Deep Subdomain Alignment for Cross-domain Image Classification

Yewei Zhao, Hu Han, Shiguang Shan, and Xilin Chen
Key Laboratory of Intelligent Information Processing, Chinese Academy of Sciences (CAS),
Institute of Computing Technology, CAS, Beijing 100190, China
University of the Chinese Academy of Sciences, Beijing 100049, China
zhaoyewei21@mails.ucas.ac.cn {hanhu, sgshan, xlchen}@ict.ac.cn

Abstract

Unsupervised domain adaptation (UDA), which aims to transfer knowledge learned from a labeled source domain to an unlabeled target domain, is useful for various cross-domain image classification scenarios. A commonly used approach for UDA is to minimize the distribution differences between two domains, and subdomain alignment is found to be an effective method. However, most of the existing subdomain alignment methods are based on adversarial learning and focus on subdomain alignment procedures without considering the discriminability among individual subdomains, resulting in slow convergence and unsatisfactory adaptation results. To address these issues, we propose a novel deep subdomain alignment method for UDA in image classification, which consists of a Union Subdomain Contrastive Learning (USCL) module and a Multi-view Subdomain Alignment (MvSA) strategy. USCL can create discriminative and dispersed subdomains by bringing samples from the same subdomain closer while pushing away samples from different subdomains. MvSA makes use of labeled source domain data and easy target domain data to perform target-to-source and target-to-target alignment. Experimental results on three image classification datasets (Office-31, Office-Home, Visda-17) demonstrate that our proposed method is effective for UDA and achieves promising results in several cross-domain image classification tasks. Our code will be available: https://github.com/zhaoyewei/DSACDIC.

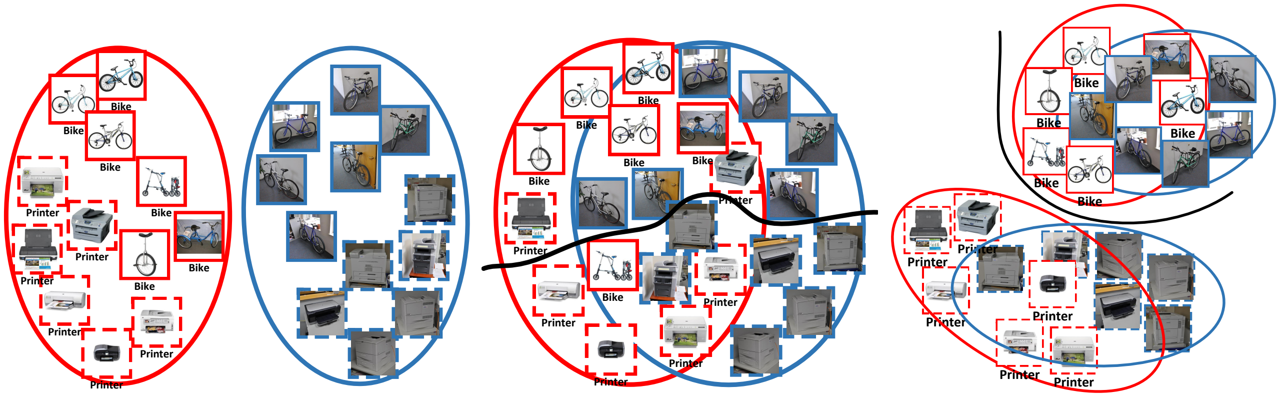
1. Introduction

Deep learning-based image classification methods have demonstrated remarkable performance when the training and testing data share a similar distribution [15]. However, their generalization ability declines significantly on the out-of-distribution data due to domain shift [31]. Additionally, these methods typically require a large amount of labeled data for training, which can be both laborious and costly to collect and annotate in practical scenarios. Hence, there is a strong incentive to develop image classification methods that can effectively make use of the useful information learned in the source domain. Unsupervised Domain Adaptation (UDA) can address this challenge by transferring the knowledge acquired from a labeled source domain to a target domain without requiring any labeled data in the target domain. Thus, UDA-based image classification has attracted increasing attention.

As shown in Figure 1a, there exists a notable distinction between the distribution of the labeled source domain and the unlabeled target domain. A key point in UDA is to formulate an appropriate metric for quantifying the divergence between two different distributions. The aim is to narrow down the distributional differences in order to facilitate the knowledge transfer from the source to the target domain. We denote those methods as statistic matching-based methods [17]. One particularly favored approach is Maximum Mean Discrepancy (MMD), a global distribution metric based on embedding distribution measures in Reproducing Kernel Hilbert Space (RKHS). Figure 1b shows an intuitive example of global distribution alignment. Due to its simplicity and solid theoretical foundation, MMD has been applied successfully in various problems.

Based on MMD, many methods [3, 23, 25, 34, 42] are proposed to align the global distribution of different distributions, but without considering the relationship of subdomains. An inevitable situation with global distribution alignment methods is that the samples from different subdomains will become intermingled. To address this issue, recent works [21, 24, 40, 41, 46, 49] focus on aligning the distributions of relevant subdomains within both the source and target domains. As shown in Figure 1c, such a subdomain alignment strategy promotes the similarity of distributions within the same subdomain, thereby enhancing

This research was supported in part by the National Key R&D Program of China (grant 2018AAA0102402) and the National Natural Science Foundation of China (grant 62176249).



(a) Distributions of domains (b) Global alignment (c) Subdomain alignment

Figure 1. Schematic representation of the distribution of source domain and target domains and the different domain adaptation methods. We use blue and red ellipses to represent the source and target domains, respectively, and use solid and dashed lines to distinguish between the different categories. In addition, the black lines indicate the decision boundary.

the discriminability of individual categories. Notably, most of the subdomain alignment methods are adversarial-based and slow to coverage. DSAN [49], a statistic matching-based subdomain adaption method, learns a deep transfer network by aligning the relevant subdomain distributions of domain-specific layer activations across different domains based on local maximum mean discrepancy (LMMD). DSAN achieves remarkable performance in domain adaption. Nevertheless, it still has some limitations: **1)** DSAN uses LMMD to align the subdomain distributions so that samples of the same subdomain can be pushed closer in RKHS. However, it overlooks the strengthening of decision boundaries among subdomains, leading to inter-subdomain overlap. **2)** All the samples, regardless of source domain or target domain, are assigned equal importance in calculating the weight, causing hard samples with much noise in target domain to have a dominant influence on the domain adaptation process and harm the final performance.

