MIRROR SAMPLE BASED DISTRIBUTION ALIGNMENT FOR UNSUPERVISED DOMAIN ADAPTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Unsupervised Domain Adaption has great value in both machine learning theory and applications. The core issue is how to minimize the domain shift. Motivated by the more and more sophisticated distribution alignment methods in sample level, we introduce a novel concept named (virtual) mirror, which represents the counterpart sample in the other domains. The newly-introduced mirror loss using the virtual mirrors establishes the connection cross domains and pushes the virtual mirror pairs together in the aligned representation space. Our proposed method does not align the samples cross domains coarsely or arbitrarily, thus does not distort the internal distribution of the underline distribution and brings better asymptotic performances. Experiments on several benchmarks validate the superior performance of our methods.

1 INTRODUCTION

Transductive Unsupervised Domain Adaption (UDA) attracts a lot of attentions in recent years. In UDA settings, one has the source domain data with labels and is expected to predict for the unlabeled target domain data under the same task (image classification or segmentation). Those transferable domains might be the images from different scenes, such as daylight and dark, or artclips and real-world photos (Saenko et al. (2010)), or the different types of coherent corpus in natural language, such as news and social media (Ramponi & Plank (2020)).

The main stream solutions are tackling with the core issue: domain shift, especially the covariate shift (Shimodaira (2000); Shi & Sha (2012)). Specifically, define $h(x)$ as the function to be learned, we expect to minimize the target domain risk only through the source domain data:

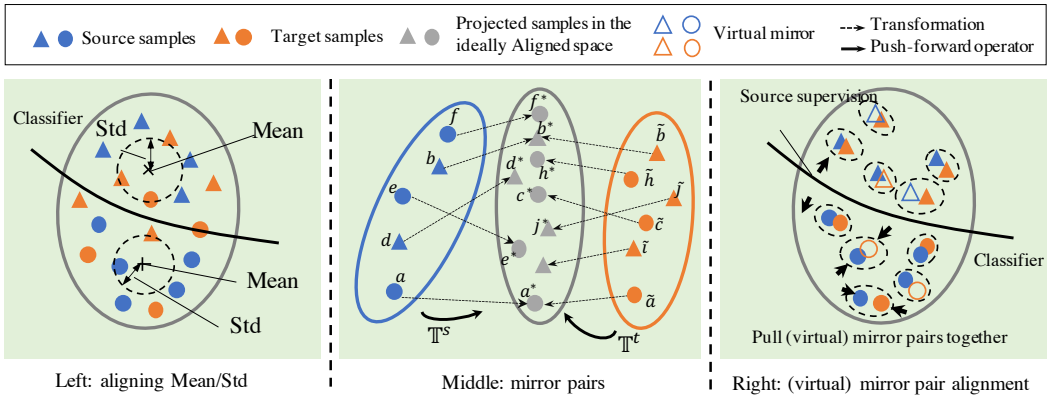
$$\mathcal{R}_{\mathcal{T}}(h) = \sum_{y\in\mathcal{Y}} \int l(h(x), y) p_{\mathcal{T}}(x, y) dx = \sum_{y\in\mathcal{Y}} \int l(h(x), y) \frac{p_{\mathcal{T}}(x, y)}{p_{\mathcal{S}}(x, y)} p_{\mathcal{S}}(x, y) dx \tag{1}$$

where $l : \mathbb{R} \times \mathcal{Y} \to \mathbb{R}$ is the risk function, $\mathcal{S}$ and $\mathcal{T}$ refer to the source and target domains and $\mathcal{Y}$ is the common label space. Imagine that if the distributions of source and target domains are consistent, i.e. $p_{\mathcal{S}}(x, y) \overset{a.s}{=} p_{\mathcal{T}}(x, y)$, Eq.(1) can be further simplified as:

$$\mathcal{R}_{\mathcal{T}}(h) = \sum_{y\in Y} \int l(h(x), y) p_{\mathcal{S}}(x, y) dx = R_{\mathcal{S}}(h) \tag{2}$$

meaning that minimizing the target risk is equivalently to minimizing the source domain risk. Note that $p_{\mathcal{S}}(x, y) = p_{\mathcal{S}}(y|x)p_{\mathcal{S}}(x)$, $p_{\mathcal{T}}(x, y) = p_{\mathcal{T}}(y|x)p_{\mathcal{T}}(x)$, where $p_{\mathcal{S}}(y|x)$ and $p_{\mathcal{T}}(y|x)$ are the labeling functions (Ben-David et al. (2010)), which cannot be manipulated (we do not consider the label shift issue). Thus eliminating the covariate shift, i.e. aligning $p_{\mathcal{T}}(x)$ and $p_{\mathcal{S}}(x)$ in certain feature space of x, can reduce the gap between the risks.

Following this line, current solutions leverage varieties of networks to extract domain-invariant features in different representation spaces under different alignment or discrepancy metrics. Those networks include one classifiers (Long et al. (2015); Chen et al. (2020)), task-specific classifier (i.e. 2 classifiers)(Long et al. (2016); Saito et al. (2018)), adversarial learning (Ganin et al. (2016); Xu et al. (2019a); Tzeng et al. (2017); Ganin et al. (2016)) and even multiple stochastic classifiers (Lu et al. (2020)). The alignment space might be the raw embedding space (Chen et al. (2019b)), reproducing kernel Hilbert space(Long et al. (2015)), normalized space(Xu et al. (2019b)) and sphere



Left: aligning Mean/Std Middle: mirror pairs Right: (virtual) mirror pair alignment

Figure 1: (Best viewed in color) Left: traditional class-aware method aligns the statistics (moment estimations) of the distribution but cannot align the "shape" of distribution. Middle: mirror pairs is the same instance in the ideally aligned common space by distribution transformation map (push-forward operator), where $(a, \tilde{a})$, $(b, \tilde{b})$ are mirror pairs since they have the same corresponding instances a^* ,b^* in the aligned space and the remaining samples do not have mirrors in the opposite domain sample sets. Right: our method will pull the (virtual) mirror instances together, where the virtual mirror instances are estimated using manifold properties of the learned feature space. Details can be found in Section 3.2.

space(Gu et al. (2020)). The discrepancy measures are the crucial part in the model designs, which range from domain-level discrepancy (Long et al. (2015),Long et al. (2016)), class-wise discrepancy(Kang et al. (2019)) and element-wise discrepancy(Chen et al. (2019b)). Typical measures include the moment estimations of distribution(Long et al. (2015); Chen et al. (2020); Borgwardt et al. (2006); Sun & Saenko (2016)), centers or prototype of each class(Pan et al. (2019)), margins of inter-intra class distances(Chen et al. (2019a); Deng et al. (2019); Kang et al. (2019)) and even the more sophisticated ones such as d-SNE (Xu et al. (2019c)), Wasserstein Distance (Lee et al. (2019)) and Margin Disparity Discrepancy(Zhang et al. (2019b)), etc. However, the statistics of distribution or other manipulated measures, more or less suffer from the loss of information for the underline distribution and distort the internal structure of the original distribution. The left part of Fig. 1 shows an example, where even the mean and standard deviation (Std) of each class are same for different domains, there are still many target outliers with respect to the classifier trained on the source data.

In this paper, we propose a novel method by aligning the distribution in the learned representation space by designing a sophisticated domain constraint in sample level. Specifically, we introduce new concepts: mirror and virtual mirror for each sample in both domains (see the middle and right of Fig. 1 and Section 3.2). The mirror of one instance is the one in the other domain having the same position in the aligned space. By constructing the constraints through the mirror pairs, i.e. the newly-proposed mirror loss in Section 3.2, the learned representation is expected to be aligned in a more fine-grained way without breaking the internal structure of the distribution. Our method takes full advantages of sample-level information and is expected to have superior asymptotic performances shown in Section 4.

2 Related Works

A large amount of domain adaption methods are committed to reduce the distribution discrepancy between source and target domains, from coarse to fine. The early work originates from the framework for analyzing and comparing distribution named maximum mean discrepancy (MMD, Gretton et al. (2012)), which refers to the largest difference in expectations over functions in the unit ball of a reproducing kernel Hilbert space (RKHS). Long et al. (2015) introduced Deep Adaption Network(DAN) for learning transferable deep embeddings with multiple kernel variant of MMD. Following this line, Sun & Saenko (2016) aligned the second-order statistics of two distributions; Zellinger et al. (2017) and Chen et al. (2020) went further to align the high order moments of domains. In addition to minimizing the domain discrepancy, many methods further work on the class-level discrepancy. Kang et al. (2019) proposed Contrastive Domain Discrepancy(CDD), which extended MMD to class-aware discrepancy, i.e. inter- and intra-class discrepancy based on conditional data distribution. Chen et al. (2019a) introduced a discriminative loss by forcibly increasing the margin between samples of different class and narrowing the distance between samples of the same class. Pan et al. (2019) introduced the prototype of each class in source domain. By predict-

ing the pseudo label for target samples, it aligned the class-aware prototypes for source, target and source-target mixed domains. Tang et al. (2020) incorporated the soft cluster assignments into the discriminative clustering to directly uncover the intrinsic target distribution as well as constrained the clustering solution on source data.

