})(\sum_{k \in \mathcal{B}_i} \frac{p_i^T p_k}{N_{\mathcal{B}_i}} + \log N_t) \\
&= -\sum_{j \in \mathcal{C}_i} p_i^T p_j + \frac{N_{\mathcal{C}_i}}{N_{\mathcal{B}_i}} \sum_{m \in \mathcal{B}_i} p_i^T p_m \\
&+ (N_{\mathcal{C}_i} - N_{\mathcal{B}_i}) \log N_t
\end{aligned} \tag{6}$$

where $N_{\mathcal{C}_i}$ and $N_{\mathcal{B}_i}$ is the number of features in $\mathcal{C}_i$ and $\mathcal{B}_i$. Note that we cannot get this upper-bound without $P(\mathcal{B}_i)$. The approximation above in the penultimate line is to estimate the average dot product using the mini-batch data. This leads to the *surprisingly simple final objective* for unsupervised domain adaptation:

$$L = \mathbb{E}[L_i(\mathcal{C}_i, \mathcal{B}_i)] \tag{7}$$

$$L_i(\mathcal{C}_i, \mathcal{B}_i) = -\sum_{j \in \mathcal{C}_i} p_i^T p_j + \lambda \sum_{m \in \mathcal{B}_i} p_i^T p_m \tag{8}$$

Note the gradient will come from both p_i and p_m. The first term aims to enforce prediction consistency between local neighbors, and the naive interpretation of second term is to disperse the prediction of potential dissimilar features, which are all other features in the mini-batch. Note that the dot product between two softmaxed predictions will be maximal when two predictions have the same predicted class and are close to one-hot vector. Our algorithm is illustrated in Algorithm. 1.

Table 1. Decomposition of methods into two terms: discriminability (*dis*) and diversity (*div*), which will be minimized for training.

Method	task	*dis* term	*div* term
MI	SFDA&Clustering	$H(Y\|X)$	$-H(Y)$
BNM	DA&SFDA	$-\\|P\\|_F$	$-rank(P)$
NC	SFDA	$-g(W_{ij} p_i^T p_j)$	$\sum_{c=1}^{C} \mathrm{KL}(\bar{p}_c \\|\\| q_c)$
InfoNCE	Contrastive	$-f(x)^T f(y)/\tau$	$\log(\frac{e}{\tau} + \sum_i e^{f(x_i^-)^T f(x)/\tau})$
Ours	SFDA	$-\sum_{j \in \mathcal{C}_i} p_i^T p_j$	$\sum_{m \in \mathcal{B}_i} p_i^T p_m$

Unlike using a constant for the second term in Eq. 6 we empirically found that using a hyperparameter λ to decay second term (starting from 1) works better. One reason may be that the approximation inside Eq. 3.1 is not necessarily accurate. And as training goes on, features are gradually clustering, the role of the second term for dispersing should be weakened. Additionally, considering the current mini-batch with the correctly predicted features z_i and z_m belonging to the same class. In this case the second term in both $L_i(\mathcal{C}_i, \mathcal{B}_i)$ and $L_m(\mathcal{C}_m, \mathcal{B}_m)$ tends to push p_m to the wrong direction, while the first term in $L_m(\mathcal{C}_m, \mathcal{B}_m)$ can potentially keep current (correct) prediction unchanged. Hence, this will suppress the negative impact of the second term. We will further deepen the understanding of these two terms in the next subsection.

3.2. Relation to Existing Works

In this section, we will relate several popular DA, SFDA and contrastive learning methods through two objectives, *discriminability* and *diversity*. This can improve our understanding of domain adaptation methods, as well as improve the understanding of our method.

Mutual Information maximizing (MI). SHOT-IM (Liang et al., 2020a) proposes to achieve source-free domain adaptation by maximizing the mutual information, which is actually widely used in unsupervised clustering (Gomes et al., 2010; Romano et al., 2014; Hu et al., 2017):

$$L_{MI} = H(Y|X) - H(Y) \tag{9}$$

which contains two terms: conditional entropy term $H(Y|X)$ to encourages unambiguous cluster assignments, and marginal entropy term $H(Y)$ to encourage cluster sizes to be uniform to avoid degeneracy. In practice, $H(Y)$ is approximated by the current mini-batch instead of using whole dataset (Springenberg, 2015; Hu et al., 2017).

Batch Nuclear-norm Maximization (BNM). BNM (Cui et al., 2020; 2021) aims to increase prediction discriminability and diversity to tackle domain shift. It is originally achieved by maximizing F-norm (for discriminability) and rank of prediction matrix (for diversity) respectively:

$$L = -\|P\|_F - rank(P) \tag{10}$$

In their paper, they further prove merely maximizing the nuclear norm $\|P\|_*$ can achieve these two goals simultaneously.

In relation to our method, if target features are well clustering during training, we can presume the K-nearest neighbors of feature z_i have the same prediction, the first term in Eq. 7 can be seen as the summation of diagonal elements of matrix PP^T, which is actually the square of F-norm ($\|P\|_F = \sqrt{trace(PP^T)}$), then it is actually minimizing prediction entropy (Cui et al., 2020). As for second term, we can regard it as the summation of non-diagonal element of PP^T, it encourages all these non-diagonal elements to be 0 thus the $rank(PP^T) = rank(P)$ is supposed to increase, which indicates larger prediction diversity (Cui et al., 2020). In a nutshell, compared to SHOT and BNM our method first considers local feature structure to cluster target features, which can be treated as an alternative way to increase discriminability at the late training stage, meanwhile as discussed above our method is also encouraging diversity.