As analyzed above, we should not only focus on aligning subdomains but also build firm decision boundary among subdomains to make deep network works better in classification. Furthermore, a new strategy is needed to make source domain samples with ground truths and easy target domain samples play a prior role in LMMD. Motivated by this, we propose our methods: Motivated by the analysis above, we propose a novel deep subdomain alignment-based UDA method. Drawing inspiration from supervised contrastive learning [19], we pull closer the samples within the same subdomain and simultaneously push away samples from different subdomains in the RKHS, leading to more discriminative and dispersed source subdomains. In addition, we design a new subdomain alignment strategy, which considers not only the alignment between the source and target subdomains but also the alignment between hard and easy samples in target domains based on a dynamic thresholding scheme. Such a multi-view subdomain alignment strategy can reduce the bias of distinct target domain samples to the domain adaptation process. We evaluate the proposed approach on three datasets, which shows the effectiveness of our methods for cross-domain image classification. We also provide ablation studies to validate the effectiveness of the key components of our method. The pipeline of our method is Figure 2. Our method follows the setting of DSAN, and compared to the origin DSAN as well as other baselines, our method achieves a remarkable performance in three primary domain adaption datasets. Ablation studies demonstrate the effectiveness of individual components of our pipeline in adapting the model from source domain to target domain. We also show that our method is more robust to noise caused by target samples and, as a result, more stable than the origin baseline method. The main contributions of our work are as follows.

- We propose a novel deep subdomain alignment-based method for cross-domain image classification, which considers not only the subdomain alignment but also the discriminability and dispersity of subdomains used for alignment.

- We design a multi-view subdomain alignment strategy that considers both target-to-source and target-to-target subdomain alignment to avoid distracter samples in target domain causing bias during adaptation.

2. Relate Works

2.1. Deep Learning Based Domain Adaption

Deep domain adaption methods [4, 20, 44] aim to transfer knowledge from labeled source domain to unlabeled target domain. Typically, those methods belong to two approaches [17]: **1)** adversarial-based methods, those methods are mainly inspired by Generative Adversarial Net(GAN) [8]. *E.g*. Domain Adversarial Neural Network (DANN) [7] integrates a domain discriminator to distinguish source features from target features and a feature generator to confuse the domain discriminator. **2)** Statistics matching-based methods provide interpretable and complementary properties to hypothesis-induced distributed distance. *E.g*. Maximum Mean Discrepancy (MMD) [9,10,23] measures source and target distribution distances in the probability space. Contrastive Adaptation Network (CAN) [18] modeled and optimized intra-class and inter-class domain discrepancy by alternatively estimating the target domain labels through clustering. Generally, adversarial-based methods achieve better performance than statistic matching-based methods [49], but properly designed statistic matching-based methods can also achieve remarkable performance and are solid in the theoretical foundation.

Maximum Mean Discrepancy. MMD and its variants are the most favored methods in statistic matching. Deep Domain Confusion (DDC) [38] directly applies MMD with a linear kernel to a feature layer. Gretton *et al*. proposes multiple-kernel MMD (MK-MMD) with multiple kernels to make feature space more distinguishable [11], and Deep Adaption Network (DAN) [23] introduces MK-MMD to measure the domain distances. Long *et al*. proposed Joint Adaptation Network (JAN) [25] with Joint Maximum Mean Discrepancy (JMMD) to measure the shift in joint distributions. Center moment discrepancy (CMD) [47] further matches the higher-order central moments of probability distribution using order-wise moment differences. Coutry *et al*. proposed Joint Distribution Optimal Transport (JDOT) [3] and Deep JDOT applied it to deep network. [42] prove that intra-class and inter-class distances are one fall and another rise, thus proposing a novel discriminable MMD to omit redundant parameters.

Subdomain Adaption. Subdomain adaption focuses on accurately aligning the distribution of relevant subdomains in both source and target domains [49]. Multi-Adversarial Domain Adaptation (MADA) [29] captures multimode structures to enable fine-grained alignment of different data distributions based on multiple domain discriminators. [21] proposed co-regularized domain alignment to construct multiple diverse feature spaces and align source and target distributions in each of them individually. Zhu *et al*. proposed Deep Subdomain Adaptation Network (DSAN) [49], which learns a deep transfer network by aligning the relevant subdomain distributions of domain-specific layer activations across different domains based on the local maximum mean discrepancy (LMMD).

2.2. Contrastive Learning

Contrastive learning [14] learns representations by contrasting positive pairs against negative paris. Typically, contrastive learning methods learn representations by pushing apart dissimilar data pairs while pulling together similar pairs [36], and the standard approach for generating positive pairs without additional annotations is to create multiple views of each datapoint [19]. Noise contrastive estimation (NCE) [13] learns to distinguish data from noise. [28] proposes InfoNCE estimate the mutual information between positive and associated negative pairs. [45] proposes to use a memory bank to store instance class representation vector. When labels are accessible, leveraging labeled data in contrastive representation learning improves performance by guiding representations towards task-relevant features [16, 19, 48]. Further, [19] extends the self-supervised batch contrastive approach to the fully supervised setting and proposes supervised contrastive learning to effectively leverage label information by clusters of points belonging to the same class being pulled together in embedding space while simultaneously pushing apart clusters of samples from different categories.

3. Proposed Method

3.1. Overview

Pipeline of our method is illustrated in Figure 2. f is the domain-specific feature representation and g is the classifier prediction. Memory bank with size $N_s + N_t$ stores features and labels of source domain samples as well as features and corresponding pseudo labels of easy samples from target domain. When the size of memory bank reaches $N_s + N_t$, we drop the oldest items and only keep the newest ones. Easy samples of target domain are screened out by a filter with a dynamic threshold decreasing along with iterations. The source domain branch (Red) is optimized with classification loss $\mathcal{L}_{cls}$ and source subdomain contrastive learning loss $\mathcal{L}_{uscl}$. The target domain branch (Blue) is optimized with alignment loss $\mathcal{L}_{align}$.