Many other measures of discrepancy are proposed in terms of manifold or optimal transport of distributions. Wang et al. (2018) embedded Geodesic Flow Kernel (Gong et al. (2012)) to transform the features to Grassman manifold and used MMD to align marginal and conditional distributions. Xu et al. (2019c) utilized a relaxed version of Hausdorff distance named d-SNE to minimize intra-class distance and maximize inter-class distance by nearest neighbors in manifold. Flamary et al. (2016); Courty et al. (2017); Damodaran et al. (2018) took advantage of simplified discrete version of optimal transport theory, concluding the domain adaption issue to find a discrete mapping between source and target data, with regularizations on either the mapping function (Flamary et al. (2016)) or the final classifiers(Damodaran et al. (2018)) Inspired by GAN(Goodfellow et al. (2014)), adversarial and generative methods are also used to learn the domain-invariant representation. DANN(Ganin et al. (2016)) introduced the gradient reversal layer to perform gradient-level adversarial updating. ADDA (Tzeng et al. (2017)) learned two different feature extraction networks with asymmetry adversarial losses on the source and target extractors. DM-ADA (Xu et al. (2019a)) incorporated domain mix-up to the VAE framework with discriminators using soft label and triplet loss. 3CATN (Li et al. (2019)) used two feature translators and three domain discriminators to minimize cycle-loss in the way like Cycle-GAN (Zhu et al. (2017)).

Note that although trying to, the current works do not take full usage of all data information. The data is a direct sampling of the underline distribution, thus a proper usage of the sample-based alignment would achieve more fine-grained and better asymptotic domain alignment. There are already some works (Chen et al. (2019a); Pan et al. (2019); Damodaran et al. (2018)) paying attention to investigate the sample level information in UDA. However, they either use intra-domain pairwise sample distance to measure the class margin, or arbitrarily construct pairwise data mapping cross domains. They do not consider the internal structure of the distribution (Flamary et al. (2016); Damodaran et al. (2018)), and even align the samples that may not be "equivalent" since the samples might be sampled differently in different domains. Tang et al. (2020) has pointed out the importance of the internal structure for domain adaption, but they only utilize the class information by structural clustering. In fact, aligning the distribution by sample data can trace back to the work of Sugiyama et al. (2008) and Gretton et al. (2009), but there is no related research in the UDA field.

3 Proposed Methods

Similar to the traditional UDA settings, we denote the source samples with ground truth as $\{(x_i^s, y_i^s)\}_{i=1}^{n^s}$, $x_i^s \in X^{\mathcal{S}} = \{x_i^s\}_{i=1}^{n^s}$, $y_i^s \in \mathcal{Y}$ and target samples as $\{x_j^t\}_{j=1}^{n^t} = X^{\mathcal{T}}$, where $\mathcal{Y}$ is the shared labeling set with C classes. $p_{\mathcal{S}}(x)$ and $p_{\mathcal{T}}(x)$ are the underline sample distributions for source and target domains, with $D^{\mathcal{S}}$ and $D^{\mathcal{T}}$ being the supports of those distributions respectively. $X^{\mathcal{S}}$ and $X^{\mathcal{T}}$ are the realization sets sampled from $D^{\mathcal{S}}$ and $D^{\mathcal{T}}$ following $p_{\mathcal{S}}(x)$ and $p_{\mathcal{T}}(x)$.

3.1 Mirrors cross Domains

We introduce a new concept named mirror that reflect equivalent samples cross domains. Formally, define the mirror pair as two realizations of random variables in the source and target distributions respectively that play "similar roles". In terms of the transportation theory (COT (2019)), let $\mathbb{T}^s$ and $\mathbb{T}^t$ be the two transformation maps (push-forwards operators) on $p_{\mathcal{S}}$ and $p_{\mathcal{T}}$ such that the resulting distributions are same, i.e. $\mathbb{T}^s_{\#}p_{\mathcal{S}} = \mathbb{T}^t_{\#}p_{\mathcal{T}}$, then $x^s \in D^{\mathcal{S}}$ and $x^t \in D^{\mathcal{T}}$ are mirrors if $\mathbb{T}^s_{\#}p_{\mathcal{S}}(x^s) = \mathbb{T}^t_{\#}p_{\mathcal{T}}(x^t)$. In general, infinite number of mirrors can be found since $\forall x^s \in D^{\mathcal{S}}$, we could always find $x^t \in D^{\mathcal{T}}$ such that $\mathbb{T}^s_{\#}p_{\mathcal{S}}(x^s) = \mathbb{T}^t_{\#}p_{\mathcal{T}}(x^t)$(COT (2019)). An ideal distribution alignment can be achieved by aligning the every mirror pairs.

Rather than investigating $\mathbb{T}^s$ and $\mathbb{T}^t$ like Flamary et al. (2016) and Damodaran et al. (2018), we resort to deep neural networks to approximate the transformations. However, it is impractical to directly find the mirrors in application. The dataset $X^{\mathcal{S}}$ and $X^{\mathcal{T}}$ are actually a random sampling from the underline support $D^{\mathcal{S}}$ or $D^{\mathcal{T}}$. The real mirror for $x_i^s \in X^{\mathcal{S}}$ may not exists in $X^{\mathcal{T}}$ at all.

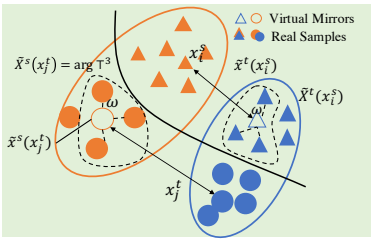


Figure 2: Illustrations of the virtual mirrors: the mirror sets $\tilde{X}^s$ and $\tilde{X}^t$ are calculated by Eq.(5) and virtual mirrors are calculated by Eq.(6). We show the case when we set $k = 3$, i.e. using the top-3 nearest samples to construct the virtual mirrors.

Some illustrative examples are shown in the middle of Fig.1, where d, e, f and $\tilde{c}, \tilde{h}, \tilde{i}, \tilde{j}$ do not have mirrors. To cope with this issue, we summarize the two asymptotic properties of the mirrors. Those properties are later used to construct the virtual mirror for real world data and further formulate cross domain distribution constraints for alignment. In the following, we also use x_i^s and x_j^s, $X^{\mathcal{S}}$ and $X^{\mathcal{T}}$, $p_{\mathcal{S}}$ and $p_{\mathcal{T}}$ to denote the samples' embedding features, sets and distributions after the certain feature extractor without introducing confusing notations.

Local Similarity of Mirrors. The most obvious mirrors are the centroids of the source and target domains for each class c, denoted by c_c^s and c_c^t. As pointed in Bengio et al. (2013) and Xu et al. (2019c), we could assume the learned features after certain deep feature extractor lie in a manifold, then one instance can be represented in terms of local probabilistic neighbors. For any x in $D^{\mathcal{S}}, D^{\mathcal{T}}$, define the position relative to the class c's centroid c_c as

$$q_c^{\{s,t\}}(x) = \frac{exp(-d(x, c_c^{\{s,t\}}))}{\sum_{c=1}^{C} exp(-d(x, c_c^{\{s,t\}}))} \tag{3}$$

where d is a distance measure. Then the representation of x considering the local similarity is $q^{\{s,t\}}(x) = [q_1^{\{s,t\}}, q_2^{\{s,t\}}, \ldots, q_C^{\{s,t\}}]$. If $x^t \in D^{\mathcal{T}}$ and $x^s \in D^{\mathcal{S}}$ are mirrors, their relative distance representation should be same, i.e. $q^s(x^s) = q^t(x^t)$.

Inter-domain Closeness of Mirrors. If the source domain and target domain are aligned ideally in the feature space, the centroids of the same class cross and the distribution of the representation x are exactly same. In other words, the source and target domains are two different views of the same distribution. The mirror pairs are exactly the same instance. In practice, strict alignment cannot be achieved. However, following the idea of d-SNE in (Xu et al. (2019c)), the mirror of an instance should be closer than any other instances in the counterpart the domains. This can be formulated as

$$x_i^s = \arg\min_{x \in D^{\mathcal{S}}} d(x, x_j^t) \tag{4}$$

where x_i^s is the mirror of $x_j^t \in D^{\mathcal{T}}$ in the source domain $D^{\mathcal{S}}$, vice versa.

The above two inter- and intra- domain properties of the mirrors pave the way to find the mirrors in real dataset and further achieve the alignment(constraint) cross domains.