Neighborhood Clustering (NC). G-SFDA (Yang et al., 2021b) and NRC (Yang et al., 2021a) are based on neighborhood clustering to tackle SFDA problem. Those works basically contain two major terms in their optimizing objective: a neighborhood clustering term for prediction consistency and a marginal entropy term $H(Y)$ for prediction diversity. NRC (Yang et al., 2021a) further introduces neighborhood reciprocity to weight the different neighbors. Their loss objective can be written as:

$$L_i = -\sum_{j \in \mathcal{C}_i} g(W_{ij} p_i^T p_j) + \sum_{c=1}^{C} \mathrm{KL}(\bar{p}_c \| q_c), \tag{11}$$
$$\text{with } \bar{p}_c = \frac{1}{n_t}\sum_i p_i^{(c)} \text{ , and } q_{\{c=1,..,C\}} = \frac{1}{C}$$

where W_{ij} will weight the importance of neighbor and $g(\cdot)$ is *log* or *identity* function. Although the first term of G-SFDA and NRC is the same as that of our final loss objective Eq. 7, note that our motivation is different as we simultaneously consider similar and dissimilar features, and Eq. 7 is deduced as an approximated upper-bound of our original objective Eq. 4.

And note actually the marginal entropy term $-H(Y) = \sum_{c=1}^{C} \bar{p}_c \log \bar{p}_c = \sum_{c=1}^{C} \mathrm{KL}(\bar{p}_c \| q_c) - \log C$. Although the second term of those methods are favoring prediction diversity to avoid the trivial solution where all images are only assigned to some certain classes, the margin entropy term presumes the prior that whole dataset or the mini-batch is class balance/uniformly distributed, which is barely true for current benchmarks or in real-world environment. In conclusion, the above three types of methods are actually all to increase discriminability and meanwhile maximize diversity of the prediction, but through different ways.

Contrastive Learning. Here we also link our method to InfoNCE (Oord et al., 2018)), which is widely used in contrastive learning. As a recent paper (Wang & Isola, 2020) points out that InfoNCE loss can be decomposed into 2 terms:

$$\begin{aligned} L_{infoNCE} &= \mathbb{E}_{(x,y)\sim p_{pos}}[-f(x)^T f(y)/\tau] \\ &+ \mathop{\mathbb{E}}_{\substack{x\sim p_{data} \\ \{x_i^-\}_{i=1}^M \sim p_{data}}}[\log(e^{1/\tau} + \sum_i e^{f(x_i^-)^T f(x)/\tau})] \end{aligned} \tag{12}$$

The first term is denoted as *alignment* term (with positive pairs) is to make positive pairs of features similar, and the second term denoted as *uniformity* term with negative pairs encouraging all features to roughly uniformly distributed in the feature space.

The Eq. 12 shares some similarity with all the above domain adaptation methods in that the first term is for the alignment with positive pairs and the second term is to encourage diversity. But note that the remarkable difference is that the *above domain adaptation methods operate in the output (prediction) space while contrastive learning is conducted in the (spherical) feature space.* Therefore, simultaneously feature representation learning and cluster assignment can be achieved for those domain adaptation methods.

Remark. The whole training process can be actually split into 2 parts: training feature extractor and classifier. *Here we regard weights of classifier as prototypes*. Training on the feature extractor encourages similar features to move towards corresponding class prototypes, since we force similar features to have similar prediction. Training on the classifier pushes class prototypes towards corresponding feature clusters. *Through this way we can simultaneously achieve feature clustering and cluster assignment without an extra process. Note in InstanceDiscrimination (Wu et al., 2018) and LA (Zhuang et al., 2019), extra KNN or a linear learnable classifier needs to be deployed for final classification.*

We list all above methods in Tab. 1. Finally, returning to Eq. 7, we can also regard the second term as a variant of diversity loss to avoid degeneration solution, but without making any category prior assumption. Intuitively, with target features forming groups during training, the second term should play less and less important role, otherwise it

Table 2. Accuracies (%) on Office-31 for ResNet50-based methods. We highlight the best result and underline the second best one.

Method	SF	A→D	A→W	D→W	W→D	D→A	W→A	Avg
MCD (Saito et al., 2018)	✗	92.2	88.6	98.5	100.0	69.5	69.7	86.5
CDAN (Long et al., 2018b)	✗	92.9	94.1	98.6	100.0	71.0	69.3	87.7
MDD (Zhang et al., 2019a)	✗	90.4	90.4	98.7	99.9	75.0	73.7	88.0
DMRL (Wu et al., 2020)	✗	93.4	90.8	99.0	100.0	73.0	71.2	87.9
MCC (Jin et al., 2020)	✗	95.6	95.4	98.6	100.0	72.6	73.9	89.4
SRDC (Tang et al., 2020)	✗	95.8	95.7	99.2	100.0	76.7	77.1	<u>90.8</u>
RWOT (Xu et al., 2020)	✗	94.5	95.1	99.5	100.0	77.5	77.9	<u>90.8</u>
LAMDA (Le et al., 2021)	✗	96.0	95.2	98.5	100.0	87.3	84.4	**93.0**
SHOT (Liang et al., 2020a)	✓	94.0	90.1	98.4	99.9	74.7	74.3	88.6
SHOT++ (Liang et al., 2021b)	✓	94.3	90.4	98.7	99.9	76.2	75.8	89.2
3C-GAN (Li et al., 2020)	✓	92.7	93.7	98.5	99.8	75.3	77.8	89.6
NRC (Yang et al., 2021a)	✓	96.0	90.8	99.0	100.0	75.3	75.0	89.4
HCL (Huang et al., 2021a)	✓	94.7	92.5	98.2	100.0	75.9	77.7	89.8
BNM-S (Cui et al., 2021)	✓	93.0	92.9	98.2	99.9	75.4	75.0	89.1
Ours	✓	94.8	93.7	98.4	100.0	75.8	76.8	89.9

may destabilize the training. This is similar to the class collision issue in contrastive learning. If our second term contains too many features belonging to the same class. Thus it is reasonable to decay the second term.