There are two key components proposed: union subdomain contrastive learning (USCL) and multi-view subdomain alignment (MvSA). USCL can create discriminative and dispersed source subdomains by bringing samples from the same subdomain closer while pushing away samples from different subdomains (see Figure 3a). MvSA makes use of labeled source domain data and easy target domain data filtered by a threshold to perform target-to-source and target-to-target alignment (see Figure 3b).

Preliminary. For a UDA task, we denote the source domain as $D_s = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$, which contains n_s labeled samples of C categories, where $y_i^s \in \mathbb{R}^K$ is the label of x_i^s, and denote the target domain as $D_t = \{(x_i^t)\}_{i=1}^{n_t}$ which contains n_t unlabeled samples but shares the same classes with D_s. D_s and D_t are sampled from different data distributions p and q respectively, where $p \neq q$. We aim to learn a neural network $y = f(x)$ that reduces the shifts of distri-

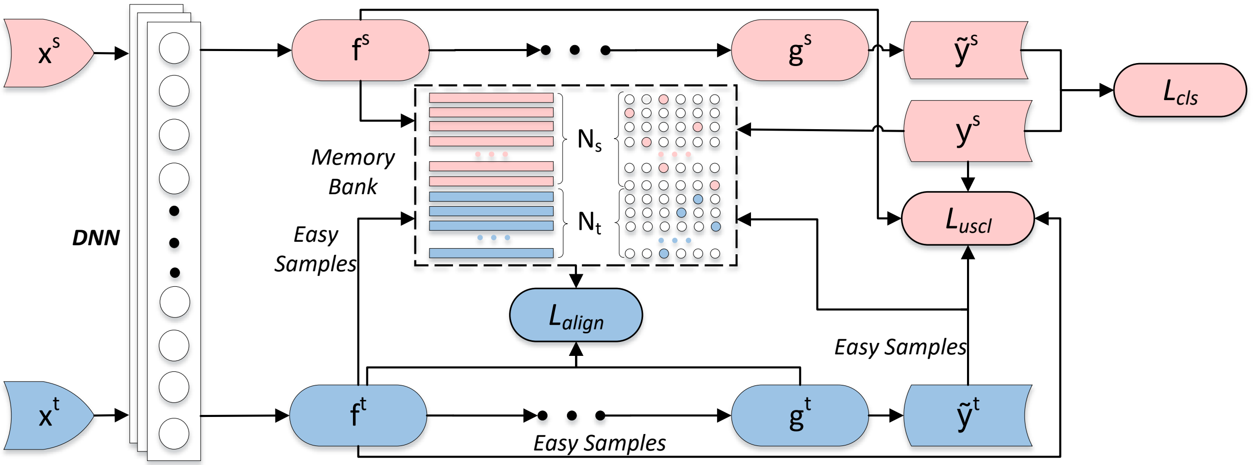


Figure 2. Pipeline of the proposed method. The source domain branch (Red) is optimized with classification loss $\mathcal{L}_{cls}$ and source subdomain contrastive learning loss $\mathcal{L}_{uscl}$. The target domain branch (Blue) is optimized with alignment loss $\mathcal{L}_{align}$. Both source and target domain branches share the same network. Meanwhile, we use a memory bank to restore the source domain data, and the easy sample from the target domain.

butions in different domains and learns transferable representations. We split D_s and D_t into C subdomains D_s^c and D_t^c respectively, where $c \in \{1, 2, ..., C\}$ denotes category label, and the distributions of D_s^c and D_t^c are p^c and q^c.

3.2. Union Subdomain Contrastive Learning

MMD assumes that if the generating distributions are identical, the statics of the two distributions should also be the same. It has been widely used to measure the distance between two distributions. However, MMD-based methods do not take into account the alignment of subdomains within the same categories. DSAN considers the relationships of relevant subdomains by proposing a new metric known as Local Maximum Mean Discrepancy (LMMD):

$$d_{\mathcal{H}}(p,q) \triangleq \boldsymbol{E}_c||\boldsymbol{E}_{p^{(c)}}[\phi(x^s)] - \boldsymbol{E}_{q^{(c)}}[\phi(x^t)]||^2_{\mathcal{H}} \qquad (1)$$

where x^s and x^t are instances from D_s and D_t, and $p^{(c)}$, $q^{(c)}$ are the distributions of $D_s^{(c)}$ and $D_t^{(c)}$. $\boldsymbol{E}_c$ is the mathematical expectation of the class. $\mathcal{H}$ is the RKHS endowed with a characteristic kernel k. $\phi(\cdot)$ denotes feature mappers that map the original samples to RKHS, the kernel k means $k(x^s, x^t) = \langle\phi(x^s), \phi(x^t)\rangle$ where $\langle\cdot,\cdot\rangle$ represents the inner product of vectors.

However, DSAN overlooks the optimization of decision boundaries between subdomains. In scenarios where we have access to the labels of source domain samples and highly confident pseudo labels of easy target domain samples, we propose a union subdomain contrastive learning module that uses the labels and pseudo labels to pull closer the samples in the same subdomains and pushes away samples in different subdomains in RKHS (see Figure 3a): f_n^s, $f_{n'}^t$ and f_n^t are the feature of source, easy target and other target domain instances, and color denotes subdomains they belong to. The brightness of the color indicates confidence level, with higher brightness meaning lower confidence, and vice versa. We formulate the distance estimator of two samples in RKHS as:

$$\hat{d'}_{\mathcal{H}}(x_i^s, x_j^s) \triangleq ||\phi(x_i^s) - \phi(x_j^s)||^2_{\mathcal{H}} \qquad (2)$$

where x_i^s and x_j^s are the i-th and j-th instance from the union of source domain samples and easy target domain samples. And inspired by supervised contrastive learning, we proposed the uscl loss:

$$\mathcal{L}_{uscl} = \sum_{i\in I} \frac{-1}{|P(i)|} \sum_{p\in P(i)} \log \frac{exp(\frac{\hat{d'}_{\mathcal{H}}(x_i^s,x_p^s)}{\tau})}{\sum\limits_{a\in A(i)} exp(\frac{\hat{d'}_{\mathcal{H}}(x_i^s,x_a^s)}{\tau})} \qquad (3)$$