3.2 Domain Alignment with Virtual Mirror Samples

The above two properties for mirror pairs are stated in terms of $D^{\mathcal{S}}$ and $D^{\mathcal{T}}$ rather than real data sets $X^{\mathcal{S}}$ and $X^{\mathcal{T}}$. The mirror of a specific instance $x_i^s \in X^{\mathcal{S}}$ may not exist in $X^{\mathcal{T}}$. Fortunately, we could define an approximated mirror, i.e. virtual mirror using the above two properties in terms of $X^{\mathcal{S}}$ and $X^{\mathcal{T}}$. The mirror property in Eq.(4) is rewritten as

$$\tilde{X}^S(x_j^t) = \arg\top^k_{x \in X^S} d(x, x_j^t) \tag{5}$$

where $\top^k$ is the "top-k" operation that selects the top k of set in ascending order, $\tilde{X}^S(x_j^t)$ is the mirror set in source domain corresponding to the sample x_j^t. The virtual mirror sample approximating the real mirror can be constructed following the local linearity assumption in manifold based on the mirror set (see Fig 2). Specifically, the virtual mirror of x_j^t is

$$\tilde{x}^s(x_j^t) = \sum_{x \in \tilde{X}^s(x_j^t)} \omega(x, x_j^t) x \tag{6}$$

where $\omega(x, x_j^t) = e^{-d(x,x_j^t)} / \sum_{x \in \tilde{X}^s(x_j^t)} e^{-d(x,x_j^t)}$ or simply $1/k$, which is experimentally investigated in Appendix C.2 and the selection of k will be experimentally investigated in Section 5.3. Note that $\tilde{x}^s(x_j^t)$ may not exist in $X^{\mathcal{S}}$. Aligning the virtual mirrors is essentially different from the

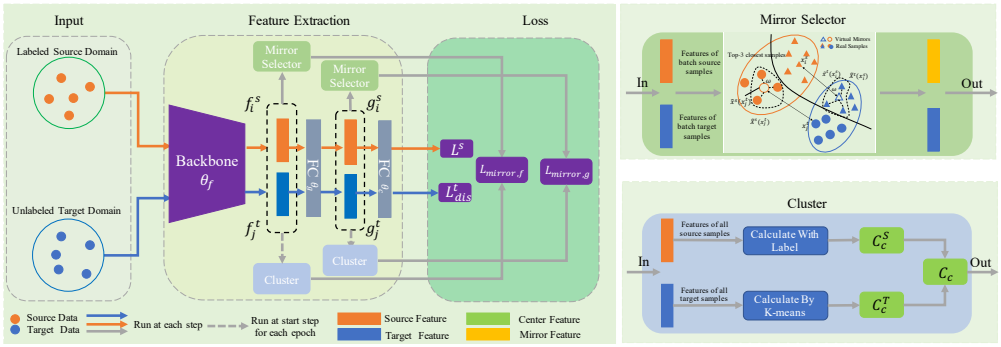


Figure 3: Overall structure of the model: the mirror selectors are applied after the averaging pooling of the backbone and the following FC layer; cluster algorithm is also applied on those representations layers to calculate the centers of both source and target domains.

work like Courty et al. (2017) and Damodaran et al. (2018) since we do not forcibly impose the pairwise mapping for the sampling data cross domains. When k approaches to n^s, the Eq.(5) will degenerate to MMD (Long et al. (2015)), i.e. the mirror of any target data is the mean of the source distribution. The alignment only considers the means of both domain distributions.

Obtaining the virtual mirror for each sample in both source and target domains, we propose the mirror loss, which is based on the local similarity property in Section 3.1 and formulated as

$$\mathcal{L}_{mirror} = \frac{1}{n^t}\sum_{i=1}^{n^t} KL(q(\tilde{x}^s(x_i^t))\|q(x_i^t)) + \frac{1}{n^s}\sum_{i=1}^{n^s} KL(q(\tilde{x}^t(x_i^s))\|q(x_i^s)) \tag{7}$$

where $x_i^s \in X^{\mathcal{S}}$, $x_j^t \in X^{\mathcal{T}}$, $q_c(x) = e^{-d(x,c_c)}/\sum_{c=1}^{C} e^{-d(x,c_c)}$ and $q(x) = [q_1(x), q_2(x), ..., q_C(x)]$. The mirror loss is to minimize the KL divergence of the miror paris' relative distances to the class centroids, which is motivated by the idea of the derivation of SNE in Hinton & Roweis (2003). For the source domain, we calculate the class-wise centroids using the ground truth $c_c^s = \frac{1}{n_c^s}\sum_{y_i^s=c} x_i^s$, n_c^s denote the number of samples belonging to the class c. Without the ground truth, we use K-Means (Dhillon et al. (2004)) to cluster target domain samples to get c_c^t. To enhance the alignment of both domains, we use the common centers of both domains as $c_c = 1/2c_c^s + 1/2c_c^t$ to calculate the local similarity in Eq.(7). Combining the Eq.(6) and (7) , the loss can be applied to any learnable representation layer to force a fine-grained alignment of distribution.

3.3 Model Structure and Algorithm

Incorporating the mirror loss into the model is simple and straightforward. Fig 3 illustrates a typical model structure. After the backbone network ends with pooling layer, two newly-added full-connected layers are appended. The first full-connected layer is to generate intermediate representation and last full-connected layer plays the role of the classifier based on specific task. We incorporate the mirror loss to do the alignment in two learnable space: the output of the final pooling layer of backbone and the first FC layer, denoted as $f \in \mathbb{R}^{d_f}$ and $g \in \mathbb{R}^{d_g}$ with dimensionality d_f and d_g respectively. As shown in Fig 3, the mirror selector uses the target feature f_j^t and the cached source feature $\{f_i^s\}$ (source feature f_i^s and cached target feature $\{f_j^t\}$) to find the virtual sample in f by Eq.(5) and (6). Same operation is carried on for g. The mirror losses for f and g are $\mathcal{L}_{mirror,f}$ and $\mathcal{L}_{mirror,g}$

For the labeled source data $\{(x_i^s, y_i^s)\}_{i=1}^{n_s}$, the generic cross entropy with respect to the source supervision is also involved, i.e. $\mathcal{L}^s = -\frac{1}{n^s}\sum_{i=1}^{n^s}\sum_{c=1}^{C}\mathbb{I}(y_i^s = c)\log p_{i,c}^s$. where $p_{i,c}^s$ is the predicted probability for class c and $\mathbb{I}$ is the indicator function. Although there is no ground truth for target domain, we use the unsupervised discriminative clustering method introduced in Jabi et al. (2019). Following Jabi et al. (2019), denote the introduced auxiliary target distribution as z_i^t, then the discriminative clustering method can be formulated as two steps: Parameter learning step, which regards the z_i^t as the pseudo label of the target domain labels and updates the network by minimizing the cross entropy loss; $\mathcal{L}_{dis}^t = -\frac{1}{n^t}\sum_{i=1}^{n^t} z_i^t \log p_i^t$, where p_i^t is the predicted probability using current learned network; Auxiliary update step, which updates the auxiliary distribution z_i^t as

$z^t_{i,c} \propto \frac{p^t_{i,c}}{(\sum_{i=1}^{n^t} p^t_{i,c})^{1/2}}$. The detailed derivation can be found in Jabi et al. (2019). The above two steps will iterate once for each epoch. The overall loss of the model is

$$\mathcal{L} = \mathcal{L}^s + \mathcal{L}^t_{dis} + \gamma(\mathcal{L}_{mirror,f} + \mathcal{L}_{mirror,g}) \tag{8}$$

where γ is the weight of mirror loss. The detailed algorithm can be found in Appendix A.2.

4 Theoretical Analysis

Although the Eq.(1) and (2) can explain the motivation of our proposed method, we will give a detailed analysis for the mirror sample alignment with respect to the theoretical framework for the domain adaption in Ben-David et al. (2010). The proofs of the propositions are in Appendix B.

Lemma 1. *Ben-David et al. (2010) Given the hypothesis class as $\mathcal{H}$, then we have*

$$\forall h \in \mathcal{H}, \mathcal{R}_{\mathcal{T}}(h) \leq \mathcal{R}_{\mathcal{S}}(h) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{S}, \mathcal{T}) + \lambda \tag{9}$$

where

$$\lambda = \min_{h \in \mathcal{H}} \{\mathcal{R}_{\mathcal{S}}(h, h_{\mathcal{S}}) + \mathcal{R}_{\mathcal{T}}(h, h_{\mathcal{T}})\} \tag{10}$$

and

$$d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{S}, \mathcal{T}) = 2 \sup_{h, h' \in \mathcal{H}} |Pr_{x \sim D^{\mathcal{S}}}[h(x) \neq h'(x)] - Pr_{x \sim D^{\mathcal{T}}}[h(x) \neq h'(x)]| \tag{11}$$

$h_{\mathcal{S}}$ and $h_{\mathcal{T}}$ are the labeling function in each domains.

Lemma 1 lays a generic analysis basis for domain adaption on the risk of target domain $\mathcal{R}_{\mathcal{T}}(h)$ under a given function space $\mathcal{H}$. The upper bound consists of three part: the risk of source domain $\mathcal{R}_{\mathcal{S}}(h)$, the domain discrepancy in terms of functional differences $d_{\mathcal{H}\Delta\mathcal{H}}$ and the combined error of the ideal joint hypothesis λ.

Proposition 1. *Define $\mathcal{H}$ as the hypothesis class of function mapping from $\mathcal{D}$ to $\mathcal{Y}$, where $\mathcal{D}$ is an intermediate representation space. $D^{\mathcal{S}}$ and $D^{\mathcal{T}}$ are the supports of distribution for source and target distribution in D. If $\mathcal{S}$ and $\mathcal{T}$ are aligned in distribution in space $\mathcal{D}$, then $d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{S}, \mathcal{T}) \to 0$.*

Proposition 1 states that the distribution alignment for certain learnable space will reduce the domain discrepancy in terms of functional differences $d_{\mathcal{H}\Delta\mathcal{H}}$.