4. Experiments

Datasets. We conduct experiments on three benchmark datasets for image classification: Office-31, Office-Home and VisDA-C 2017. **Office-31** (Saenko et al., 2010) contains 3 domains (Amazon, Webcam, DSLR) with 31 classes and 4,652 images. **Office-Home** (Venkateswara et al., 2017) contains 4 domains (Real, Clipart, Art, Product) with 65 classes and a total of 15,500 images. **VisDA** (VisDA-C 2017) (Peng et al., 2017) is a more challenging dataset, with 12-class synthetic-to-real object recognition tasks, its source domain contains of 152k synthetic images while the target domain has 55k real object images.

Evaluation. The column **SF** in the tables denotes source-free. For Office-31 and Office-Home, we show the results of each task and the average accuracy over all tasks (*Avg* in the tables). For VisDA, we show accuracy for all classes and average over those classes (*Per-class* in the table). All results are the average of three random runs for target adaptation.

Model details. To ensure fair comparison with related methods, we adopt the backbone of a ResNet-50 (He et al., 2016) for Office-Home and ResNet-101 for VisDA. Specifically, we use the same network architecture as SHOT (Liang et al., 2020a), BNM-S (Cui et al., 2021), G-SFDA (Yang et al., 2021b) and NRC (Yang et al., 2021a), *i.e.*, the final part of the network is: *fully connected layer - Batch Normalization (Ioffe & Szegedy, 2015) - fully connected layer with weight normalization (Salimans & Kingma, 2016)*. We adopt SGD with momentum 0.9 and batch size of 64 for all datasets. The learning rate for Office-31 and Office-Home is set to 1e-3 for all layers, except for the last two newly added fc layers, where we apply 1e-2. Learning rates are set 10 times smaller for VisDA. We train 40 epochs for Office-31 and Office-Home while 15 epochs for VisDA.

There are two hyperparameters $N_{\mathcal{C}_i}$ (K which is number of nearest neighbors) and λ, to ensure fair comparison we set $N_{\mathcal{C}_i}$ to the same number as previous works G-SFDA (Yang et al., 2021b) and NRC (Yang et al., 2021a), which also resort to nearest neighbors. That is, we set K to 3 on Office-31 and Office-Home, 5 on VisDA. For λ, we set it as $\lambda = (1 + 10 * \frac{iter}{max\_iter})^{-\beta}$, where the decay factor β controls the decaying speed. β is set to 0 on Office-Home, 1 on Office-31 and 5 on VisDA, we will analyse these hyperparameters in the following section.

4.1. Results and Analysis

Quantitative Results. As shown in Tables 2-4, where the top part shows results for the source-present methods that use source data during adaptation, and the bottom part shows results for the source-free DA methods. On Office-31 and VisDA, our method gets state-of-the-art performance compared to existing source-free domain adaptation methods, especially on VisDA our method outperforms others by a large margin (2.1% compared to NRC). And our method achieves similar results on Office-Home compared to the more complex A^2Net method (which combines three classifiers and five objective functions). The reported results clearly demonstrate the efficiency of the proposed method for source-free domain adaptation. It also achieves similar or better results compared to domain adaptation methods with access to source data on both Office-Home and VisDA. SHOT++ (Liang et al., 2021b) deploys extra self-supervised training and semi-supervised learning, which are general to improve the results (*an evidence is that the source model after these 2 tricks gets huge improvement, e.g., 60.2% improves to 66.6% on Office-Home.*), however, our method is still higher than SHOT++ on VisDA (88.0% versus 87.3%) even when starting from the inferior source model.

Toy dataset. We carry out an experiment on the twinning moona dataset to ablate the influence of two terms in our objective Eq. 7. For the twinning moons dataset, the data from the source domain are represented by two inter-twinning moons, which contain 300 samples each. Data in the target domain are generated through rotating source data by 30°. The domain shift here is instantiated as the rotation degree. First we train the model only on the source domain, and test the model on all domains. As shown in the first image in Fig. 1, the source model performs badly on target data. Then we conduct several variants of our method to train the model. *Note here we treat these 2D samples as (fixed) features and only train a classifier* which consists of three fully

Table 3. Accuracies (%) on Office-Home for ResNet50-based methods. We highlight the best result and underline the second best one.

Method	SF	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	**Avg**
ResNet-50 (He et al., 2016)	✗	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
MCD (Saito et al., 2018)	✗	48.9	68.3	74.6	61.3	67.6	68.8	57.0	47.1	75.1	69.1	52.2	79.6	64.1
CDAN (Long et al., 2018b)	✗	50.7	70.6	76.0	57.6	70.0	70.0	57.4	50.9	77.3	70.9	56.7	81.6	65.8
SAFN (Xu et al., 2019)	✗	52.0	71.7	76.3	64.2	69.9	71.9	63.7	51.4	77.1	70.9	57.1	81.5	67.3
MDD (Zhang et al., 2019a)	✗	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
TADA (Wang et al., 2019)	✗	53.1	72.3	77.2	59.1	71.2	72.1	59.7	53.1	78.4	72.4	60.0	82.9	67.6
SRDC (Tang et al., 2020)	✗	52.3	76.3	81.0	69.5	76.2	78.0	68.7	53.8	81.7	76.3	57.1	85.0	71.3
LAMDA (Le et al., 2021)	✗	57.2	78.4	82.6	66.1	80.2	81.2	65.6	55.1	82.8	71.6	59.2	83.9	72.0
SHOT (Liang et al., 2020a)	✓	57.1	78.1	81.5	68.0	78.2	78.1	67.4	54.9	82.2	73.3	58.8	84.3	71.8
SHOT++ (Liang et al., 2021b)	✓	57.9	79.7	82.5	68.5	79.6	79.3	68.5	57.0	83.0	73.7	60.7	84.9	**73.0**
A^2Net (Xia et al., 2021)	✓	58.4	79.0	82.4	67.5	79.3	78.9	68.0	56.2	82.9	74.1	60.5	85.0	<u>72.8</u>
G-SFDA (Yang et al., 2021b)	✓	57.9	78.6	81.0	66.7	77.2	77.2	65.6	56.0	82.2	72.0	57.8	83.4	71.3
NRC (Yang et al., 2021a)	✓	57.7	80.3	82.0	68.1	79.8	78.6	65.3	56.4	83.0	71.0	58.6	85.6	72.2
BNM-S (Cui et al., 2021)	✓	57.4	77.8	81.7	67.8	77.6	79.3	67.6	55.7	82.2	73.5	59.5	84.7	72.1
Ours	✓	59.3	79.3	82.1	68.9	79.8	79.5	67.2	57.4	83.1	72.1	58.5	85.4	72.7