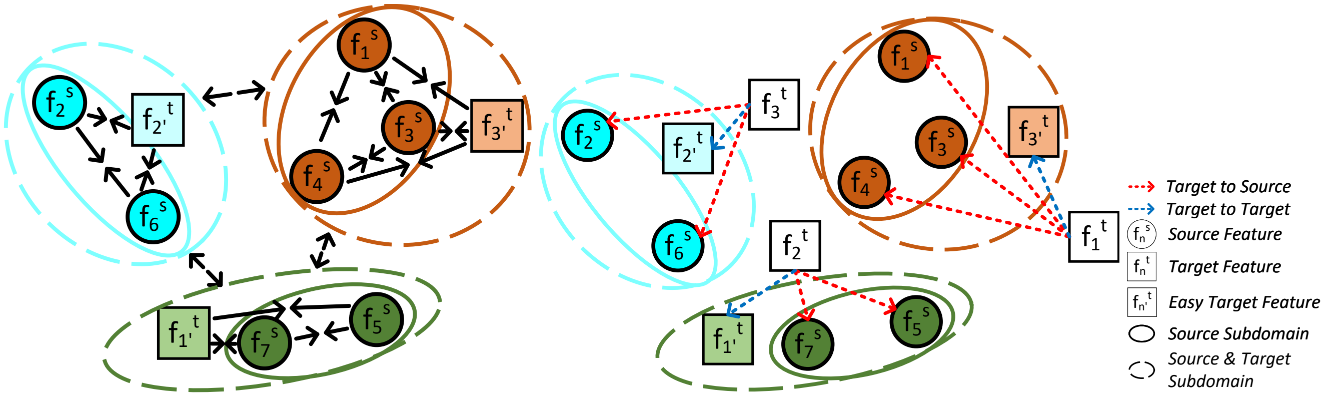
where $i \in I \equiv \{1, ..., n_u\}$ is the index of the union of source and easy target domain samples, $A(i) \equiv I\backslash\{i\}$. $P(i) \equiv \{p \in A(i) : y_p = y_i\}$ is the set of indices of samples that share the same category with x_i, and $|P(i)|$ is its corresponding cardinality. τ is the temperature parameter.

3.3. Multi-view Subdomain Alignment

Multi-view subdomain alignment consists of two modules: target-to-source alignment and target-to-target alignment. Target-to-source and target-to-target align target domain samples to source domain samples and easy target domain samples in subdomains respectively.

3.3.1 Target-to-source Alignment

As discussed in 3.2, we have built source subdomains with firm decision boundaries. It is intuitive to align samples from target domain to those subdomains while keeping subdomains unchanged. To suppress the noise, LMMD uses a weight w^c to measure how a target domain sample x_i be-



(a) USCL

(b) MvSA

Figure 3. Schematic diagram of the proposed (a) Union Subdomain Contrastive Learning (USCL), and (b) Multi-view Subdomain Alignment (MvSA). Features are represented by shapes, with circles indicating source domain and rectangles indicating target domain. Different subdomains are shown by different colored ellipses. The brightness of the color indicates confidence level, with higher brightness meaning lower confidence, and vice versa.

longs to each class:

$$w_i^c = y_{ic} / \sum\nolimits_{(x_j, y_j) \in D} y_{jc} \tag{4}$$

where y_{ic} is the c-th entry of vector y_i. We follow this setting and adopt a strategy for aligning source and target subdomains. We create a memory bank M to store features of source samples with their labels at a maximum size N_s. Every mini-batch, source and target samples first go through the network to get their features and prediction. Then we store features of source domain samples with their labels at a maximum number of N_s and drop the oldest ones. Target domain samples align with source features in the memory bank. We formulate the source to target loss as follows:

$$\mathcal{L}_{t2s} = \frac{1}{C} \sum_{c=1}^{C} || \sum_{x_i^s \in D_s^c} \frac{\phi(x_i^s)}{|D_s^c|} - \sum_{x_i^t \in D_t} w_i^{tc} \phi(x_i^t) ||_{\mathcal{H}}^2 \tag{5}$$

here, D_s^c is the indices of c-th category in source domain features in M, and $|D_s^c|$ is corresponding cardinality. And w_i^c is calculated following Equation 4.

3.3.2 Target-to-target Alignment

Target samples from the overlap of source and target distribution may be easily aligned for the reason that they are also from source distribution. But some target samples may be hard to align to their corresponding subdomains due to the large distribution difference from source distribution. Because all the target samples share the same distribution, intuitively, aligning target samples to easy samples already aligned in target domains may help samples far away from source domain samples but near easy target domain samples to be correctly aligned.

According to [1], deep networks prioritize learning simple patterns. So at the early stage of training, easy samples in target domain are more likely to be learned with high confidence. Therefore, we filter out easy samples and align target samples to those samples. We set a dynamic threshold h_t to filter out the highly confident target domain samples and store those samples in memory bank M. Similarly, we store only at most N_t easy target domain features and pseudo labels and drop the oldest ones. With training going on, some samples might be classified but with low confidence, so we decrease the threshold from high to low with the iteration grows to contain those samples:

$$h_t = h_{high} - (h_{high} - h_{low}) * i_{iter} / n_{iter} \tag{6}$$

where h_{high} and h_{low} are the high and low threshold, i_{iter} and n_{iter} are current iterations and total iterations threshold decrease rounds. At training, target features in the current mini-batch also align to the easy target features stored in M. Target samples whose confidence exceeds h_t are denoted as easy samples and will be added into M for the subsequent target-to-target alignment. Refer to Equation 5, we formulate target to target alignment loss as:

$$\mathcal{L}_{t2t} = \frac{1}{C} \sum_{c=1}^{C} || \sum_{x_j^t \in D_{t'}^c} \frac{\phi(x_j^t)}{|D_{t'}^c|} - \sum_{x_i^t \in D_t} w_i^{tc} \phi(x_i^t) ||_{\mathcal{H}}^2 \tag{7}$$

where, $D_{t'}^c$ is the indices of c-th category of target samples stored in the M_t, $|D_{t'}^c|$ is corresponding cardinality. w_i^c is calculated following Equation 4.