The above distribution alignment can be achieved **empirically** by minimizing the $\mathcal{L}_{mirror}$ in Eq.(7). From the definition, we can see that $\mathcal{L}_{mirror}$ is minimized if and only if $q^s(x^s_i) = q^t(x^t_j)$ for every mirror pairs between the domains. It means: 1) the class centers for both source and target domain are same, 2) the mirror pairs cross domains have the same position relative to the common centers $c_c, c = 1, 2, \cdots, C$. Thus the empirical density function $\tilde{\Phi}_{\mathcal{S}}(x)$ and $\tilde{\Phi}_{\mathcal{T}}(x)$ over $X^{\mathcal{S}}$ and $X^{\mathcal{T}}$ are same. $\tilde{\Phi}_{\mathcal{S}}(x) = \tilde{\Phi}_{\mathcal{T}}(x)$. Under the assumption that $X^{\mathcal{S}}$ and $X^{\mathcal{T}}$ are unbiasedly sampled from $D^{\mathcal{S}}$ and $D^{\mathcal{T}}$ according to the density function Φ_S and Φ_T, Glivenko–Cantelli theorem (Rachev et al. (2013)) could assure when $n^t, n^s \to \infty$, we have

$$\Phi_{\mathcal{S}}(x) \overset{a.s.}{=} \tilde{\Phi}_{\mathcal{S}}(x) = \tilde{\Phi}_{\mathcal{T}}(x) \overset{a.s.}{=} \Phi_{\mathcal{T}}(x) \tag{12}$$

where $a.s.$ means almost surely. Minimizing $\mathcal{L}_{mirror}$ enforces the learned feature space to be the same, further reducing $d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{S}, \mathcal{T})$ to zero empirically when the number of sample is large.

Proposition 2. *Define $\lambda_m + \frac{1}{2} d^m_{\mathcal{H}\Delta\mathcal{H}}$ as the corresponding $\lambda + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}$ with the $\mathcal{L}_{mirror}$ minimized, $\lambda_o + \frac{1}{2} d^o_{\mathcal{H}\Delta\mathcal{H}}$ as the same part without minimizing $\mathcal{L}_{mirror}$. If minimizing $\mathcal{L}_{mirror}$ aligns the distribution in the learned space, we have*

$$\lambda_m + \frac{1}{2} d^m_{\mathcal{H}\Delta\mathcal{H}} \leq \lambda_o + \frac{1}{2} d^o_{\mathcal{H}\Delta\mathcal{H}} \tag{13}$$

Proposition 2 states when using the mirror loss would get a lower gap between the target risk and source risk, which is consistent to our motivation in Eq.(2). The key insight behind proposition 2 is that if the discrepancy of domains is empirically approaching to 0 by the loss $\mathcal{L}_{mirror}$ according to proposition 1, we can have a more relaxed feasible hypothesis set of $\mathcal{H}$, leading to a lower value of λ.

5 EXPERIMENTS AND RESULTS

5.1 DATASETS AND IMPLEMENTATIONS

We use **Office-31** (Saenko et al. (2010)), **Office-Home**(Venkateswara et al. (2017)), **ImageCLEF** and **VisDA2017**(Peng et al. (2017)) to validate our proposed method. For the first three datasets, we use all the tasks. For the VisDA2017, we use "train" set as source domain and "validation" set as target domain on classification task. We implement our experiments in PyTorch. For all tasks, we use ResNet50 or ResNet101 pre-trained on the ImageNet as backbone. To adapt the task, the number of units in fully connected layers are changed. Details are in Appendix A.1.

5.2 RESULTS AND COMPARISON WITH THE STATE-OF-THE-ART

Table 1: Average accuracy(%) on Office-31, Office-Home, ImageCLEF and VisDA2017. All the results are trained based on ResNet50 except those with mark†, which are trained based on ResNet101. Red indicates the best result while Blue means the second best.

Method	Office-31	Office-Home	ImageCLEF	VisDA2017
Source Model(He et al. (2016))	76.1	46.1	80.7	52.4†
DAN (Long et al. (2015))	82.3	56.3	—	—
DANN (Ganin et al. (2016))	82.6	57.6	—	—
ADDA (Tzeng et al. (2017))	83.2	—	—	—
JDDA (Chen et al. (2019a))	80.2	—	—	—
DSR (Cai et al. (2019))	88.6	64.9	—	—
DM-ADA (Xu et al. (2019a))	81.6	—	—	75.6†
rRevGrad+CAT (Deng et al. (2019))	87.6	—	87.3	—
SAFN (Xu et al. (2019b))	87.1	67.3	88.9	—
MDD (Xu et al. (2019b))	88.9	68.1	—	74.6
SymNets (Zhang et al. (2019a))	—	—	89.9	—
CAN (Kang et al. (2019))	90.6	—	—	**87.2**†
SHOT (Liang et al. (2020))	88.7	—	—	79.6†
MCSD (Zhang et al. (2020))	—	—	90.0	71.3
SRDC (Tang et al. (2020))	**90.8**	**71.3**	**90.9**	—
Ours	**91.1**	**72.6**	**91.6**	**87.9**†

Table 1 shows the average results of our method on the four datasets (The detailed results for each task can be found in Appendix C.1). We can find that our method has made a significant improvement over the existing SOTA methods. For the relatively simple datasets Office-31 and ImageCLEF, our method improves by 0.3% and 0.7%, respectively. For the more challenging dataset Office-Home, our method improves by 1.3%. The average accuracy of our method can also be significantly improved on large-scale dataset VisDA2017, i.e. a 0.7% improvement on it.

5.3 ABLATION STUDY AND PARAMETER SENSITIVITY

Table 2: Ablation studies using Office-Home dataset based on ResNet50.($K = 3$)

Baseline	DC	FC Mirror	Backbone Mirror	Avg
✓				46.1
✓	✓			71.6
✓	✓	✓		71.7
✓		✓	✓	65.5
✓	✓	✓	✓	**72.0**

We take Office-Home as an example to investigate the different components of the proposed model. In Table **??**, "Baseline" only uses backbone and the labeled source data to train the model and test on target domain; "DC" means discriminative clustering described in Section 3.3 and Eq.(8); "Backbone Mirror" means applying the mirror loss to the last pooling layer of the backbone; "FC Mirror" means applying the mirror loss to the output of first fully-connected layer after the backbone. We set $K = 3$ for all the experiments. From the results, we can observe that both "DC" and mirror alignment can improve the accuracy on target domain. Without discriminative clustering, the accuracy is reduced by 6.5%. Without mirror alignment, the accuracy is reduced by 0.4%. Applying the mirror loss on both the backbone and FC layer is helpful for the final performance. The detailed results for each task are in in Appendix C.2.

The parameter K in Eq.(5) controls how we construct the virtual mirrors. A larger K means choosing a larger mirror set and it may lead to more indistinguishable mirrors. A smaller K may get a wrong/unstable mirror. In Table 3, we investigate the accuracies with different Ks, i.e. $K = 1, 3, 5, 7, 9$. In terms of average accuracy, we can see that $K = 3$ is the best choice. However,

Table 3: The influence of K in Mirror Selector for Office-Home

K	Ar–Cl	Ar–Pr	Ar–Rw	Cl–Ar	Cl–Pr	Cl–Rw	Pr–Ar	Pr–Rw	Pr–Cl	Rw–Pr	Rw–Cl	Rw–Ar	Avg.
1	**52.37**	75.3	**81.4**	71.0	76.9	69.98	77.3	54.57	82.3	85.2	57.22	76.64	71.69
3	52.01	75.7	81.1	**71.9**	**77.0**	69.65	**78.7**	54.89	82.1	**85.2**	58.55	77.3	**72.00**
5	51.18	75.3	81.2	70.7	76.2	70.85	76.8	54.89	**82.3**	84.8	58.59	76.93	71.64
7	51.69	74.7	80.9	70.3	76.2	70.31	77.2	**55.71**	81.6	84.9	**59.37**	**77.87**	71.72
9	51.07	**75.8**	80.5	70.9	76.3	**70.89**	77.3	55.34	82.2	84.7	57.93	77.21	71.67

for different tasks, the choice might be different. For example, for the task of Pr-Ar, the optimal K is 3 while for Rw-Cl, the optimal K will be 7 with a large margin.

Table 4: The sensitivity of γs for the Mirror Loss for Office-Home

λ_{mirror}	Ar–Cl	Ar–Pr	Ar–Rw	Cl–Ar	Cl–Pr	Cl–Rw	Pr–Rw	Pr–Ar	Pr–Cl	Rw–Pr	Rw–Cl	Rw–Ar	Avg.
0.0	52.05	74.9	**81.7**	70.3	76.6	77.9	**82.2**	70.56	**55.3**	84.8	58.09	76.43	71.74
1.0	52.01	**75.7**	81.1	**71.9**	77.0	**78.7**	82.1	69.65	54.89	**85.2**	58.55	77.3	**72.01**
2.0	**51.98**	75.6	81.1	71.3	**77.2**	77.0	81.4	**72**	54.68	84.8	**58.71**	**77.46**	71.94
3.0	51.43	75.2	80.8	70.5	77.1	77.6	81.9	70.02	54.68	85.1	57.52	76.84	71.56

In Eq.(8), γ controls the weight of mirror loss. In Table 4, we tried different values of γs (from 0.0 to 3.0) in order to find the best choice suitable for different tasks. Note the $\gamma = 0.0$ means we do not use mirror loss. We can see that different tasks have different optimal γs. In most of the tasks, the optimal γ is in range $1.0 \sim 2.0$.