Table 4. Accuracies (%) on VisDA-C (Synthesis → Real) for ResNet101-based methods. We highlight the best result and underline the second best one.

Method	SF	plane	bcycl	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	truck	**Per-class**
ResNet-101 (He et al., 2016)	✗	55.1	53.3	61.9	59.1	80.6	17.9	79.7	31.2	81.0	26.5	73.5	8.5	52.4
CDAN+BSP (Chen et al., 2019)	✗	92.4	61.0	81.0	57.5	89.0	80.6	90.1	77.0	84.2	77.9	82.1	38.4	75.9
SAFN (Xu et al., 2019)	✗	93.6	61.3	84.1	70.6	94.1	79.0	91.8	79.6	89.9	55.6	89.0	24.4	76.1
SWD (Lee et al., 2019)	✗	90.8	82.5	81.7	70.5	91.7	69.5	86.3	77.5	87.4	63.6	85.6	29.2	76.4
MCC (Jin et al., 2020)	✗	88.7	80.3	80.5	71.5	90.1	93.2	85.0	71.6	89.4	73.8	85.0	36.9	78.8
STAR (Lu et al., 2020)	✗	95.0	84.0	84.6	73.0	91.6	91.8	85.9	78.4	94.4	84.7	87.0	42.2	82.7
RWOT (Xu et al., 2020)	✗	95.1	80.3	83.7	90.0	92.4	68.0	92.5	82.2	87.9	78.4	90.4	68.2	84.0
3C-GAN (Li et al., 2020)	✓	94.8	73.4	68.8	74.8	93.1	95.4	88.6	84.7	89.1	84.7	83.5	48.1	81.6
SHOT (Liang et al., 2020a)	✓	94.3	88.5	80.1	57.3	93.1	94.9	80.7	80.3	91.5	89.1	86.3	58.2	82.9
SHOT++ (Liang et al., 2021b)	✓	97.7	88.4	90.2	86.3	97.9	98.6	92.9	84.1	97.1	92.2	93.6	28.8	<u>87.3</u>
A^2Net (Xia et al., 2021)	✓	94.0	87.8	85.6	66.8	93.7	95.1	85.8	81.2	91.6	88.2	86.5	56.0	84.3
G-SFDA (Yang et al., 2021b)	✓	96.1	88.3	85.5	74.1	97.1	95.4	89.5	79.4	95.4	92.9	89.1	42.6	85.4
NRC (Yang et al., 2021a)	✓	96.8	91.3	82.4	62.4	96.2	95.9	86.1	80.6	94.8	94.1	90.4	59.7	85.9
HCL (Huang et al., 2021a)	✓	93.3	85.4	80.7	68.5	91.0	88.1	86.0	78.6	86.6	88.8	80.0	74.7	83.5
Ours	✓	97.4	90.5	80.8	76.2	97.3	96.1	89.8	82.9	95.5	93.0	92.0	64.7	**88.0**

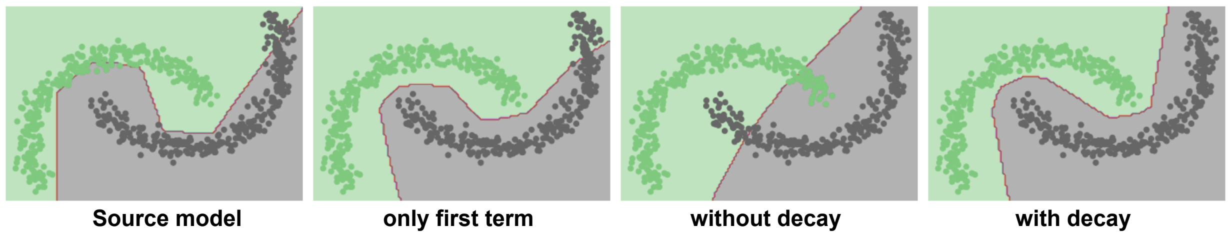

Figure 1. Visualization of decision boundary on target data with different training objective. *The twin moons dataset is balanced.*

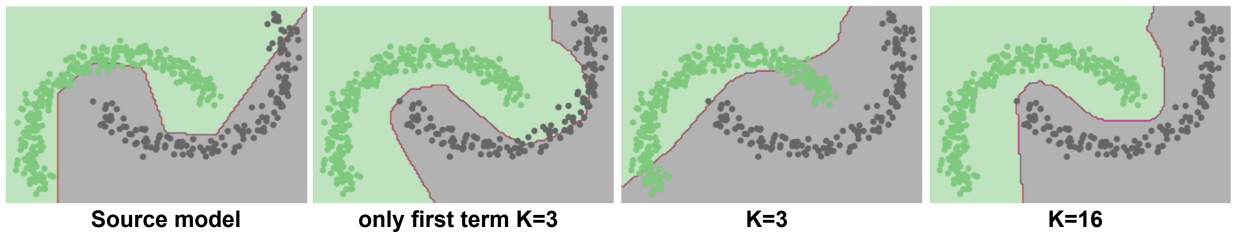

Figure 2. Visualization of decision boundary on target data with different training objective and K, *The twin moons dataset is imbalanced with class ratio 2:1.*