3.3.3 Alignment Loss

According to 3.3.1 and 3.3.2, we combine all the samples in memory bank M and formulate the overall multi-view subdomain alignment loss as:

$$\begin{aligned} \mathcal{L}_{mvsa} &= \mathcal{L}_{t2s} + \mathcal{L}_{t2t} \\ &= \frac{1}{C} \sum_{c=1}^{C} || \sum_{x_j^t \in D_u} \frac{\phi(x_j^t)}{|D_u|} - \sum_{x_i^t \in D_t} w_i^{tc} \phi(x_i^t) ||_{\mathcal{H}}^2 \end{aligned} \tag{8}$$

where $D_u = D_s^c \cup D_{t'}^c$. For classification on source domain, we introduce label smoothing [26] as $\mathcal{L}_{cls}$ into our method. Thus, the overall loss of our methods is

$$\mathcal{L} = \mathcal{L}_{cls} + \lambda_1 \mathcal{L}_{mvsa} + \lambda_2 \mathcal{L}_{uscl} \tag{9}$$

We set $\lambda_1 = 1.0$, $\lambda_2 = 0.5$ through experiments. The influence of different λ_1 and λ_2 will be explored in experiments.

4. Experiments

4.1. Datasets and Settings

Dataset Office-31 [32] is a domain adaption benchmark dataset containing 4,110 images in 31 classes from three distinct domains: Amazon (**A**), Webcam (**W**), and DSLR (**D**). Domain Amazon is collected from amazon.com, the web camera collects a webcam, and DSLR is taken by digital SLR camera with different photographic settings. We evaluate on all 6 transfer tasks **A** → **W**, **D** → **W**, **W** → **D**, **A** → **D**, **D** → **A**, **W** → **A** to avoid biases evaluations.

Office-Home [39] is a widely used benchmark dataset for domain adaptation comprising 15,500 images. It contains four domains: Art, Clipart, Product, and Real-World. Art consists of artistic images in the form of sketches, paintings, ornamentation, etc. Clipart is a collection of clipart images. Product is collected with the images without background and Real-World is a set of images of objects captured with a regular camera. Each domain contains 65 categories, with an average of around 70 images per class and a maximum of 99 images in a class. We test on all the 12 transfer tasks of **A** → **C**, **A** → **P**, **A** → **R**, **C** → **A**, **C** → **P**, **C** → **P**, **P** → **A**, **P** → **C**, **P** → **R**, **R** → **A**, **R** → **C**, **R** → **P**.

VisDA-17 [30] is a challenging domain adaption classification dataset with two distinct domains: **Real**, consisting of natural images, and **Synthetic**, consisting of 3D models from different angles and under different lighting conditions. VisDA-17 contains 280K images of 12 categories distributed across the training, validation, and testing domains. Our assessment primarily focuses on the synthetic-to-real image classification transfer task.

Implementation Details. For all experiments, we use the same settings. We linearly decrease the confidence threshold from 0.9 to 0.0 through half the epochs. The optimizer and learning rate annealing strategy follow [49]: mini-batch stochastic gradient descent (SGD) with the momentum of 0.9 is used, and SGD adjusts the learning rate using the formula: $\eta_\theta = \eta_0/(1+\alpha\theta)^\beta$, where θ is the training progress linearly changing from 0 to 1, $\eta_0 = 0.01$,$\alpha = 10$ and $\beta = 0.75$. To avoid noisy activations at the early stages of training, we also followed the adaption factor scheduler from 0 to 1 by the formula: $\lambda_\theta = \frac{2}{\exp(-\gamma\theta)} - 1$, and γ=10 is fixed throughout all the experiments [6]. In practice, λ scales both $\mathcal{L}_{uscl}$ and $\mathcal{L}_{align}$:

$$\mathcal{L} = \mathcal{L}_{cls} + \lambda_1 \lambda \mathcal{L}_{mvsa} + \lambda_2 \lambda \mathcal{L}_{uscl} \tag{10}$$

We report our method's average accuracy over 3 random trials on Office-31. We store at most 5% of the source and target domain features in all the datasets. For Office-31, we compare our methods with several deep learning methods and deep transfer learning methods: Deep Convolutional Neural Network (ResNet) [15], Deep Adaptation Network (DAN) [23], Domain Adversarial Neural Networks (DANN) [7], Adversarial Discriminative Domain Adaptation (ADDA) [37], Joint Adaptation Networks (JAN) [25], Multi-Adversarial Domain Adaptation (MADA) [29], Conditional Adversarial Domain Adaptation (CDAN and CDAN+E) [24] and Rethink MMD I and II [42], SRDC [35], RSDA-MSTN [12], FixBi [27]. For Office-Home, we compare methods with ResNet [15], DAN [23], DANN [7], JAN [25], CDAN [24], CDAN+E [24], DSAN [49], and Rethink MMD I and II [42], SRDC [35], BIWAA [43], FixBi [27]. For VisDA-17, we compare our method with ResNet [15], DANN [7], DAN [23], JAN [25], MCD [33], DSAN [49], SHOT [22], MCC+NWD [2] and FixBi [27]. All the results of baselines mentioned above are extracted from either [49] or original papers.

4.2. Results

4.2.1 Classification Results

The classification results on Office-31, Office-Home, and VisDA-17 are shown in Table 4, 1, and 2. From those experiment result tables, we can see:

1) Compared with global statistic matching-based methods (*E.g*. DAN [23], DANN [7], Rethink MMD [42]), subdomain alignment methods (*E.g*. CDAN [24], DSAN [49], ours) achieve better performance in general. We can see that aligning the global distributions of different domains only without considering the relations in subdomains does not perform well, while accurately aligning subdomains will largely improve the transfer ability of the deep network. This phenomenon reveals that aligning subdomains accurately is crucial for domain adaption.

2) Compared with other subdomain alignment methods (*E.g*. CDAN [24], DSAN [49]), our method outperform other methods as well. This result verifies the effectiveness of our proposed method.

3) Compared with non-adversarial methods (*E.g*. DAN [23], JAN [25], DSAN [49], Rethink MMD [42]), ours promotes the average performance in classification tasks.

4) Compared with the origin DASN, our method achieves better performance, which implies that our proposed USCL and MvSA do well in exactly alignment source and target subdomains.