5.4 Visual Cases Analysis

Figure 4: Visualization of mirror sets in Office-Home. The source domain is Product(Pr) and target domain is Art(Ar). In each class, the first row is the "top-3" mirror set using embeddings trained without mirror loss and the second row is obtained by the proposed method.

To further illustrate what the virtual mirror our method can find, we visualize the mirror set defined in Eq.(5) in Fig.4. We can see that the top-3 similar samples with mirror loss are more similar compared with the results without using mirror loss. The results without mirror loss might even consist of quite dissimilar samples although they belong to the same class label (see the "clock" class in the upper-right class of Fig.4). This means our proposed method can align the distribution in more fine-grained way than the methods without using mirror samples. We further visualize the feature distributions by t-SNE (Hinton & Roweis (2003)) in Appendix D.2 to show our methods have better alignment for intra-cluster distributions.

6 Conclusion and Future Work

In this paper, we introduce a new concept called (virtual) mirror in unsupervised domain adaption. Utilizing the asymptotic properties of the mirror samples, we could setup a connection between the source and target domains by mirror loss. Under the unbiased sampling assumption, the proposed method is expected to have lower asymptotic empirical risk. We performed extensive experiments on benchmarks and achieved competitive results. However, the performance of the proposed method relies on the sampling process of the training data from the unknown underline distribution(although we generally assumes the sampling is unbiased). Another limitation might be the case when the dataset is large, the mirror selector might be time-consuming. The optimal approximated scheme for mirror selection as well as its impacts on the bound of empirical risk can be further investigated using theory like Ordinal Optimization(Ho et al. (1992)).

REFERENCES

Computational optimal transport. *Foundations and Trends in Machine Learning*, 11(5-6):355–607, 2019.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1-2):151–175, 2010.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

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.

Ruichu Cai, Zijian Li, Pengfei Wei, Jie Qiao, Kun Zhang, and Zhifeng Hao. Learning disentangled semantic representation for domain adaptation. In *IJCAI: proceedings of the conference*, volume 2019, pp. 2060. NIH Public Access, 2019.

Chao Chen, Zhihong Chen, Boyuan Jiang, and Xinyu Jin. Joint domain alignment and discriminative feature learning for unsupervised deep domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 3296–3303, 2019a.

Chao Chen, Zhihang Fu, Zhihong Chen, Sheng Jin, Zhaowei Cheng, Xinyu Jin, and Xian-Sheng Hua. Homm: Higher-order moment matching for unsupervised domain adaptation. *order*, 1(10): 20, 2020.

Chaoqi Chen, Weiping Xie, Wenbing Huang, Yu Rong, Xinghao Ding, Yue Huang, Tingyang Xu, and Junzhou Huang. Progressive feature alignment for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 627–636, 2019b.

Nicolas Courty, Rémi Flamary, Amaury Habrard, and Alain Rakotomamonjy. Joint distribution optimal transportation for domain adaptation. In *Advances in Neural Information Processing Systems*, pp. 3730–3739, 2017.

Bharath Bhushan Damodaran, Benjamin Kellenberger, Remi Flamary, Devis Tuia, and Nicolas Courty. Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

Zhijie Deng, Yucen Luo, and Jun Zhu. Cluster alignment with a teacher for unsupervised domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 9944–9953, 2019.

Inderjit S Dhillon, Yuqiang Guan, and Brian Kulis. Kernel k-means: spectral clustering and normalized cuts. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 551–556, 2004.

R Flamary et al. Optimal transport for domain adaptation, 2016.

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

Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2066–2073. IEEE, 2012.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Arthur Gretton, Alex Smola, Jiayuan Huang, Marcel Schmittfull, Karsten Borgwardt, and Bernhard Schölkopf. Covariate shift by kernel mean matching. *Dataset shift in machine learning*, 3(4):5, 2009.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(25):723–773, 2012. URL `http://jmlr.org/papers/v13/gretton12a.html`.

Xiang Gu, Jian Sun, and Zongben Xu. Spherical space domain adaptation with robust pseudo-label loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9101–9110, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Geoffrey E Hinton and Sam T Roweis. Stochastic neighbor embedding. In *Advances in neural information processing systems*, pp. 857–864, 2003.

Yu-Chi Ho, R_S Sreenivas, and P Vakili. Ordinal optimization of deds. *Discrete event dynamic systems*, 2(1):61–88, 1992.

Mohammed Jabi, Marco Pedersoli, Amar Mitiche, and Ismail Ben Ayed. Deep clustering: On the link between discriminative models and k-means. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4893–4902, 2019.

Chen-Yu Lee, Tanmay Batra, Mohammad Haris Baig, and Daniel Ulbricht. Sliced wasserstein discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 10285–10295, 2019.

Jingjing Li, Erpeng Chen, Zhengming Ding, Lei Zhu, Ke Lu, and Zi Huang. Cycle-consistent conditional adversarial transfer networks. In *Proceedings of the 27th ACM International Conference on Multimedia*, pp. 747–755, 2019.

Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pp. 97–105, 2015.

Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Unsupervised domain adaptation with residual transfer networks. In *Advances in neural information processing systems*, pp. 136–144, 2016.

Zhihe Lu, Yongxin Yang, Xiatian Zhu, Cong Liu, Yi-Zhe Song, and Tao Xiang. Stochastic classifiers for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9111–9120, 2020.

Yingwei Pan, Ting Yao, Yehao Li, Yu Wang, Chong-Wah Ngo, and Tao Mei. Transferrable prototypical networks for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2239–2247, 2019.

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

Svetlozar T. Rachev, Lev B. Klebanov, Stoyan V. Stoyanov, and Frank Fabozzi. Glivenko–cantelli theorem and bernstein–kantorovich invariance principle. 10.1007/978-1-4614-4869-3(Chapter 12):283–296, 2013.

Alan Ramponi and Barbara Plank. Neural unsupervised domain adaptation in nlp—a survey. *arXiv preprint arXiv:2006.00632*, 2020.

Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. permission. transferring visual category models to new domains. 2010.

Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3723–3732, 2018.

Yuan Shi and Fei Sha. Information-theoretical learning of discriminative clusters for unsupervised domain adaptation. *arXiv preprint arXiv:1206.6438*, 2012.

Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244, 2000.

Masashi Sugiyama, Shinichi Nakajima, Hisashi Kashima, Paul V Buenau, and Motoaki Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. In *Advances in neural information processing systems*, pp. 1433–1440, 2008.

Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European conference on computer vision*, pp. 443–450. Springer, 2016.

Hui Tang, Ke Chen, and Kui Jia. Unsupervised domain adaptation via structurally regularized deep clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8725–8735, 2020.

Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7167–7176, 2017.

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

Jindong Wang, Wenjie Feng, Yiqiang Chen, Han Yu, Meiyu Huang, and Philip S Yu. Visual domain adaptation with manifold embedded distribution alignment. In *Proceedings of the 26th ACM international conference on Multimedia*, pp. 402–410, 2018.

Minghao Xu, Jian Zhang, Bingbing Ni, Teng Li, Chengjie Wang, Qi Tian, and Wenjun Zhang. Adversarial domain adaptation with domain mixup. *arXiv preprint arXiv:1912.01805*, 2019a.

Ruijia Xu, Guanbin Li, Jihan Yang, and Liang Lin. Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1426–1435, 2019b.

Xiang Xu, Xiong Zhou, Ragav Venkatesan, Gurumurthy Swaminathan, and Orchid Majumder. d-sne: Domain adaptation using stochastic neighborhood embedding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2497–2506, 2019c.

Werner Zellinger, Thomas Grubinger, Edwin Lughofer, Thomas Natschläger, and Susanne Saminger-Platz. Central moment discrepancy (cmd) for domain-invariant representation learning. *arXiv preprint arXiv:1702.08811*, 2017.

Yabin Zhang, Hui Tang, Kui Jia, and Mingkui Tan. Domain-symmetric networks for adversarial domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5031–5040, 2019a.

Yabin Zhang, Bin Deng, Hui Tang, Lei Zhang, and Kui Jia. Unsupervised multi-class domain adaptation: Theory, algorithms, and practice. *arXiv preprint arXiv:2002.08681*, 2020.

Yuchen Zhang, Tianle Liu, Mingsheng Long, and Michael I Jordan. Bridging theory and algorithm for domain adaptation. *arXiv preprint arXiv:1904.05801*, 2019b.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2223–2232, 2017.

A ALGORITHM AND IMPLEMENTATION DETAILS

A.1 DATASETS

We use **Office-31** (Saenko et al. (2010)), **Office-Home**(Venkateswara et al. (2017)), **ImageCLEF** and **VisDA2017**(Peng et al. (2017)) to validate our proposed method. Office-31 has three domains: Amazon(A), Webcam(W) and Dslr(D) with 4,110 images belonging to 31 classes. Office-Home is a more challenging benchmark dataset for unsupervised domain adaption. It contains 15,500 images of 65 classes with four domains: Art(Ar), Clipart(Cl), Product(Pr) and RealWorld(Rw). ImageCLEF contains 600 images of 12 classes, where the images are divided into three domains: Caltech-256(C), ILSVRC 2012(I), Pascal VOC 2012(P). For the above three datasets, we use all the adaption tasks. VisDA2017 is a large-scale dataset which contains $\sim$280K images belonging to 12 classes. These images are divided into three parts: train, validation and test. We use "train" as source domain and "validation" as target domain. The source domain is composed of 152,397 images, which are generated from synthetic renderings of 3D models. The target domain is composed of 55,388 images cropped from Microsoft COCO dataset Lin et al. (2014).