Table 5. Ablation study on number of nearest neighbors K ($N_{\mathcal{C}_i}$) and decay coefficient β, where *bs* means batch-size (64 on all datasets) and *nc* means number of classes. We highlight the best score and underline the second best one for each dataset.

bs/C	K	β	**Avg**
		Office-31	
2.1	1	1	89.1
2.1	2	1	89.5
2.1	2	2	89.5
2.1	3	0.25	89.1
2.1	3	1	**89.9**
2.1	3	2	$\underline{89.8}$
		Office-Home	
1	1	0	72.2
1	1	0.25	72.2
1	1	1	71.6
1	2	0	$\underline{72.6}$
1	2	0.25	**72.7**
1	2	1	71.8
1	3	0	**72.7**
1	3	0.25	$\underline{72.6}$

bs/C	K	β	**Per-class**
		VisDA	
5.3	3	5	86.7
5.3	4	5	87.4
5.3	5	5	$\underline{88.0}$
5.3	6	5	$\underline{88.0}$
5.3	7	5	$\underline{88.0}$
5.3	5	0	77.5
5.3	5	1	83.8
5.3	5	2	86.7
5.3	5	3	87.6
5.3	5	4	$\underline{88.0}$
5.3	5	6	**88.1**
5.3	5	7	**88.1**
5.3	6	4	$\underline{88.0}$
5.3	6	5	$\underline{88.0}$

connected layers, with batch-size set to 32, $N_{\mathcal{C}_i}/K$ is set to 3 and β as 2. The visualization of the decision boundary in Fig. 1 indicates that both terms in Eq. 7 are necessary, and decay of second term is also shown to be important.

Number of nearest neighbors ($N_{\mathcal{C}_i}$ or K). Here we discuss the choice of $N_{\mathcal{C}_i}/K$, the number of retrieved nearest neighbors for $\mathcal{C}_i$. We posit that the element amount of the first term in Eq. 7 ideally should be larger than the number of features belonging to the same class in the second term. Since we cannot know the class distribution in the mini-batch, we estimate that there could be bs/C images per class (where bs is the batch size). To avoid that the supervision signal from the first term gets neutralized by the second term, we posit that it is better that $N_{\mathcal{C}_i} \geq bs/C$.

Although in the previous experiment on the twinning moons dataset, it also works when $N_{\mathcal{C}_i}$ is set to 3, which is much smaller than $bs/C = 16$. We conduct another experiment on a new twinning moons dataset which is imbalanced (300 and 150 samples per class), since existing benchmarks and data in the real-world are barely class balanced. As shown in Fig. 2, adopting a small $N_{\mathcal{C}_i} = 3$ cannot perform well. We conjecture that the reason may be that for features from the majority category (green) there are more samples in the current mini-batch ($\mathcal{B}_i$) belonging to the same class, leading to a too strong supervision from the second term and poor clustering. Simply increasing $N_{\mathcal{C}_i}$ to $bs/C = 16$ can lead to good performance. Although we cannot know the real category distribution, we empirically find that setting $N_{\mathcal{C}_i}$ to bs/C (or larger) achieves good performance.

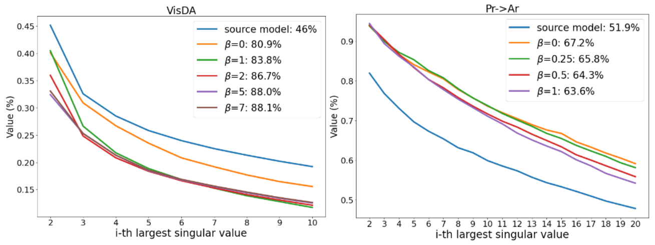

Figure 3. 10/20 largest singular values (max-normalized, the first one is 1) of features of VisDA (**Left**) and Pr→Ar on Office-Home (**Right**).

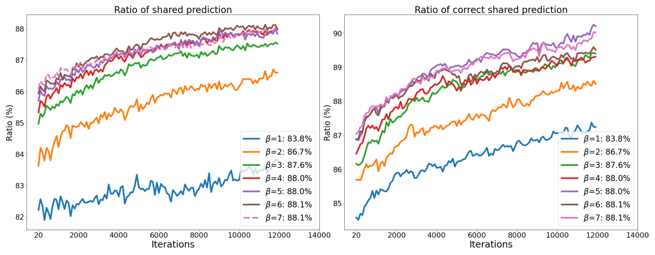

Figure 4. (**Left**) Ratio of features which have 3 nearest neighbor features sharing the same predicted label. (**Right**) Ratio among **above features** which have 3 nearest neighbor features sharing the same and **correct** predicted label.

Decaying factor β. According to the analysis in Sec. 3.2, the second term acts like a diversity term to avoid that all target features collapse to a limited set of categories. The role of the second term should be weakened during the training, but how to decay the second term is non-trivial. We conduct the ablation study for the decay factor β along with $N_{\mathcal{C}_i}$ in Tab. 5. Note that with larger β the second term decays faster. From the table we can roughly draw the conclusion that β is also related to bs/C. With larger bs/C, the second term will have more chance to contain features in the same class, thus the second term should decay quickly.