4.2.2 Ablation Study

Proposed Modules. We conduct an ablation study to evaluate the effectiveness of the proposed components, in-

Method	SMB	A→C	A→P	A→R	C→A	C→P	C→R	P→A	P→C	P→R	R→A	R→C	R→P	Avg
ResNet [15]		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 [23]	✓	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 [7]	✓	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
JAN [25]	✓	45.9	61.2	68.9	50.4	59.7	61.0	45.8	43.4	70.3	63.9	52.4	76.8	58.3
CDAN [24]		49.0	69.3	74.5	54.4	66.0	68.4	55.6	48.3	75.9	68.4	55.4	80.5	63.8
CDAN+E [24]		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
DSAN [49]	✓	54.4	70.8	75.4	60.4	67.8	68.0	62.6	55.9	78.5	73.8	60.6	83.1	67.6
Rethink MMD-I [42]	✓	58.4	77.8	79.3	61.6	72.8	73.0	62.7	55.3	78.9	70.4	60.1	83.2	69.5
Rethink MMD-II [42]	✓	57.2	76.9	78.9	61.2	72.4	72.6	62.3	54.2	79.4	70.6	60.1	83.2	69.1
RSDA-MSTN [12]		53.2	77.7	81.3	66.4	74.0	76.5	67.9	53.0	82.0	75.8	57.8	85.4	70.9
Ours	✓	59.4	76.0	77.4	64.4	73.2	73.0	66.9	58.8	79.1	75.2	61.7	84.8	70.9
SRDC [35]		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
BIWAA [43]		56.3	78.4	81.2	68.0	74.5	75.7	67.9	56.1	81.2	75.2	60.1	83.8	71.5
FixBi [27]		58.1	77.3	80.4	67.7	79.5	78.1	65.8	57.9	81.7	76.4	62.9	86.7	72.7

Table 1. Classification accuracy (%) on Office-Home (ResNet-50). SMB denotes Statics-Matching Based.

Method	SMB	airplane	bicycle	bus	car	horse	knife	motorcycle	person	plant	skateboard	train	truck	Avg
ResNet [15]		72.3	6.1	63.4	91.7	52.7	7.9	80.1	5.6	90.1	18.5	78.1	25.9	49.4
DANN [7]		81.9	77.7	82.8	44.3	81.2	29.5	65.1	28.6	51.9	54.6	82.8	7.8	57.4
DAN [23]		68.1	15.4	76.5	87.0	71.1	48.9	82.3	51.5	88.7	33.2	88.9	42.2	62.8
JAN [25]	✓	75.7	18.7	82.3	86.3	70.2	56.9	80.5	53.8	92.5	32.2	84.5	54.5	65.7
MCD [33]		87.0	60.9	83.7	64.0	88.9	79.6	84.7	76.9	88.6	40.3	83.0	25.8	71.9
DSAN [49]	✓	90.9	66.9	75.7	62.4	88.9	77.0	93.7	75.1	92.8	67.6	89.1	39.4	75.1
Ours	✓	93.3	70.2	76.4	75.5	93.6	86.5	90.9	73.2	91.9	67.8	87.9	31.7	78.2
SHOT [22]		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
MCC+NWD [2]		-	-	-	-	-	-	-	-	-	-	-	-	83.7
FixBi [27]		96.1	87.8	90.5	90.3	96.8	95.3	92.8	88.7	97.2	94.2	90.9	25.7	87.2

Table 2. Classification accuracy (%) on VisDA-17 (ResNet101). SMB denotes Statics-Matching Based.

cluding USCL and MvSA on Office-Home by discarding one component from our method each time, see Table 5. From the result, we can see that after discarding USCL and MvSA, respectively, the accuracy of our method drops from 70.9% to 70.2% and 68.6%, respectively, but still outperforming the baseline method over 2.6% and 1.0%, showing the effectiveness of all the proposed modules in our method.

Memory Bank Size. Storage capacity can be a critical consideration in memory-sensitive systems, so it is essential to observe the relationship between performance and memory bank sizes. We evaluate our method with various memory bank sizes of different proportions of source and target domain samples. The results are shown in Table 6. Under diverse memory bank sizes, the performance doesn't make many changes. Even in a small memory bank size, *E.g.* take a number of only 5% source domain samples and 5% target domain samples, the performance outperforms others with a larger memory bank size. This phenomenon shows that our method is not memory-sensitive, contributing to the practicality and applicability of the proposed approach.

Hyper Parameters. The total loss of our proposed method is Equation 10. The influence of hyperparameters λ_1 and λ_2 is also a key factor that makes an impact on the classification performance. So we evaluate different combinations of λ_1 and λ_2 to explore the influence of those hyperparameters. Results are shown in Table 4. When $\lambda_2 = 0.0$, our method only works with MvSA, equally to the item which discards USCL module in the ablation study. With λ_2 increasing from 0.0 to 0.5, there is a performance improvement. But when λ_2 increases from 0.5 to 2.0, the performance decreases generally, so we use $\lambda_1 = 1$ and $\lambda_2 = 0.5$ through our experiments.

4.2.3 Visualization

Figure 5 visualizes the network activations of three tasks on Office-Home by DSAN and our method using t-SNE embeddings [5]. Although DSAN aligns subdomains and captures some fine-grained information for each category, many points are still scattered everywhere, making those samples hard to classify. Thanks to the USCL and the MvSA, our method promotes the similarity of distributions within the same subdomain and enhances the discriminability of individual categories. And source and target subdomains are better aligned than the baseline method DSAN, enhancing the classification performance.

4.2.4 Stability

Many domain adaption methods, including DSAN, suffer from a significant performance drop at the late stage of

Method	A→C	A→P	A→R	C→A	C→P	C→R	P→A	P→C	P→R	R→A	R→C	R→P	Avg
DSAN [49]	54.4	70.8	75.4	60.4	67.8	68.0	62.6	55.9	78.5	73.8	60.6	83.1	67.6
DSAN*(best)	54.0	70.0	75.3	59.5	69.0	66.7	61.4	56.0	77.7	73.1	60.4	82.4	67.2
DSAN*(stop)	40.8	52.3	69.3	54.8	63.9	62.2	57.7	52.6	76.2	72.7	57.8	81.6	61.8
DSAN*(change)	-13.2	-17.7	-6.0	-4.7	-5.1	-4.5	-3.7	-3.4	-1.5	-0.4	-2.6	-0.8	-5.4
Ours(best)	59.4	76.0	77.4	64.4	73.2	73.0	66.9	58.8	79.1	75.2	61.7	84.8	70.9
Ours(stop)	57.4	72.9	76.5	63.7	72.9	71.8	66.3	58.3	78.6	74.9	59.8	84.8	69.8
Ours(change)	-2.0	-3.1	-0.9	-0.7	-0.3	-1.2	-0.6	-0.5	-0.5	-0.3	-1.9	-0.0	-1.1

Table 3. Classification accuracy (%) on Office-Home at the same early stop epoch. Methods marked with * are reproduced by us. (best) shows the highest test accuracy, (stop) shows accuracy at the early stop, and (change) shows their difference.