A.2 ALGORITHM AND IMPLEMENTATIONS

Following the standard protocol in UDA, we use all the labeled source domain samples and all unlabeled target domain samples for training. We implement our model in PyTorch. For all tasks, we use ResNet50 or ResNet101 pre-trained on the ImageNet as backbone. The number of units in final fully connected layers are changed according to different tasks. All the input images are resized to 224 and normalized by mean [0.485, 0.456, 0.406] and standard deviation [0.229, 0.224, 0.225]. Data augmentations include "RandomCrop" and "RandomHorizontalFlip" for train images and only "CenterCrop" for test images. We adopt SGD to optimize the total loss. The learning rate is adjusted by $\eta_p = \eta_0(1+\alpha p)^{-\beta}$ like Ganin et al. (2016), where p is the epoch which is normalized in [0, 1], and $\eta_0 = 0.001$, $\alpha = 10$, $\beta = 0.75$. And the learning rate of fully connected layers is 10 times that of the backbone layers. The detailed algorithm is presented in Algorithm 1. One should

Algorithm 1 Mirror Alignment Algorithm for UDA

Require:
Source Data $X^{\mathcal{S}} = \{x_i^s, y_i^s\}_{i=1}^{n_s}$
Target Data $X^{\mathcal{T}} = \{x_i^t\}_{i=1}^{n_t}$
Total epochs M, batch size B, Iterations for each epoch T, where $T = \lfloor M/B \rfloor$.
The learned parameters in backbone θ_f, in newly-added FC layer θ_g and the classifier parameters θ_c

Create feature caches for source sample $F_S \in \mathbb{R}^{d_f \times n_s}$, $G_S \in \mathbb{R}^{d_g \times n_s}$ and target samples $F_T \in \mathbb{R}^{d_f \times n_t}$and $G_T \in \mathbb{R}^{d_g \times n_t}$.
for $epoch = 1$ **to** M **do**
  Clear the caches G_S, F_S, G_T, G_S.
  Calculate the source features f_i^s and g_i^s for each sample and the centers $c_{f,c}^s$, $c_{g,c}^s$ for each class using current parameters by $c_{f,c}^s = \frac{1}{n_c^s}\sum_{y_i^s=c} f_i^s$ and $c_{g,c}^s = \frac{1}{n_c^s}\sum_{y_i^s=c} g_i^s$. Save the source features to G_S and F_S accordingly.
  Calculate target features and class centers c_f^t and c_g^t by K-means(Dhillon et al. (2004)), save the features to G_T and F_t respectively.
  for $t = 1$ **to** T **do**
    Choose a batch data $\{x_i^s\}_{b=1}^B$ and $\{x_j^t\}_{b=1}^B$ from $X^{\mathcal{S}}$, $X^{\mathcal{T}}$, with features $\{f_i^s\}_{b=1}^B$, $\{f_i^t\}_{b=1}^B$, $\{g_i^s\}_{b=1}^B$, $\{g_i^t\}_{b=1}^B$.
    Find the mirrors based on Eq.(5) and further calculate the loss by Eq.(8).
    Update parameters θ_g, θ_f and θ_c by SGD.
  end for
end for

note that:

- When calculating the centers $c^t_{f,c}$ and $c^t_{g,c}$ for the target domain, we use the centers of source domain for each class $c^s_{f,c}$ and $c^s_{g,c}$ as the initial centers. The resulting clusters will share the label of the initial centers.
- We need to cache all the features for both domains, denoted by F_S, G_S and F_T, G_T respectively in order to select mirrors. Those features and centers are updated per epoch rather than per batch.
- For large scale dataset such as VisDA2017, the Eq.(5) is memory and time consuming. Merge sort with top-k minimum heap for each partition is used since we only care about the top-k features rather than the orders of all samples.

B PROOFS OF PROPOSITIONS

B.1 PROOF OF PROPOSITION 1

Proof. Denote $\Phi_S(x), \Phi_T(x)$ as the density function for domain $\mathcal{S}$ and $\mathcal{T}$ in the learned feature space $\mathcal{D}$, with supports as $D^{\mathcal{S}}$ and $D^{\mathcal{T}}$ respectively. $\mathcal{H}$ is the hypothesis class of functions mapping from $\mathcal{D}$ to $\mathcal{Y}$. Following the definition of $d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{S},\mathcal{T})$ in Lemma 1, we have

$$\begin{aligned} d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{S},\mathcal{T}) = \sup_{h,h'\in\mathcal{H}} | & \int_{x\sim D^{\mathcal{S}}} \Phi_S(x)\mathbb{I}(h(x)\neq h'(x))dx \\ & - \int_{x\sim D^{\mathcal{T}}} \Phi_T(x)\mathbb{I}(h(x)\neq h'(x))dx| \end{aligned} \tag{14}$$

considering the fact that $Pr[x] = E[\mathbb{I}(x)]$, where $\mathbb{I}$ is the indicator function. If $\mathcal{S}$ and $\mathcal{T}$ are aligned in space $\mathcal{D}$, then both the density functions Φ_S, Φ_T and their supports $D^{\mathcal{S}}$, $D^{\mathcal{T}}$ are same. We have

$$\begin{aligned} & | \int_{x\sim D^{\mathcal{S}}} \Phi_S(x)\mathbb{I}(h(x)\neq h'(x))dx - \int_{x\sim D^{\mathcal{T}}} \Phi_T(x)\mathbb{I}(h(x)\neq h'(x))dx| \\ & \leq \int_{x\sim\mathcal{D}} |\Phi_S(x) - \Phi_T(x)||\mathbb{I}(h(x)\neq h'(x))|dx \to 0 \end{aligned} \tag{15}$$

□

B.2 PROOF OF PROPOSITION 2

Proof. Based on Ben-David et al. (2010), we have

$$\lambda_o = \min_{h\in\mathcal{H}}\{\mathcal{R}_{\mathcal{S}}(h,h_{\mathcal{S}}) + \mathcal{R}_{\mathcal{T}}(h,h_{\mathcal{T}})\} \tag{16}$$

Based on the Proposition 1, $d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{S},\mathcal{T})$ will approach 0 empirically if $\mathcal{L}_{mirror}$ is minimized indepent with $\mathcal{H}$. Thus $\forall h\in\mathcal{H}$, we have

$$\lambda_m + \frac{1}{2}d^m_{\mathcal{H}\Delta\mathcal{H}} = \lambda_m = \min_{h\in\mathcal{H}}\{\mathcal{R}_{\mathcal{S}}(h,h_{\mathcal{S}}) + \mathcal{R}_{\mathcal{T}}(h,h_{\mathcal{T}})\} \tag{17}$$

When the model is trained without $\mathcal{L}_{mirror}$, we can define a set $\mathcal{H}'\subset\mathcal{H}$ that satisfies $d_{\mathcal{H}'\Delta\mathcal{H}'} = 0$.

If $\mathcal{H}' = \emptyset$, the Eq.(13) holds naturally.

If $\mathcal{H}'\neq\emptyset$, then $\forall h\in\mathcal{H}$, we have

$$\begin{aligned} \lambda_o + \frac{1}{2}d^o_{\mathcal{H}\Delta\mathcal{H}} = \min\{ & \min_{h\in\mathcal{H}'}\{\mathcal{R}_{\mathcal{S}}(h,h_{\mathcal{S}}) + \mathcal{R}_{\mathcal{T}}(h,h_{\mathcal{T}})\}, \\ & \min_{h\in\mathcal{H}-\mathcal{H}'}\{\mathcal{R}_{\mathcal{S}}(h,h_{\mathcal{S}}) + \mathcal{R}_{\mathcal{T}}(h,h_{\mathcal{T}})\} + \frac{1}{2}d^o_{\mathcal{H}\Delta\mathcal{H}}\} \\ & \geq \min_{h\in\mathcal{H}}\{\mathcal{R}_{\mathcal{S}}(h,h_{\mathcal{S}}) + \mathcal{R}_{\mathcal{T}}(h,h_{\mathcal{T}})\} \\ & = \lambda_m + \frac{1}{2}d^m_{\mathcal{H}\Delta\mathcal{H}} \end{aligned} \tag{18}$$

since for any subset $\mathcal{H}'\subseteq\mathcal{H}$, $\min_{h\in\mathcal{H}'}\{h\} \geq \min_{h\in\mathcal{H}}\{h\}$. $h_{\mathcal{S}}$ and $h_{\mathcal{T}}$ are the labeling functions for source and target domains Ben-David et al. (2010).

□

C DETAILED RESULTS

C.1 DETAILED COMPARISIONS WITH SOTA METHODS

In this section, we describe the task-level detailed results of our experiment comparing with SOTA methods.