Transferability and Discriminability. A recent work (Chen et al., 2019) adopts singular value decomposition to analyze the spectral properties of feature representations for domain adaptation. It proves that a sharp distribution of singular values intuitively implies deteriorated discriminability, since the eigenvectors corresponding to smaller singular values are related to discriminability, while the eigenvectors with the top singular values dominate the transferability. We conduct this analysis trying to understand how different decaying terms β influence these different properties. As shown in Fig. 3 where we plot singular values vectors with different β, it shows that on both datasets a larger decaying factor will lead to a sharp distribution of singular values which means favouring transferability. However, when it comes

Table 6. Accuracy(%) on Office-Home using ResNet-50 as backbone for **open-set DA**. $|\mathcal{C}_s| = 25$, $|\mathcal{C}_t| = 65$, $|\mathcal{C}_s| \cap |\mathcal{C}_t| = 25$. ***OS**** means average per-class accuracy across known classes, ***UNK*** means unknown accuracy and ***HOS*** means harmonic mean between known and unknown accuracy. **All results are picked from the last iteration**.

	Ar → Cl			Ar → Pr			Ar → Rw			Cl→ Ar			Cl → Pr			Cl → Rw					
	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS			
SHOT	67.0	28.0	39.5	81.8	26.3	39.8	87.5	32.1	47.0	66.8	46.2	54.6	77.5	27.2	40.2	80.0	25.9	39.1			
LPA	50.7	66.4	**57.6**	64.6	69.4	**66.9**	73.1	66.9	**69.9**	48.2	81.1	**60.5**	59.5	63.5	**61.4**	67.4	68.3	**67.8**			
	Pr → Ar			Pr → Cl			Pr → Rw			Rw→ Ar			Rw → Cl			Rw → Pr			**Avg.**		
	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS
SHOT	66.3	51.1	57.7	59.3	31.0	40.8	85.8	31.6	46.2	73.5	50.6	59.9	65.3	28.9	40.1	84.4	28.2	42.3	74.6	33.9	45.6
LPA	47.3	82.4	**60.1**	45.4	72.8	**55.9**	68.4	72.8	**70.6**	54.5	79.0	**64.6**	49.0	69.6	**57.5**	69.7	70.6	**70.1**	58.2	71.9	**63.6**

Table 7. Accuracy(%) on Office-Home using ResNet-50 as backbone under **partial-set DA**. $|\mathcal{C}_s| = 65$, $|\mathcal{C}_t| = 25$, $|\mathcal{C}_s| \cap |\mathcal{C}_t| = 25$.

Partial-set DA	Ar→Cl	Ar→Pr	Ar→Re	Cl→Ar	Cl→Pr	Cl→Re	Pr→Ar	Pr→Cl	Pr→Re	Re→Ar	Re→Cl	Re→Pr	**Avg.**
SHOT-IM	57.9	83.6	88.8	72.4	74.0	79.0	76.1	60.6	90.1	81.9	**68.3**	**88.5**	76.8
SHOT	64.8	**85.2**	92.7	76.3	**77.6**	**88.8**	**79.7**	64.3	89.5	80.6	66.4	85.8	79.3
LPA	**67.0**	83.5	**93.1**	**80.5**	76.0	87.6	78.1	**65.6**	**90.2**	**83.5**	64.3	87.3	**79.7**

to performance, conclusions are totally on the contrary as the sharp distribution for VisDA gets better results while the flat one for Office-Home gets better results. Note that also the curves of the source model are located on different sites of the target curves when comparing the two datasets. The results indicate that for different datasets the priority of discriminability or transferability may be different: discriminability dominates for Office-Home and transferability for VisDA. *And the optimal β is empirically found to be with the singular values curve which is the farthest from the source model.*

Degree of clustering during training. We also plot how features are clustered with different decaying factors β on VisDA in Fig. 4. The left one shows the ratio of features which have 3-nearest neighbors all sharing the same prediction, which indicates the degree of clustering during training, and the right one shows the ratio among above features which have 3-nearest neighbor features sharing the same and *correct* predicted label. Those curves in Fig. 4 *left* show that the target features are clustering, and those in Fig. 4 *right* indicate that clear category boundaries are emerging. The numbers in the legends denote the deployed β and the corresponding final accuracy. From the figures we can draw the conclusion that with a larger decay factor β on VisDA, features are quickly clustering and forming inter-class boundaries, since the ratio of features which share the same and correct prediction with neighbors are increasing faster. When decaying factor β is too small, meaning training signal from the second term is strong, the clustering process is actually impeded.

5. Conclusion

We proposed to tackle source-free domain adaptation by encouraging similar features in feature space to have similar predictions while dispersing predictions of dissimilar features in feature space, to achieve simultaneously feature clustering and cluster assignment. We introduced an upper bound to our proposed objective, resulting in two simple terms. Further we showed that we can unify several popular domain adaptation, source-free domain adaptation and contrastive learning methods from the perspective of discriminability and diversity. The approach is simple but achieves state-of-the-art performance on several benchmarks.

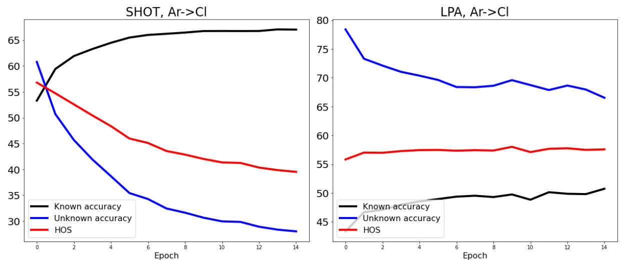

Figure 5. Training curves on Ar→Cl task (Office-Home) of SHOT (left) and LPA (Right) under ODA setting.

Acknowledgement

We acknowledge the support from Huawei Kirin Solution, and the project PID2019-104174GB-I00 (MINECO, Spain), and the CERCA Programme of Generalitat de Catalunya.