Method	SMB	A→W	D→W	W→D	A→D	D→A	W→A	Avg
ResNet [15]		68.4±0.5	96.7±0.5	99.3±0.1	68.9±0.2	62.5 0.3	60.7±0.3	76.1
DAN [23]	✓	83.8±0.4	96.8±0.2	99.5±0.1	78.4±0.2	66.7±0.3	62.7±0.2	81.3
DANN [7]		82.0±0.4	96.9±0.2	99.1±0.1	79.7±0.4	68.2±0.4	67.4±0.5	82.2
ADDA [37]		86.2±0.5	96.2±0.3	98.4±0.3	77.8±0.3	69.5±0.4	68.9±0.5	82.9
JAN [25]	✓	85.4±0.3	97.4±0.2	99.8±0.2	84.7±0.3	68.6±0.3	70.0±0.4	84.3
MADA [29]		90.0±0.1	97.4±0.1	99.6±0.1	87.8±0.2	70.3±0.3	66.4±0.3	85.2
CDAN [24]		93.1±0.2	98.2±0.2	100.0±0.0	89.8±0.3	70.1±0.4	68.0±0.4	86.6
CDAN+E [24]		94.1±0.1	98.6±0.1	100.0±0.0	92.9±0.2	71.0±0.3	69.3±0.3	87.7
Rethink MMD-I [42]	✓	88.4	98.7	99.8	90.4	74.1	74.8	88.4
Rethink MMD-II [42]	✓	88.9	98.5	99.8	90.8	75.4	75.2	88.1
DSAN [49]	✓	93.6±0.2	98.3±0.1	100.0±0.0	90.2±0.7	73.5±0.5	74.8±0.4	88.4
Ours	✓	95.1±0.2	98.6±0.1	100.0±0.0	91.8±0.5	76.2±0.3	75.5±0.4	89.5
SRDC [35]		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
RSDA-MSTN [12]		96.1±0.2	99.3±0.2	100.0±0.0	95.8±0.3	77.4±0.8	78.9±0.3	91.1
FixBi [27]		96.1±0.2	99.3±0.2	100.0±0.0	95.0±0.4	78.7±0.5	79.4±0.3	91.4

Figure 4. Classification accuracy (%) on Office-31 (ResNet-50). SMB denotes Statics-Matching Based.

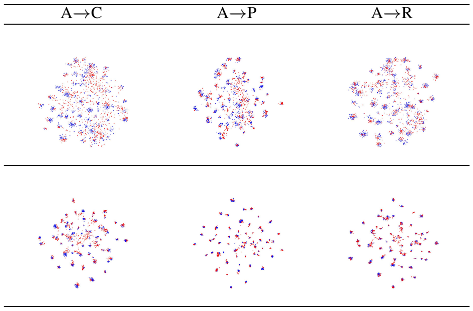


Figure 5. t-SNE of DSAN (first row) and ours (second row) on Office-Home. Red/Blue: source/target domain samples.

λ_1	λ_2	Accuracy
1.0	2.0	69.5%
1.0	1.0	70.8%
1.0	0.5	70.9%
1.0	0.0	70.2%

Table 4. Classification accuracy under different λ_1 and λ_2 on Office-Home.

Method	MvSA	USCL	Accuracy
DSAN	w/o	w/o	67.6%
Ours	w/o	w	68.6%
Ours	w	w/o	70.2%
Ours	w	w	70.9%

Table 5. Ablation studies of proposed modules on Office-Home. w indicates with while w/o without.

N_s/n_s	N_t/n_t	Accuracy
100%	100%	70.8%
50%	50%	70.8%
25%	50%	70.8%
10%	10%	70.8%
5%	5%	70.9%
1%	1%	67.8%

Table 6. Accuracy under different sizes of memory bank (proportion of total samples) on Office-Home.

training. To address this issue, our method incorporates USCL and MvDA modules. We compare the stability of our method and DASN by recording the results at the best performance epoch and the corresponding early stopping epoch on Office-Home (see Table 3): DSAN exhibits a substantial average performance decrease of 5.4% on average and a maximum decrease of 17.7%. In contrast, our method demonstrates a much smaller average decrease of only 1.1% and a maximum decrease of 2.0%. We attribute this improvement to the fact that in DSAN, once it reaches its optimal performance, hard samples from the target domain introduce significant noise and adversely impact the domain adaptation process, resulting in poor performance.

5. Conclusion

In this paper, we present a novel deep subdomain alignment method for unsupervised domain adaptation (UDA) in image classification. Our approach includes a source subdomain contrastive learning (USCL) module, which brings samples within the same subdomain closer together while pushing samples from different subdomains apart in the Reproducing Kernel Hilbert Space (RKHS). We also introduce a multi-view subdomain alignment (MvSA) strategy to reduce bias in the domain adaptation process by aligning target domain samples to both source domain samples and easy target domain samples using a dynamic thresholding scheme. Our method is evaluated on three major domain adaptation datasets and shows improved accuracy and stability. However, our method does not outperform the newest non-statistic matching-based methods, *E.g.* FixBi [27], SRDC [35] and BIWAA [43]. Further research on this phenomenon will be conducted in our future work to release this issue.