Table 5 shows the result of 6 tasks on Office-31 dataset. Compared with the SOTA result of SRDC(Tang et al. (2020)), the average accuracy of our model increases by 0.3%. Specially, we achieve a 1.5% improvement on task A to W.

Table 6 shows the result of 12 tasks on Office-Home dataset. The average accuracy of our method is 72.6%, which gains a 1.3% improvement compared with SRDC (Tang et al. (2020)). For some difficult tasks such as Cl to Ar, Pr to Ar, our method have achieved more than 1% improvement.

Table 7 shows the result of 6 tasks on ImageCLEF dataset. For all tasks, our method gain a new SOTA result and the average accuracy is 91.6% which has a 0.7% improvement.

For large-scale dataset VisDA2017, we migrated the proposed mirror loss to the existing method CAN Kang et al. (2019). We can observe from Table 8 that the average accuracy increases by 0.7% when using mirror on CAN.

Table 5: Test accuracy(%) on Office-31 dataset for unsupervised domain adaptation based on ResNet50.

Method	A-W	D-W	W-D	A-D	D-A	W-A	Avg
Source Model He et al. (2016)	68.4	96.7	99.3	68.9	62.5	60.7	76.1
DAN Long et al. (2015)	81.3±0.3	97.2±0.0	99.8±0.0	83.1±0.2	66.3±0.0	66.3±0.1	82.3
DANN Ganin et al. (2016)	81.7±0.2	98.0±0.2	99.8±0.0	83.9±0.7	66.4±0.2	66.0±0.3	82.6
ADDA Tzeng et al. (2017)	86.2±0.3	78.8±0.4	96.8±0.2	99.1±0.2	69.5±0.1	68.5±0.1	83.2
JDDA Chen et al. (2019a)	82.6±0.4	95.2±0.2	99.7±0.0	79.8±0.1	57.4±0.0	66.7±0.2	80.2
DSR Cai et al. (2019)	93.1	98.7	99.8	92.4	73.5	73.9	88.6
DM-ADA Xu et al. (2019a)	83.9±0.4	99.8±0.1	99.9±0.1	77.5±0.2	64.6±0.4	64.0±0.5	81.6
rRevGrad+CAT Deng et al. (2019)	94.4±0.1	98.0±0.2	100.0±0.0	90.8±1.8	72.2±0.6	70.2±0.1	87.6
SAFN Xu et al. (2019b)	90.1±0.8	98.6±0.2	99.8±0.0	90.7±0.5	73.0±0.2	70.2±0.3	87.1
MDD Xu et al. (2019b)	94.5±0.3	98.4±0.1	100.0±0.0	93.5±0.2	74.6±0.3	72.2±0.1	88.9
CAN Kang et al. (2019)	94.5±0.3	99.1±0.2	99.8±0.2	95.0±0.3	**78.0±0.3**	77.0±0.3	90.6
SHOT Liang et al. (2020)	90.9	98.8	99.9	93.1	74.5	74.8	88.7
SRDC Tang et al. (2020)	95.7±0.2	99.2±0.1	100.0±0.0	95.8±0.2	76.7±0.3	77.1±0.1	90.8
Ours	**97.2±0.3**	**99.2±0.1**	**100.0±0.0**	**96.2±0.1**	75.3±0.1	**78.2±0.1**	**91.1**

Table 6: Test accuracy(%) on Office-Home dataset for unsupervised domain adaptation based on ResNet50.

Method	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
Source Model 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
DAN Long et al. (2015)	43.6	57.0	67.9	45.8	56.5	60.4	44.0	43.6	67.7	63.1	51.5	74.3	56.3
DANN Ganin et al. (2016)	45.6	59.3	70.1	47.0	58.5	60.9	46.1	43.7	68.5	63.2	51.8	76.8	57.6
DSR Cai et al. (2019)	53.4	71.6	77.4	57.1	66.8	69.3	56.7	49.2	75.7	68.0	54.0	79.5	64.9
SAFN Xu et al. (2019b)	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. (2019b)	**54.9**	73.7	77.8	60.2	71.4	71.8	61.2	53.6	78.1	72.5	**60.2**	82.3	68.1
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
Ours	52.4	**76.5**	**81.4**	**72**	**77.2**	**78.6**	**72.1**	**55.7**	**82.4**	**77.8**	59.4	**85.3**	**72.6**

Table 7: Test accuracy(%) on ImageCLEF dataset for unsupervised domain adaptation based on ResNet50.

Method	I-P	P-I	I-C	C-I	C-P	P-C	Avg
Source Model He et al. (2016)	74.8±0.3	83.9±0.1	91.5±0.3	78.0±0.2	65.5±0.3	91.2±0.3	80.7
DAN Long et al. (2015)	74.5±0.3	82.2±0.2	92.8±0.2	86.3±0.4	69.2±0.4	89.8±0.4	82.5
DANN Ganin et al. (2016)	75.0±0.6	86.0±0.3	96.2±0.4	87.0±0.5	74.3±0.5	91.5±0.6	85.0
rRevGrad+CAT Deng et al. (2019)	77.2±0.2	91.0±0.3	95.5±0.3	91.3±0.3	75.3±0.6	93.6±0.5	87.3
SAFN Xu et al. (2019b)	79.3±0.1	93.3±0.4	96.3±0.4	91.7±0.0	77.6±0.1	95.3±0.1	88.9
SymNets Zhang et al. (2019a)	80.2±0.3	93.6±0.2	97.0±0.3	93.4±0.3	78.7±0.3	96.4±0.1	89.9
MCSD Zhang et al. (2020)	79.2±0.2	96.2±0.3	96.8±0.1	93.8±0.2	77.8±0.4	96.2±0.0	90.0
SRDC Tang et al. (2020)	80.8±0.3	94.7±0.2	97.8±0.2	94.1±0.2	80.0±0.3	97.7±0.1	90.9
Ours	**82.4±0.1**	**95.3±0.1**	**97.9±0.2**	**95.2±0.2**	**81.0±0.1**	**98.0±0.1**	**91.6**

Table 8: Test accuracy(%) on VisDA dataset for unsupervised domain adaptation based on ResNet101.

Method	airplane	bicycle	bus	car	horse	knife	motorcycle	persion	plant	skateboard	train	truck	Avg
Source Model 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
DAN Long et al. (2015)	87.1	63.0	76.5	42.0	90.3	42.9	85.9	53.1	49.7	36.3	85.8	20.7	61.1
SAFN Xu et al. (2019b)	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
SHOT Liang et al. (2020)	92.6	81.1	80.1	58.5	89.7	86.1	81.5	77.8	89.5	84.9	84.3	49.3	79.6
CAN Kang et al. (2019)	97.0	87.2	82.5	74.3	**97.8**	**96.2**	90.8	80.7	**96.6**	96.3	87.5	59.9	87.2
CAN+Mirror(Ours)	**97.2**	**88.2**	**84.9**	**76.0**	97.2	95.8	89.2	**86.4**	96.1	**96.6**	85.9	**61.2**	**87.9**

C.2 DETAILED RESULTS OF ABLATION STUDY

The detailed result of ablation study on Office-Home is shown in this section. In Table 9, "Baseline" only uses backbone and the labeled source data to train the model and test on target domain; "DC" means Discriminative Clustering which is described in Section 3.3 and Eq.(8); "Backbone Mirror" means applying the mirror loss to the last pooling layer of the backbone; "FC Mirror" is the results applying the mirror loss to the output of first fully-connected layer after the last pooling layer. Besides those above, we also investigate two methods synthesizing the virtual mirror in Eq.(6). One is $\omega(x, x_j^t) = e^{-d(x,x_j^t)} / \sum_{x \in \tilde{X}^s(x_j^t)} e^{-d(x,x_j^t)}$, which is denoted as "weighted mirror sample" the other is $\omega(x, x_i^t) = 1/k$. From Table 9, we can observe that both "DC" and mirror alignment can improve the accuracy on target domain. Without discriminative clustering, the accuracy is reduced by 6.5%. Without mirror alignment, the accuracy is reduced by 0.4%. Applying the mirror loss on both the backbone and FC layer is helpful for the final performance. We could also find that our method is less sensitivity to way we construct the virtual samples, i.e. either the average sample or "Weighted Mirror Sample".