Appendix

Partial and open set DA. We provide additional results under ODA and PDA setting in Tab. 6 and Tab. 3 (*Left*) respectively, where the open-set detection in ODA follows the same protocol as SHOT. For the hyperparameter β, it works well for ODA with set to 1, and it can be set to 2 3 to achieve good performance for PDA. On ODA, instead of reporting average *per-class* accuracy $OS = \frac{|\mathcal{C}_s| \times OS^*}{|\mathcal{C}_s|+1} + \frac{1 \times UNK}{|\mathcal{C}_s|+1}$ where $|\mathcal{C}_s|$ is the number of known categories on source domain, we report results of $HOS = \frac{2 \times OS^* \times UNK}{OS^* + UNK}$, which is *harmonic mean* between known categories accuracy OS^* and unknown accuracy *UNK*. As pointed out by (Bucci et al., 2020), OS is problematic since this metric can be quite high even when unknown class accuracy *UNK* is 0, while unknown category detection is the key part in open-set DA. We reproduce SHOT under open-set DA and report results of OS^*, *UNK* and *HOS* in Tab. 6, which shows our method gets much better balance between known and unknown accuracy. In Fig. 5, we plot the training curve of these accuracy metrics for SHOT and our LPA, it shows the unknown accuracy of SHOT is degrading significantly during training resulting in a lower *HOS* metric.

References

Bucci, S., Loghmani, M. R., and Tommasi, T. On the effectiveness of image rotation for open set domain adaptation. In *European Conference on Computer Vision*, pp. 422–438. Springer, 2020.

Chang, J., Wang, L., Meng, G., Xiang, S., and Pan, C. Deep adaptive image clustering. In *ICCV*, pp. 5879–5887, 2017.

Chen, X., Wang, S., Long, M., and Wang, J. Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation. In *ICML*, pp. 1081–1090, 2019.

Cicek, S. and Soatto, S. Unsupervised domain adaptation via regularized conditional alignment. In *ICCV*, pp. 1416–1425, 2019.

Cui, S., Wang, S., Zhuo, J., Li, L., Huang, Q., and Tian, Q. Towards discriminability and diversity: Batch nuclear-norm maximization under label insufficient situations. *CVPR*, 2020.

Cui, S., Wang, S., Zhuo, J., Li, L., Huang, Q., and Tian, Q. Fast batch nuclear-norm maximization and minimization for robust domain adaptation. *arXiv preprint arXiv:2107.06154*, 2021.

Deng, Z., Luo, Y., and Zhu, J. Cluster alignment with a teacher for unsupervised domain adaptation. In *ICCV*, pp. 9944–9953, 2019.

Dwibedi, D., Aytar, Y., Tompson, J., Sermanet, P., and Zisserman, A. With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. *ICCV*, 2021.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. Domain-adversarial training of neural networks. *JMLR*, 17(1):2096–2030, 2016.

Goldberger, J., Hinton, G. E., Roweis, S., and Salakhutdinov, R. R. Neighbourhood components analysis. *NIPS*, 17, 2004.

Gomes, R., Krause, A., and Perona, P. Discriminative clustering by regularized information maximization. In *NIPS*, 2010.

Gong, B., Shi, Y., Sha, F., and Grauman, K. Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*, pp. 2066–2073. IEEE, 2012.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *CVPR*, pp. 770–778, 2016.

Hu, W., Miyato, T., Tokui, S., Matsumoto, E., and Sugiyama, M. Learning discrete representations via information maximizing self-augmented training. In *ICML*, pp. 1558–1567, 2017.

Huang, J., Guan, D., Xiao, A., and Lu, S. Model adaptation: Historical contrastive learning for unsupervised domain adaptation without source data. *NeurIPS*, 34, 2021a.

Huang, Z., Chen, J., Zhang, J., and Shan, H. Exploring non-contrastive representation learning for deep clustering. *arXiv preprint arXiv:2111.11821*, 2021b.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

Ji, X., Henriques, J. F., and Vedaldi, A. Invariant information clustering for unsupervised image classification and segmentation. In *ICCV*, pp. 9865–9874, 2019.

Jin, Y., Wang, X., Long, M., and Wang, J. Minimum class confusion for versatile domain adaptation. *ECCV*, 2020.

Kundu, J. N., Venkat, N., and Babu, R. V. Universal source-free domain adaptation. *CVPR*, 2020a.

Kundu, J. N., Venkat, N., Revanur, A., Babu, R. V., et al. Towards inheritable models for open-set domain adaptation. In *CVPR*, pp. 12376–12385, 2020b.

Le, T., Nguyen, T., Ho, N., Bui, H., and Phung, D. Lamda: Label matching deep domain adaptation. In *ICML*, pp. 6043–6054. PMLR, 2021.

Lee, C.-Y., Batra, T., Baig, M. H., and Ulbricht, D. Sliced wasserstein discrepancy for unsupervised domain adaptation. In *CVPR*, pp. 10285–10295, 2019.

Li, J., Zhou, P., Xiong, C., and Hoi, S. Prototypical contrastive learning of unsupervised representations. In *ICLR*, 2021a.

Li, R., Jiao, Q., Cao, W., Wong, H.-S., and Wu, S. Model adaptation: Unsupervised domain adaptation without source data. In *CVPR*, pp. 9641–9650, 2020.

Li, Y., Hu, P., Liu, Z., Peng, D., Zhou, J. T., and Peng, X. Contrastive clustering. In *AAAI*, 2021b.

Liang, J., Hu, D., and Feng, J. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. *ICML*, 2020a.

Liang, J., Hu, D., Wang, Y., He, R., and Feng, J. Source data-absent unsupervised domain adaptation through hypothesis transfer and labeling transfer. *arXiv preprint arXiv:2012.07297*, 2020b.

Liang, J., Hu, D., and Feng, J. Domain adaptation with auxiliary target domain-oriented classifier. In *CVPR*, pp. 16632–16642, 2021a.

Liang, J., Hu, D., Wang, Y., He, R., and Feng, J. Source data-absent unsupervised domain adaptation through hypothesis transfer and labeling transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021b.