References

[1] Devansh Arpit, Stanisław Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. In *International Conference on Machine Learning*, pages 233–242, 2017. 5

[2] Lin Chen, Huaian Chen, Zhixiang Wei, Xin Jin, Xiao Tan, Yi Jin, and Enhong Chen. Reusing the task-specific classifier as a discriminator: Discriminator-free adversarial domain adaptation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7181–7190, 2022. 6, 7

[3] Nicolas Courty, Rémi Flamary, Amaury Habrard, and Alain Rakotomamonjy. Joint distribution optimal transportation for domain adaptation. *Advances in Neural Information Processing Systems*, 30:3733–3742, 2017. 1, 3

[4] Gabriela Csurka. A comprehensive survey on domain adaptation for visual applications. In *Domain Adaptation in Computer Vision Applications*, pages 1–35, 2017. 2

[5] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *International Conference on Machine Learning*, pages 647–655, 2014. 7

[6] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*, pages 1180–1189, 2015. 6

[7] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(1):2096–2030, 2016. 3, 6, 7, 8

[8] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 27:139–144, 2020. 3

[9] Arthur Gretton, Karsten Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex Smola. A kernel method for the two-sample-problem. *Advances in Neural Information Processing Systems*, 19:513–520, 2006. 3

[10] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012. 3

[11] Arthur Gretton, Dino Sejdinovic, Heiko Strathmann, Sivaraman Balakrishnan, Massimiliano Pontil, Kenji Fukumizu, and Bharath K Sriperumbudur. Optimal kernel choice for large-scale two-sample tests. *Advances in Neural Information Processing Systems*, 25:1205–1213, 2012. 3

[12] Xiang Gu, Jian Sun, and Zongben Xu. Spherical space domain adaptation with robust pseudo-label loss. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9101–9110, 2020. 6, 7, 8

[13] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pages 297–304, 2010. 3

[14] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1735–1742, 2006. 3

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 1, 6, 7, 8

[16] Olivier Henaff. Data-efficient image recognition with contrastive predictive coding. In *International Conference on Machine Learning*, pages 4182–4192, 2020. 3

[17] Junguang Jiang, Yang Shu, Jianmin Wang, and Mingsheng Long. Transferability in deep learning: A survey. *arXiv preprint arXiv:2201.05867*, 2022. 1, 2

[18] Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4893–4902, 2019. 3

[19] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33:18661–18673, 2020. 2, 3

[20] Wouter M. Kouw and Marco Loog. A review of domain adaptation without target labels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(03):766–785, 2021. 2

[21] Abhishek Kumar, Prasanna Sattigeri, Kahini Wadhawan, Leonid Karlinsky, Rogerio Feris, Bill Freeman, and Gregory Wornell. Co-regularized alignment for unsupervised domain adaptation. *Advances in Neural Information Processing Systems*, 31:9367–9378, 2018. 1, 3

[22] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 6028–6039, 2020. 6, 7

[23] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International Conference on Machine Learning*, pages 97–105, 2015. 1, 3, 6, 7, 8

[24] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. *Advances in Neural Information Processing Systems*, 31:1647–1657, 2018. 1, 6, 7, 8

[25] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *International Conference on Machine Learning*, pages 2208–2217, 2017. 1, 3, 6, 7, 8

[26] Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? *Advances in Neural Information Processing Systems*, 32:1–10, 2019. 6

[27] Jaemin Na, Heechul Jung, Hyung Jin Chang, and Wonjun Hwang. Fixbi: Bridging domain spaces for unsupervised domain adaptation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1094–1103, 2021. 6, 7, 8

[28] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 3

[29] Zhongyi Pei, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. Multi-adversarial domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, pages 1–8, 2018. 3, 6, 8

[30] 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. 6

[31] Joaquin Quinonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset shift in machine learning*. Mit Press, 2008. 1

[32] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *Proceedings of the European Conference on Computer Vision*, pages 213–226, 2010. 6

[33] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3723–3732, 2018. 6, 7

[34] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *ECCV 2016 Workshops*, pages 443–450, 2016. 1

[35] Hui Tang, Ke Chen, and Kui Jia. Unsupervised domain adaptation via structurally regularized deep clustering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8725–8735, 2020. 6, 7, 8

[36] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? *Advances in Neural Information Processing Systems*, 33:6827–6839, 2020. 3

[37] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7167–7176, 2017. 6, 8

[38] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014. 3

[39] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5018–5027, 2017. 6

[40] Jindong Wang, Yiqiang Chen, Lisha Hu, Xiaohui Peng, and Philip S. Yu. Stratified transfer learning for cross-domain activity recognition. In *IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 1–10, 2018. 1

[41] Jindong Wang, Yiqiang Chen, Han Yu, Meiyu Huang, and Qiang Yang. Easy transfer learning by exploiting intra-domain structures. In *IEEE International Conference on Multimedia and Expo*, pages 1210–1215, 2019. 1

[42] Wei Wang, Haojie Li, Zhengming Ding, Feiping Nie, Junyang Chen, Xiao Dong, and Zhihui Wang. Rethinking maximum mean discrepancy for visual domain adaptation. *IEEE Transactions on Neural Networks and Learning Systems*, 34(1):264–277, 2023. 1, 3, 6, 7, 8

[43] Thomas Westfechtel, Hao-Wei Yeh, Qier Meng, Yusuke Mukuta, and Tatsuya Harada. Backprop induced feature weighting for adversarial domain adaptation with iterative label distribution alignment. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 392–401, 2023. 6, 7, 8

[44] Garrett Wilson and Diane J Cook. A survey of unsupervised deep domain adaptation. *ACM Transactions on Intelligent Systems and Technology*, 11(5):1–46, 2020. 2

[45] Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3733–3742, 2018. 3

[46] Shaoan Xie, Zibin Zheng, Liang Chen, and Chuan Chen. Learning semantic representations for unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 5423–5432, 2018. 1

[47] Werner Zellinger, Thomas Grubinger, Edwin Lughofer, Thomas Natschläger, and Susanne Saminger-Platz. Central moment discrepancy (CMD) for domain-invariant representation learning. In *International Conference on Learning Representations*, pages 1–11, 2017. 3

[48] Liheng Zhang, Guo-Jun Qi, Liqiang Wang, and Jiebo Luo. Aet vs. aed: Unsupervised representation learning by auto-encoding transformations rather than data. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2547–2555, 2019. 3

[49] Yongchun Zhu, Fuzhen Zhuang, Jindong Wang, Guolin Ke, Jingwu Chen, Jiang Bian, Hui Xiong, and Qing He. Deep subdomain adaptation network for image classification. *IEEE Transactions on Neural Networks and Learning Systems*, 32(4):1713–1722, 2020. 1, 2, 3, 6, 7, 8