Table 9: Ablation Studies using Office-Home dataset based on ResNet50.($K = 3$)

Baseline	DC	FC Mirror	Backbone Mirror	Weighted Mirror Sample	Ar-Cl	Ar-Pr	Ar-Rw	Cl-Ar	Cl-Pr	Cl-Rw	Pr-Ar
✓					34.9	50.0	58.0	37.4	41.9	46.2	38.5
✓	✓				51.1	**76.1**	**81.9**	70.7	76.18	77.5	**71.0**
✓	✓	✓			52.0	74.9	81.6	70.2	76.5	77.9	70.5
✓		✓	✓		47.2	69.1	76.1	60.9	66.5	69.7	62.8
✓	✓	✓	✓		**52.0**	75.6	81.0	**71.9**	**77.0**	**78.6**	69.6
✓	✓	✓	✓	✓	51.7	75.5	81.3	71.1	76.4	77.8	70.2

Baseline	DC	FC Mirror	Backbone Mirror	Weighted Mirror Sample	Pr-Cl	Pr-Rw	Rw-Ar	Rw-Cl	Rw-Pr	Avg
✓					31.2	60.4	53.9	41.2	59.9	46.1
✓	✓				54.5	82.2	76.2	56.8	85.1	71.6
✓	✓	✓			**55.3**	82.1	76.4	58.0	84.8	71.7
✓		✓	✓		48.3	77.2	73.1	53.7	81.3	65.5
✓	✓	✓	✓		54.8	82.1	**77.3**	**58.5**	85.1	**72.0**
✓	✓	✓	✓	✓	54.8	**82.3**	76.6	57.5	**85.3**	71.7

D VISUALIZATIONS

D.1 VISUAL CASES ANALYSIS

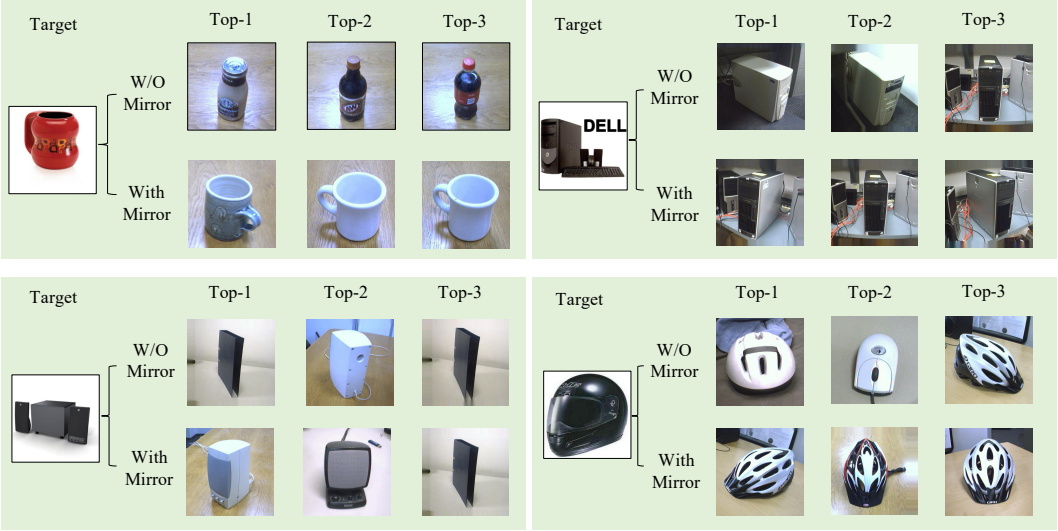
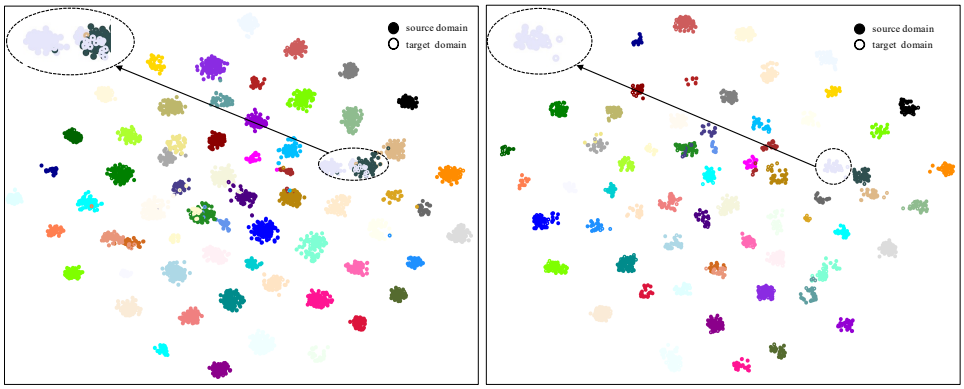


Figure 5: Visualization of mirror sets in Office-31. The source domain is "webcame" and target domain is "amazon". In each class, the first row is the top-3 mirror set using embeddings trained without mirror loss and the second row is obtained by the proposed method.

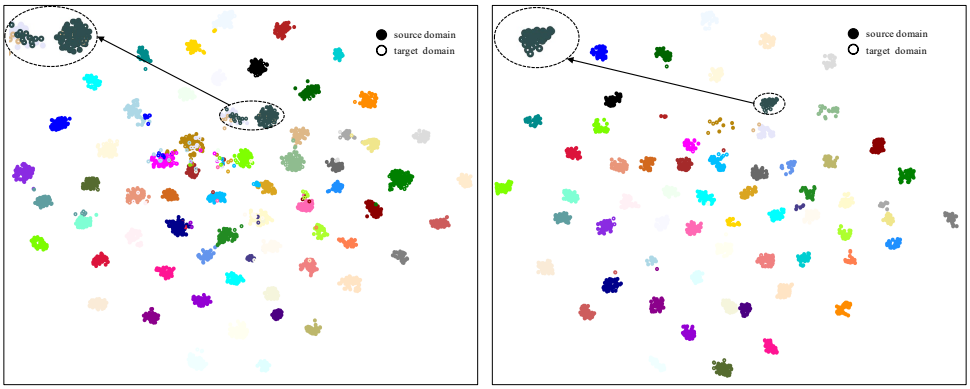
Besides the case visualization for Office-Home in Section 5.4, we also present the results for Office-31 in Fig.5. We could have similar observations as Fig.4. Specially, although the "Bottle" in the upper-left class of the Fig.5 has mirror sets belonging to the same class, but our proposed method gives results much more similar. For the "Helmet" class in the bottom-right class, the results without mirror loss consist of even different class samples, such as "mouse". Those means our proposed method can align the distribution in more fine-grained way than the methods without using mirror samples.

D.2 VISUALIZATION OF ALIGNED EMBEDDINGS

In order to show the effectiveness of our method in distribution alignment more vividly, we visualize the embedded features of samples using t-SNE(Hinton & Roweis (2003)) for tasks Cl-Rw, Rw-Cl in Office-Home and tasks A-W and W-A in Office-31 in Fig.6 and Fig.7. Besides the clusters are more tight for results with mirror loss, the "shape" of each class sample distribution cross domains are more alike for the results with mirror loss. In some cases, even the source and target samples are clustered together, the results without mirror loss show more non-overlap samples, meaning the distributions are not aligned well enough.

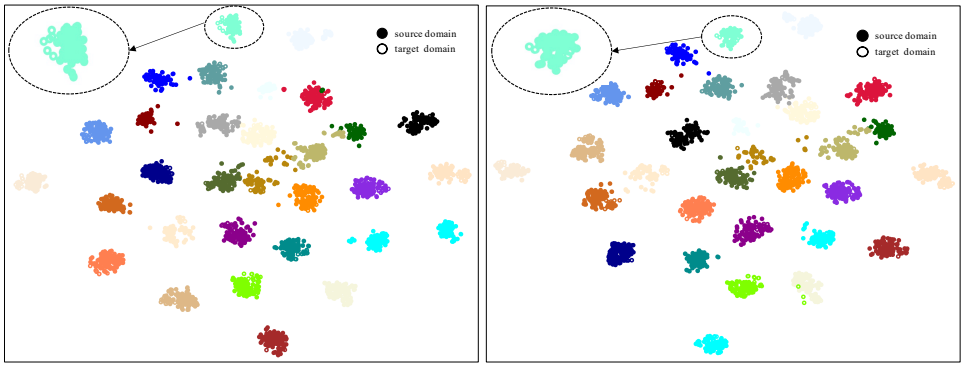


(a) w/o Mirror, Cl-Rw (b) Mirror Alignment, Cl-Rw

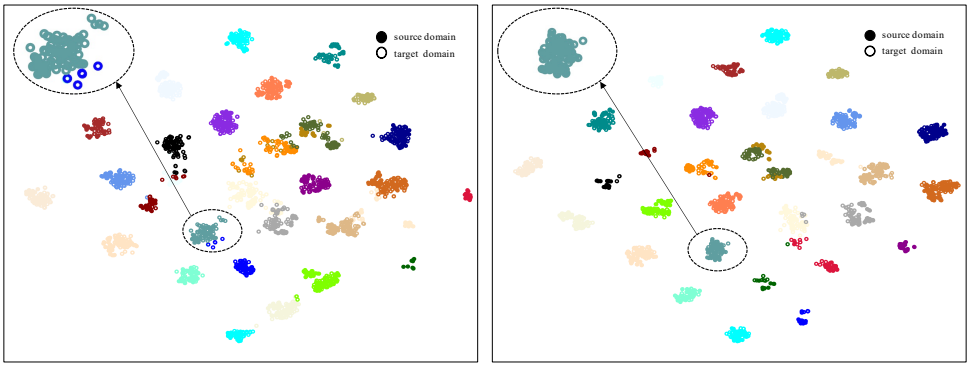


(c) w/o Mirror, Rw-Cl (d) Mirror Alignment, Rw-Cl

Figure 6: The t-SNE visualization of feature embeddings for 2 tasks of Office-Home. The solid points denote source data and circles denote target data. Different classes are distinguished by color.



(a) w/o Mirror, A-W (b) Mirror Alignment, A-W



(c) w/o Mirror, W-A (d) Mirror Alignment, W-A

Figure 7: The t-SNE visualization of feature embeddings for 2 tasks of Office-31. The solid points denote source data and circles denote target data. Different classes are distinguished by color.