Liu, H., Wang, J., and Long, M. Cycle self-training for domain adaptation. In *NeurIPS*, 2021.

Long, M., Cao, Y., Wang, J., and Jordan, M. I. Learning transferable features with deep adaptation networks. *ICML*, 2015.

Long, M., Zhu, H., Wang, J., and Jordan, M. I. Unsupervised domain adaptation with residual transfer networks. In *NIPS*, pp. 136–144, 2016.

Long, M., Cao, Y., Cao, Z., Wang, J., and Jordan, M. I. Transferable representation learning with deep adaptation networks. *TPAMI*, 41(12):3071–3085, 2018a.

Long, M., Cao, Z., Wang, J., and Jordan, M. I. Conditional adversarial domain adaptation. In *NIPS*, pp. 1647–1657, 2018b.

Lu, Z., Yang, Y., Zhu, X., Liu, C., Song, Y.-Z., and Xiang, T. Stochastic classifiers for unsupervised domain adaptation. In *CVPR*, pp. 9111–9120, 2020.

Oord, A. v. d., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Pan, S. J. and Yang, Q. A survey on transfer learning. *TKDE*, 22(10):1345–1359, 2009.

Peng, X., Usman, B., Kaushik, N., Hoffman, J., Wang, D., and Saenko, K. Visda: The visual domain adaptation challenge. *arXiv preprint arXiv:1710.06924*, 2017.

Romano, S., Bailey, J., Nguyen, V., and Verspoor, K. Standardized mutual information for clustering comparisons: one step further in adjustment for chance. In *ICML*, pp. 1143–1151, 2014.

Saenko, K., Kulis, B., Fritz, M., and Darrell, T. Adapting visual category models to new domains. In *ECCV*, pp. 213–226. Springer, 2010.

Saito, K., Watanabe, K., Ushiku, Y., and Harada, T. Maximum classifier discrepancy for unsupervised domain adaptation. In *CVPR*, pp. 3723–3732, 2018.

Saito, K., Kim, D., Sclaroff, S., and Saenko, K. Universal domain adaptation through self supervision. *NeurIPS*, 33, 2020.

Salimans, T. and Kingma, D. P. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. *arXiv preprint arXiv:1602.07868*, 2016.

Shen, Y., Shen, Z., Wang, M., Qin, J., Torr, P. H., and Shao, L. You never cluster alone. In *NeurIPS*, 2021.

Shu, R., Bui, H. H., Narui, H., and Ermon, S. A dirt-t approach to unsupervised domain adaptation. *ICLR*, 2018.

Springenberg, J. T. Unsupervised and semi-supervised learning with categorical generative adversarial networks. In *ICLR*, 2015.

Sun, B., Feng, J., and Saenko, K. Return of frustratingly easy domain adaptation. In *AAAI*, 2016.

Tang, H., Chen, K., and Jia, K. Unsupervised domain adaptation via structurally regularized deep clustering. In *CVPR*, pp. 8725–8735, 2020.

Tsai, T. W., Li, C., and Zhu, J. Mice: Mixture of contrastive experts for unsupervised image clustering. In *ICLR*, 2021.

Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., and Darrell, T. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.

Tzeng, E., Hoffman, J., Saenko, K., and Darrell, T. Adversarial discriminative domain adaptation. In *CVPR*, pp. 7167–7176, 2017.

Venkateswara, H., Eusebio, J., Chakraborty, S., and Panchanathan, S. Deep hashing network for unsupervised domain adaptation. In *CVPR*, pp. 5018–5027, 2017.

Wang, T. and Isola, P. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *ICML*, pp. 9929–9939. PMLR, 2020.

Wang, X., Li, L., Ye, W., Long, M., and Wang, J. Transferable attention for domain adaptation. In *AAAI*, volume 33, pp. 5345–5352, 2019.

Wu, J., Long, K., Wang, F., Qian, C., Li, C., Lin, Z., and Zha, H. Deep comprehensive correlation mining for image clustering. In *ICCV*, pp. 8150–8159, 2019.

Wu, Y., Inkpen, D., and El-Roby, A. Dual mixup regularized learning for adversarial domain adaptation. *ECCV*, 2020.

Wu, Z., Xiong, Y., Yu, S. X., and Lin, D. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, pp. 3733–3742, 2018.

Xia, H., Zhao, H., and Ding, Z. Adaptive adversarial network for source-free domain adaptation. In *ICCV*, pp. 9010–9019, 2021.

Xu, R., Li, G., Yang, J., and Lin, L. Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation. In *ICCV*, October 2019.

Xu, R., Liu, P., Wang, L., Chen, C., and Wang, J. Reliable weighted optimal transport for unsupervised domain adaptation. In *CVPR*, pp. 4394–4403, 2020.

Yang, S., van de Weijer, J., Herranz, L., Jui, S., et al. Exploiting the intrinsic neighborhood structure for source-free domain adaptation. *NeurIPS*, 34, 2021a.

Yang, S., Wang, Y., van de Weijer, J., Herranz, L., and Jui, S. Generalized source-free domain adaptation. In *ICCV*, pp. 8978–8987, 2021b.

Zhang, Y., Liu, T., Long, M., and Jordan, M. Bridging theory and algorithm for domain adaptation. In *ICML*, pp. 7404–7413, 2019a.

Zhang, Y., Tang, H., Jia, K., and Tan, M. Domain-symmetric networks for adversarial domain adaptation. In *CVPR*, pp. 5031–5040, 2019b.

Zhang, Y., Deng, B., Jia, K., and Zhang, L. Label propagation with augmented anchors: A simple semi-supervised learning baseline for unsupervised domain adaptation. In *ECCV*, pp. 781–797, 2020.

Zhuang, C., Zhai, A. L., and Yamins, D. Local aggregation for unsupervised learning of visual embeddings. In *ICCV*, pp. 6002–6012, 2019.